

Quantum Finite State Machines - a Circuit Based Approach

MARTIN LUKAC¹, MICHITAKA KAMEYAMA¹ AND
MAREK PERKOWSKI²

¹*Graduate School of Information Sciences, Tohoku University,
Sendai, 980-0812, Japan*

E-mail: lukacm@ecei.tohoku.ac.jp, kameyama@ecei.tohoku.ac.jp

²*Department of Computer and Electrical Engineering,
Portland State University, Portland, OR, 97201, USA.*

Received: October 16, 2012. Accepted: November 17, 2012.

We present a model of Quantum Finite State Machines (QFSM) based on the quantum circuit representation. In this model, properties of quantum bits are exploited in order to implement Quantum Finite State Machines without an explicit external memory. We show that because of the the natural ability of quantum systems to retain the quantum state, quantum circuits can be directly used to implement some classes of the quantum finite state machines. In particular, the circuit model allows to remove the logic-memory distinction as it is used in the classically established model of finite state machines. Rather, the quantum register implementing the QFSM as a circuit is alternatively used as memory and as logic. In order to allow such behavior the quantum circuit (processor) is controlled by a classical circuit that provides the necessary timing of the quantum operations as well as the interface between the quantum and the classical world. Moreover, additional mechanism for preserving unknown quantum qubit state is used and a novel computational protocol allowing the implementation of the QFSM is proposed. This framework is called the Classically Controlled Quantum Computer (CCQC). The CCQC thus combines a quantum processor with a classical controller in such manner that computation is carried from the classical inputs, through quantum processing to classical outputs.

We discuss a high level synthesis approach to quantum sequence detection based on the synthesis of quantum sequential circuits. Such machines can find many applications for instance in robotics; for example a robot is required to recognize certain sequences of input sensor values.

1 INTRODUCTION

The logic synthesis of quantum circuits focuses on obtaining permutative (Boolean Logic), quantum probabilistic, quantum fuzzy or even measurement dependent input-output mappings realized as combinational quantum devices [3, 32–35, 45, 46]. This means that the quantum circuit represents logic functions synthesized using only quantum-permutative gates [36, 45] such as [*CNOT*], [*NOT*], [*CCNOT*], [*CSWAP*] or quantum primitives such as [Controlled-*V*], [Controlled- V^\dagger] [3, 20, 29].

This approach (quantum combinational circuit) is now a base of synthesis of quantum circuits discussed by many authors [3, 20, 29, 45]. It is generally situated within a computational protocol that includes the initialization and the measurement as parts of the computation in such a manner that the combinational circuit represents the relation between input and output vectors [11, 15, 41]. Despite this general computational protocol, in the general logic synthesis procedures for quantum circuits, the measurement and initialization are often omitted; the focus of quantum logic synthesis is mainly on reversible quantum circuits where the measurement does not change the output state and the initialization is assumed not to be relevant for the synthesis process as well.

In the synthesis and design of classical or reversible Finite State Machines (FSM), in addition to the logic component a register is required [23]. The design procedure of FSM is different from standard permutative circuits because one needs to take care not only of the input values but also of the currently stored state. As such, various issues arise depending on what type of machines are being designed, e.g. synchronous vs. asynchronous, state assignment, etc. This additional constraint in the design of FSMs is the result of the distinction between the memory component and the logic component.

With respect to Quantum Finite State Machines (QFSM), also known as Quantum Finite automata (QFA), in the appropriate literature no effort has been made to separate the computation and the storage. The difference between classical and quantum QFSMs naturally exists (both technological and functional), but from the implementation point of view, this difference is not clearly seen as in the case of the classical finite state machines counterparts. This lack of distinction is due to the fact that these future quantum devices have been described mainly from a functional point of view only. Consequently, due to the lack of any specifications, the general approach to the design of QFSMs assumes a similar architecture than one would find in the classical finite state machines; a logic block and a memory block.

While quantum automata have been the subjects of many works [2, 4, 19, 26, 51] there are no papers about technical aspects of their implementation

that would clearly distinguish between the implementation of the memory part and of the logic part.

For instance the work by Bernstein *et al.* [4] defines formally the quantum Turing machines and proves theoretically their usefulness but does not describe any approach to implement such machines in quantum circuits. Similarly for other authors in the area of quantum automata; the main corpus of work is concentrated on computational capabilities of quantum automata but no technical details and design problems are discussed. This means that in order to build some of the quantum automata [19, 26] one has to first answer the question: what methodologies and techniques are required? Moreover if one considers the computationally most powerful quantum automaton [51] (one that can recognize the $\{a^n b^n\}$ language), how can such device be implemented and even more, how can it be tested? Despite the fact that in this paper we don't answer directly most of these questions, we provide a basis for implementing quantum finite state machines in realizable quantum circuit models built with standard synthesis methods.

Comparing the synthesis of reversible and quantum circuit to the the case of the QFSM, the measurement is used in a more extensive manner: the QFSMs' output is determined via an explicit usage of the measurement process [2, 4, 26, 50]. The usage of the measurement define the two main classes of the QFSMs: the one way QFSM [1, 5, 26] and the two way QFSM [2, 39, 51]. The measurement is performed to determine the final or the intermediary state of the QFSM and thus it is the closest distinction between classical usage of quantum permutative and sequential circuits. Therefore, in order to allow the synthesis of the sequential quantum circuits care must be thus taken of both the initialization and of the measurement process as well.

Another difference between the classical and quantum FSMs comes from the difference in the information carrier: the quantum particle vs. the electric current. It is well known that a quantum particle retains its state as long as it is undisturbed by the environment or by an external observer [11, 15]. Under such consideration, the memory versus logic separation is not as clear as it is in the classical non-quantum finite state machines. For instance, one can assume that if a quantum particle is artificially preserved from the natural process of decoherence, then it represents a non-decaying quantum single qubit memory. Moreover, applying to such particle a unitary operation has for effect the change of its quantum (logic) state. Such point of view on the properties of quantum systems conceptually reduces the distance between the sequential and combinatorial quantum computational devices: one can see a standard quantum circuit model [41] as a model for a sequential quantum device.

In this paper we propose a model of quantum finite state machines built in the model of classically controlled quantum circuits(CCQC) that was initially introduced in [27, 31]. We postulate that within the current technological state-of-art and under particular imposed conditions on quantum states, the quantum finite state machines can be theoretically realized directly using quantum circuits (quantum arrays) without an explicit external memory. We provide a set of theoretical requirements and definitions that allow building QFSM in the proposed model. Additionally we propose the usage of a discrete time multiplexer to control the quantum circuit's operation and a mechanism for preserving qubit(s) in unknown quantum state(s) as a part of an generalized quantum computation protocol. This protocol - if realizable in quantum technology - has the advantage of providing a novel approach to the design of quantum finite state machines because it does not differentiate between the memory and the logic operations as is done in classical CMOS circuits. Rather, all operations (storage, logic, measurement) are multiplexed in discrete time steps but are all performed on the same physical qubit.

The main contributions of this paper are:

1. A modified computational protocol that theoretically allows using quantum circuits in combination with classical controllers as quantum state machines (sequential devices). This protocol has the advantage of providing a novel approach to the design of quantum finite state machines because it does not differentiate between the memory and the state transition function.
2. The model of the Classically Controlled Quantum Computer - a classical circuit that controls a quantum circuit by switching its function between memory and logic.
3. A design method for sequence detectors - using the CCQC approach the design of sequence detectors is reduced to the design of combinatorial circuits. The problem of state assignment is resolved simply by designing the unitary transform of the reversible circuit. Contrary to the classical design of Finite State Machines that requires different procedures for state assignment which is only used for FSMs [9, 23].

This paper is organized as follows. First a background required for the understanding of the paper as well as the standard combinatorial quantum circuit protocol is given in Section 2. Section 3 introduces a quantum register, describes the differences between the quantum information and classical information with respect to a storage procedure. Section 4 introduces the Classically controlled Quantum Circuit and Section 5 introduces the concept of the Classical-Computer Quantum Circuit Finite State Machine

(CCQC-FSM) - also called QFSM in this paper. Section 6 illustrates the proposed mechanism of CCQC-FSM by implementing quantum counterparts of the well known Flip-Flops. Section 7 shows procedures of designing QFSMs sequence detection from classical data. Section 8 discusses differences between the one-way and two-way QFSMs in the CCQC-FSM model. The paper is concluded by Section 9 that discusses the presented approach, obtained results and future work.

2 BACKGROUND

2.1 Quantum Circuits

In the quantum circuit model of quantum computation the quantum system is described by the following elements:

- A quantum register (QR) represents a set of ordered qubits. Let $|\phi\rangle$ be a quantum register in an initial state given by $|\phi\rangle = |i\rangle^{\otimes n}$ for n qubits, with i being the individual qubits of the register and q_i be one of the values of the quantum system obtained by the projection on one of the observable orthonormal basis. Without any additional logic operation the quantum register can be seen as a vector of quantum bits representing the identity logic operation. Such register is in general given by a set of elementary particles evolving in a complex Hilbert space H spanned by an orthonormal set of basis states $\{\alpha_i|q_i\rangle\}$, $\sum_i |\alpha_i|^2 = 1$ [19, 41]. In other words, a quantum register initialized to one of the possible observable states $|\phi\rangle \rightarrow |m_j\rangle$ with $\{m_i\}$, $i = 0, \dots, k$ being the set of all observables and consequently measured $M|\phi\rangle \rightarrow |m_i\rangle$ will result in the same state of the register $|m_j\rangle$.
- A unitary matrix representing the logic operation on the quantum register. Such operation is for instance given by $U : H^{\otimes n} \rightarrow H^{\otimes n}$ such that $U|q\rangle = \sum_{i=0}^{2^n-1} \alpha_i|q_i\rangle$, with $|q_i\rangle$ being the basic quantum states of the system.
- A set of measurement operators; every quantum system must be measured to obtain an observable result (in the context of quantum computing it is typically $\{0, 1\}$). An example is the projective measurement operator m that projects the qubit state onto the unitary basis states $|\psi_m\rangle = \sum_{i=0}^{2^n-1} |i\rangle$. The measurement operator M_i is a non-unitary operator and for the state $|q_i\rangle$ it specifies the state $|i\rangle$ after the measurement: $|q\rangle \rightarrow \frac{M_i|q\rangle}{\sqrt{\langle q|M_iM_i^\dagger|q\rangle}} \rightarrow |i\rangle$ where the final state was obtained with probability $p_i = |\alpha_i|^2$.

When dealing with observable states, it is preferable to represent them with a density matrix notation; observables of a quantum register are

represented by the density matrix $\rho = \sum_i p_i |i\rangle\langle i|$. The density matrix is a very convenient notation to render visible the probability of observations of the observables. The manipulation of the density matrices allows to represent measurement and decomposition of a system to subsystems. For instance, a quantum register is represented by $\rho = \rho_A \otimes \rho_B$; to obtain the state of the sub-register A, one can just trace over the register B $\rho_A = \text{tra}_B(\rho)$. It is easy to see that the trace over the density matrix of ρ_B is 1; trace sums the coefficients and as each subsystem is complete by default tracing over one of them, it effectively isolates the remaining system from the traced elements.

Let a quantum gate such as Feynman, Toffoli or a Controlled-V (CV) [41] be realized on a set of qubits. A standard implementation is in general associated with a working computing protocol that can be described in three steps (schematically represented in Figure 1):

Definition 1 (Quantum Circuit Computing Protocol I (QCCPI)). A *Quantum Circuit Computing Protocol (QCCP)* for a combinatorial quantum circuit includes the following three steps:

1. Initialization - Initialize the quantum array to an initial state (Step a - Figure 1).
2. Quantum Evolution (Computation) - Perform a unitary transformation on the quantum array (Step b - Figure 1). In the Figure 1 the computation is represented by a single operator U which is built from single-qubit and two-qubit quantum gates. These gates can be quantum permutative, quantum non-permutative or in particular quantum multiple-valued or quantum fuzzy [22, 44].
3. Measurement - Measure the array (represented by a set of single qubit measurements) and observe the results (Step c - Figure 1).

The QCCP described above is a technology dependent procedure. In optical quantum computation, the initialization is represented by the controlled stream of polarized photons obtained for instance by the parametric down

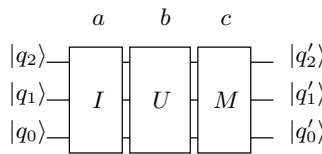


FIGURE 1

Three steps of computation in a quantum circuit. a: initialization, b: quantum evolution, c: measurement

conversion [41]. The computation is represented by a set of optical elements such as beam splitters, phase shifters and mirrors and the measurement operation is performed by observing a result of a coincidental measurement [6, 17, 18, 25] for instance. This is different from the ion trap approach to quantum computation where the initial state is generated by laser cooling a set of ions trapped in a electromagnetic field (using for instance a linear Pauli trap) and the computation is performed by a laser that changes the energy levels of the ions by throwing electrons to higher orbits. The measurement process in this case is performed also by a laser that initiates the stimulated emission that can be observed individually as fluorescence [54].

2.2 Decoherence

One of the most important problems in the quantum computation is the decoherence [41]. For the problem discussed in this paper the decoherence can be seen as a spontaneous process that irreversibly destroys a quantum state. This process can be seen as an arbitrary phase shift, a measurement operation or spontaneous absorption measurement. Such phenomena are common in all quantum technologies as a perfectly isolated quantum system does not exist in practical implementation.

For a quantum circuit, let τ be the time required for the quantum register to decohere. Then in the time interval τ , a single qubit will spontaneously evolve from its initial state to an unknown quantum state. For instance such evolution can be represented by an arbitrary $|j\rangle \rightarrow e^{-i\omega 2\pi}|j\rangle$.

For illustration some approximate results of the decoherence time are shown in (Table 1) [41].

With regards to the decoherence, a successful circuit computation can be represented as a time sequence requiring a time smaller than the minimal time required to at least apply two consecutive U_j and U_{j+1} within one valid state transition. Introducing decoherence after a time interval τ the computation on the system now can be either done in $t < \tau$ or after each operation the qubit can be repaired. This also means that the amount of computation performed before the decoherence occurs is proportional to $\frac{\tau}{t_{operation}}$. Consequently, for

System	$T_{decoherence}$	$T_{operation}$	$Max_{operations}$
Ion Trap	10^{-1}	10^{-14}	10^{13}
Quantum dot	10^{-6}	10^{-9}	10^3
Microwave cavity	10^{-5}	10^{-14}	10^9
Optical Cavity	10^0	10^{-4}	10^4

TABLE 1

Time of decoherence $T_{decoherence}$, time of operation $T_{operation}$ and maximum number of operations $Max_{operations}$ for selected technologies in seconds.

the circuit in Definition 1, the U operator can represent up to $\frac{\tau}{t_{operation}} - 1$ logic operations in series assuming that each logic operation can be done in exactly $t_{operation}$ time. Moreover it also requires that the measurement can be done before the qubit decohere, thus the time between the last operation and the end of the measurement must be at maximum as long as the time required for a logic operation.

3 PROPERTIES OF QUANTUM REGISTER

3.1 Quantum State

Definition 2 (The classical state vs. quantum state). *A state of a QR or of one of its traced out sub-elements can be preserved indefinitely in ideal conditions of isolation. Practically such quantum state can only be preserved for an extended amount of time depending on the environmental interactions and technology. This is unlike in the classical circuits; in order to preserve a classical state, energy must be continuously spend to keep the correct logic value. In quantum computation the energy spend is used to preserve unwanted interaction from reaching the quantum state.*

Definition 2 shows the main difference between the storage of information in a quantum register and in a classical register. In a quantum register, the energy is used to preserve the state by limiting its interaction with the environment; if the environment is inert no energy is required to be spent. In contrast, a classical logic circuit (for instance built in CMOS technology), needs a constant energy input in order to preserve the outputs valid.

Definition 2 is important with respect to QCCPI as in general the decoherence specifies the interval within which all the computation must be done. Definition 2 implies one more important fact. The classical logic signal is represented by a voltage value and thus exists as a traveling signal through a set of semi-conductor logic gates. Once the voltage representing a signal arrives to a storage cell it is stored and can be later on retrieved for further processing; a classical circuit requires space separation between the logic element and the storage element. In quantum circuits however, the logic signal is represented by a unique quantum particle in a specific state representing the logic value: the separation between quantum logic circuit and quantum storage resides in time.

3.2 Preserving the Unmeasured Quantum State

As was shown in Section 2.2, despite the fact that a quantum state can be seen as an naturally state-preserving medium, quantum particles are subject

to decoherence [41]. This means that due to interaction with environment or other particles or even quantum vacuum [14] a given quantum particle will in-fact be able to remain in a given state only for a limited time. Thus, a resulting state from a computation must be preserved until the next step of the computation, which can be after the natural decoherence time interval. This is very often the case of sequential devices that receive inputs from the real world.

In fact to preserve an arbitrary quantum state various techniques have been proposed such as in [10, 12, 21, 48, 52, 53]. For instance, a coherent state preservation was described to last up to 10 minutes [16] in a Spin Qubit memory. Thus one approach to avoid the decoherence is to use such technology that would allow the largest possible storage of the quantum information.

In the ion trap technology the decoherence of quantum states is minimized by a proper laser tuning and by selecting such quantum states which without strong external energy source will spontaneously transit to another state only very rarely [54]. In quantum optical computing the photons hardly interact between each other but the interaction with the environment is quite high and the life of a photon is relatively short (cf. Table 1). One could also look into liquid NMR quantum computing where the states are represented by the spin of atoms in larger molecules. The NMR approach - when proper molecules are selected - can be very stable with respect to the decoherence [41].

Another approach to prevent the decoherence is to envisage an information transfer between a computational qubit and a measurement qubit. An example of the later is for instance the mechanism of absorption and refraction of photons by atoms in the cavity QED [41]. However such mechanism is only valid for well known quantum states because the absorbed or emitted photon projects the atom to one of the orthonormal quantum states. Also, this method of preservation is not considered in this paper as it requires a new qubit and thus act as a separate in-space quantum memory.

Taking into account the preservation techniques and the Definition 2 a definition of the preservation is given:

Definition 3 (Preservation of Arbitrary Quantum State). *Preserving a quantum state corresponds to prevent decoherence or any other operation on a single or multiple qubits between two logic operations separated by an arbitrary time that would result in the change of the qubit(s). Preservation of quantum state however does not include any operation that transfer a quantum state from one qubit to another.*

Note that Definition 3 represents any active or passive mechanism (or absence of) that prevents the change of the quantum state. Naturally, the target

of the Definition 3 is mainly the preservation of a quantum state beyond decoherence as described above.

3.3 Measuring the Quantum State

Let a quantum register (QR) be a set of qubits representing a particular state $|\phi\rangle$ of a quantum circuit. Furthermore let two qubits $|a\rangle$ and $|b\rangle$ be part of the QR.

Definition 4 (The state of a QR). *The state of a qubit a in a QR is read using a measurement protocol [19, 41] while the state of the non-measured qubit b in the same QR can be described in the four categories below.*

1. *The non-measured qubit is an element of a non-entangled and non-superposed basis state, and thus the state of the system before and after the measurement will be the same up to the phase.*
2. *The non-measured qubit is an element of a fully superposed quantum state and its measurement will reduce the superposition from a superposition of 2^n basis quantum states to a superposition of 2^{n-1} basis quantum states.*
3. *The non-measured qubit is an element of a fully entangled quantum state and its measurement will result in the collapse of the whole quantum register to one of the states from the entanglement (this is the case when EPR circuit [13, 41] is used as U).*
4. *The non-measured qubit is an element of a partially entangled quantum state and its measurement will result in the destruction of a subset of qubits while leaving the rest of the register unchanged.*

Cases 1 and 2 are equivalent from the point of view of the definition 4: the measurement of the a particular qubit leaves the other qubits unaffected. Case 3 changes the state of the unmeasured qubit based on the outcome of the measurement and collapses the whole register to a non-superposed state. Case 4, being a combination of cases 1, 2 and 3, the state of the non measured qubit depends on whether or not it was entangled with the measured qubit(s).

3.4 Discrete Time-Multiplexing of Quantum Operations

Putting together Definitions 2, 3 and 4 with Section 2.1, a new general operation is added to the three operations of the QCCPI (Definition 1): when the sequence of the operations cannot be fitted into the required time period satisfying the decoherence the qubit must be preserved or repaired in order to properly store the desired information. Thus, all four operations on a single qubit can be schematically represented in Figure 2.

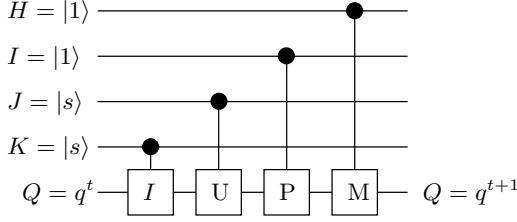


FIGURE 2

Circuit representing the four required operations for performing the initialization (I), the computation (U), the maintenance (P) and the measurement (M) of the quantum qubit.

The preservation operation (P) in Figure 2 is technology dependent. For the purpose of the problem at hand however the different types of technology related mechanisms are all represented as the maintenance operation P. Observe that both the initialization (I) and the computation (U) are controlled using the state of the input data $|s\rangle$. In the case of the U operation, this means that depending on the value of $|s\rangle$ different unitary transform will be applied. In the case of the I operation, this means that the target qubit will be initialized to the value of $|s\rangle$.

The usage of the preservation operation (P) allows to use a qubit as a memory, however as postulated in Definition 3, the preservation acts on the same qubit that is used for computation. Thus every operation - computation and preservation - must be performed at time-discrete intervals.

Lemma 1. *The four operations required to manipulate the quantum state as shown in Figure 2 must be multiplexed in a real discrete time intervals.*

Proof. The quantum state cannot be used for computation and measurement at the same time or for that matter for any two or more of the four operations. This is also not possible in Ion-trap for instance where the wavelength of the laser pulse used to preserve the stored quantum state interfere with the wavelength of the laser pulse used for the computation for instance [53]. The consequence is that a real classical clock must be used and is characterized by:

- if the clock is directly used to generate the pulse for the preservation, the clocks frequency must be equal to the inverse of the wavelength of the preserving frequency. The pulse generation can be replaced by a slower clock device connected to a device generating EM signals more reliably,
- the clock frequency must be at least equal (or larger) to the frequency of the arrival of the input data, \square

operations on the quantum circuit. This means that a quantum clock can be used if it does not generate superposed or entangled clock signals. For instance having a clock generating the Bell states is satisfactory as long as such states can be used to trigger discretely the quantum operations required to the functioning of the quantum circuit.

Finally, observe that the presented computational protocol switches the quantum circuit between two main modes: the memory mode (P) and the computation mode (I,U,M). Following the Definition 3, the memory mode contains all operations on the quantum circuit that preserve its state while the computation mode contains all operations that modify the state of the quantum circuit. Unlike in classical circuits, where these two modes are represented by two distinct physical elements in space, here the two modes are distinct only in time (Section 3.1). the main consequence for the manipulation of logic state is that, while two distinct physical elements allow for a simultaneous reading and computation, in the model presented here this is not possible because the memory and the logic is represented by the same qubit. Thus the requirement for the memory from classical circuits and classical separation in space between memory and logic is now represented in time.

4 CLASSICALLY CONTROLLED QUANTUM CIRCUIT

The discrete time multiplexing and a pseudo-Boolean quantum computer [19, 41] defines a different model of quantum computer. A computer that requires two components: (1) a quantum circuit (performing the actual quantum computation - also called the quantum processor) and (2) a classical digital computer (a generalized controller that controls the quantum circuit).

Definition 5 (Classically Controlled Quantum Circuit). *A Classically Controlled Quantum Circuit (CCQC) is composed of the following: quantum register (QR), a set of initialization operators (I), a set of Unitary operators (U), a set of preservation operators (P), a set of measurement operators (M) and a classical computer (C). The quantum register is used to represent and store a quantum state, the Unitary operators are the logic and arithmetic operations allowed on the quantum register. The measurement operators are used to observe and determine the state of the quantum register. The preservation operator is an operation allowing to keep the current state of one or more qubits in the QR. The classical computer decides the timing θ of application of each of the components of this machine to the quantum register.*

A schematic of the CCQC architecture can be represented as shown in Figure 4, with I representing the initialization mechanism, M representing

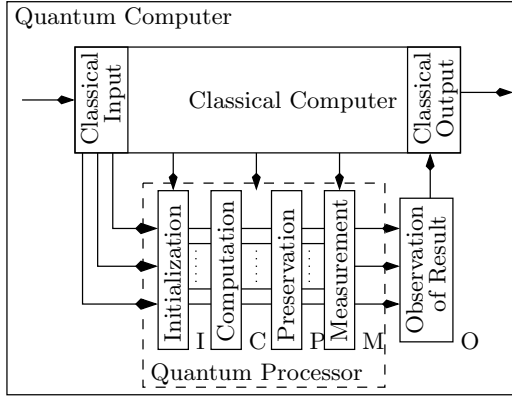


FIGURE 4

A CCQC uses a classical controller that synchronizes the quantum operations and applies appropriate transforms to the quantum register (quantum processor) in each of the phases described in Figure 1.

the measurement apparatus, O representing the actual outputs of the quantum circuit, U representing the quantum evolution/computation on the quantum register (a quantum processor) and P represents the quantum state preservation mechanism.

4.1 Computation on a Classically Controlled Quantum Circuit

To see how a quantum circuit behavior changes from computation to memory and back, let eq. 1 be the procedure to obtain the state of a set of qubits in a Quantum Circuit (QC).

$$\begin{aligned}
 \theta = 0; |\phi\rangle &= |0\rangle^{\otimes n} \rightarrow \rho_\phi = |00\dots 0\rangle\langle 00\dots 0| \\
 \theta + \delta\theta; \rho_k &= I_k \rho_\phi I_k^\dagger \\
 \theta + 2\delta\theta; \rho &= U \rho_k U^\dagger \\
 \theta + 3\delta\theta; \rho_Q &= \text{tr}_{I,O}(\rho) \\
 \theta + 4\delta\theta; \rho_Q &= P(\rho)
 \end{aligned} \tag{1}$$

The qubits of the quantum processor containing the resulting state ρ_Q - assuming they can be reused - represent a unique storage of a quantum state that upon measurement can be extracted. Observe that the stored information is shown by tracing over a density matrix (eq. 1 - this essentially means that we are not directly concerned about the exact quantum state but rather only by

the observable result. The measured qubits represent the inputs - considered to be accepted in series - and the output bits that are read on request.

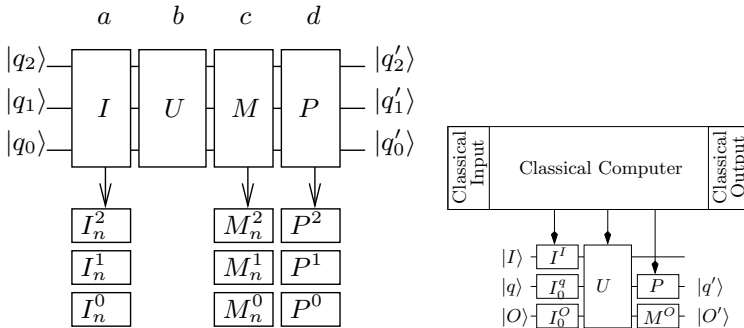
The resulting unmeasured - thus unknown quantum state given after each cycle of the computation in eq. 1 requires a method of overcoming the decoherence problem. This can be seen where the ρ_Q is processed by the P operator: after the output of the measurement has been obtained the rest of the quantum register has been preserved. Some of such techniques have been described in Section 3.2.

5 QUANTUM FINITE STATE MACHINE

Considering the Definition 2 and by manipulating the initialization and the measurement steps, a generalized computational protocol is obtained:

Definition 6 (Quantum Circuit Computing Protocol II). *A Generalized Quantum Circuit Computing Protocol (QCCP) for a generalized quantum circuit includes the following four steps:*

1. Initialization - Initialize the quantum array to an initial state (Step a - Figure 5(a)). Observe that the introduced notation for this step is parameterized: I_n^l represents initialization for each new input n



(a) The four steps of the generalized protocol for the CCQC. (b) A quantum finite state machine built from a sequential quantum circuit. Observe that there is no need to show the feedback as in a standard FSM diagram because the non-measured qubit $q - q'$ preserves its state until the next operation here represented by operation P.

FIGURE 5

The generalized CCQC shown in a circuit form and a corresponding QFSM built using the CCQC.

on each qubit l . This is different than in standard (combinational) quantum circuits where the desired initialization can be represented by I_n - the initialization is done on the whole quantum register before each computational step.

2. Quantum Evolution (Computation) - Perform a unitary transformation on the quantum array (Step b - Figure 5(a)). In the Figure 5(a) the computation is represented by a single operator U which is built from single-qubit and two-qubit quantum gates. These gates can be quantum permutative, quantum non-permutative or in particular quantum multiple-valued or quantum fuzzy [22, 44]. This step is equivalent to the one used in the standard quantum circuit model [41] shown in Figure 1.
3. Measurement - Measure the array (represented by a set of single qubit measurements) and observe the results (Step c - Figure 5(a)). Each individual measurement is represented by the operation M^l applied on each qubit l and resulting in observable m_o taking the value of 0 or 1. It is reasonable to expect that the measurement will be parameterized in a similar manner to the initialization M_n^l ; this means that while the initialization is parameterized by an input value, the measurement operator's parameter represents the expected value. For instance the measure for $|0\rangle$ means that the quantum register is being probed for a successful projection onto the axis of the observable m_0 .
4. Preservation - Apply a technology dependent operation on non measured qubits.

Using this protocol, a state machine could be created from a quantum array by allowing only certain additional assumptions. These assumptions are:

- A. Given a quantum array, any qubit from this array can be initialized individually.
- B. The individual initialization perturbs the whole quantum array in the same way as a single qubit measurement followed by a single qubit initialization.
- C. Unobserved qubits can be preserved in time for as long as it is required.

Assuming that A is satisfied, the quantum state machine is represented in Figure 5(b). In particular this means that any of the qubits in the quantum register can be initialized at different times.

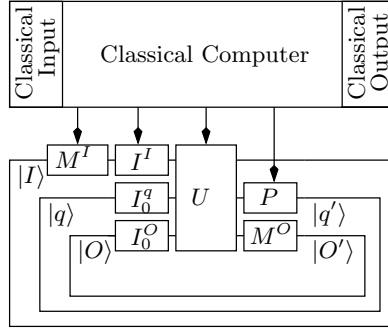


FIGURE 6

A quantum finite state machine build from a sequential quantum circuit with all its components showed. Observe that the feedback is shown for clarity; in the physical implementation there is no such feedback, but rather the same qubit is re-used in the next computational step. The feedback drawn here does not suggest using of a separate quantum memory. It is used only for notational didactic purposes.

Assumption *B* implies that from the quantum measurement point of view, we can describe the possible outcomes of a single qubit measurement on a multi-qubit system with four distinct cases (Definition 4).

In this case we consider only that the input qubit is required to be set individually while all other qubits can be only evolved. The process of setting a particular individual qubit to a given value is represented in Figure 6, where the qubit represented as $|I\rangle$ is first measured and then initialized to a desired input state.

Assumption *C* imposes that the internal state of the QFSM can be reused when next input qubit is detected.

Note that both assumptions *A* and *B* require the re-initialization of individual qubits. Such re-initializations require a measurement operation as proven by [43]. For the purpose of this study we reformulate the result of paper [43] in Definition 7.

Definition 7 (Re-initialization of Unknown Quantum State). *In order to properly reinitialize an unknown quantum state of a qubit, this qubit must be first made visible by a measurement. Thus $|j\rangle \rightarrow U_{j,k}|k\rangle \rightarrow M_k|\phi\rangle$ is valid only if the observable for a given basis of the quantum state $|\phi\rangle$ was obtained previously using the measurement operation. Such observable then allows to apply the Unitary transform $U_{j,k}$ that will change the state of the qubit to the desired target qubit $|j\rangle$.*

Definition 7 requires that the given observable is made visible on the known orthonormal basis. This means that for instance if one would use the

POVM measurement, the output of the measurement must be the result of observing one of the k expected values and not be situated in the residual measurement result of the POVM.

Theorem 1. *The only two requirements for a QFSM to be successfully implemented using the CCQC approach (assuming that Definition 2 holds true) is the availability of the selective-single qubit measurement/setting and the preservation of unknown non-measured quantum states.*

Proof. First, the requirement for the preservation of the unknown state does no need to be proven as it is a natural condition for any sequential device. Second, the proof for the selective-single qubit measurement/setting is carried in multiple steps below.

1. As the measurement or the initialization of a single qubit has for consequence its complete isolation from its previous system; for instance a system described by a state $|\psi'\rangle = U|\psi\rangle$ with U being unitary transform will, after the measurement of a single qubit a , become $\rho_a \text{tra}(|\psi_{b\dots z}\rangle) \Leftrightarrow \rho_a \otimes |\psi_{-a}\rangle$, where $|\psi_{-a}\rangle$ represents the state of the system after being measured for the qubit a .
2. Unless the measured qubit and the rest of the quantum system is entangled, the remaining system $|\psi_{b\dots z}\rangle$ is unaffected by the measurement of $|\psi_a\rangle$. This is because the quantum system is in a decomposable (and perhaps superposed) quantum state that can be written as
$$|\psi\rangle = \sum_{j=0}^{2^n} \alpha_j |j\rangle = \bigotimes_{i=0}^k |\phi_k\rangle$$
3. If the target measured qubit a is entangled and measured, the result is described by:

$$M_a |\psi\rangle = \rho_a \otimes \text{tra}(|\psi_{m_a}\rangle) \quad (2)$$

with $\text{tra}(|\psi_{m_a}\rangle)$ being the rest of the quantum system affected by the observation of the result m_a of the measurement of the qubit a M_a .

4. The initialization of a single qubit within a quantum system has similar properties as the measurement of it. The setting of a qubit with an initial value $|0\rangle$ is equivalent to: a) a measurement of the given qubit and b) to its initialization. This is a natural implication of the fact that the result of the measurement is a basis state, while the result of the initialization is a desired state in general starting from a basis state. Moreover because there is no such function that $\forall |j\rangle, U|j\rangle = |k\rangle$, with

$k = \text{constant}$, the initial state is as important as the final state. From this point of view the initialization can be formulated as a QFSM such that for a given input state it generates an appropriate initial state.

5. To prove that the quantum circuit model with the selective measurement/initialization is sufficient to build QFSM the only step remaining is a technological remark. (Other necessary requirements have been proven from a theoretical point of view [19, 26, 50]). In the presented QC representation the problems described concern mainly the speed at which the state of the machine must be retrieved before it is destroyed by the decoherence. Thus the machine must perform all required computation in $t_{\text{decoherence}} > \tau$ and the speed of the machine is given by $\frac{1}{t_{\text{operation}}}$. □

To illustrate the Definition 5 we introduce the hypothetical model of the Ring Quantum Computer (RQC). This didactic model does not restrict our approach to QFSM but shows how the classical computer controls the quantum processor as well as how given a computational protocol, the quantum bits can be reused or not. This means that this model of quantum computer behaves depending on how the classical computer controls the quantum processor. Assuming that the quantum register is reachable for a relatively arbitrary long time (trapped ions, cavity QED, particle accelerator) the stored quantum state after the computation can be seen as either the result of the computation or a memory of the previous operation.

Example 1. *Figure 7 shows a quantum computer build as a ring computer. In this example we assume three concentric rings, each containing a group of elementary particles; the input, the output and the state. The computation proceeds as follows: a set of quantum particles are emitted from the emitter E at speed s_i directly to the initialization device. We assume that either there are three emitters that can generate three separate groups of particles in parallel - one group for each ring or that the given emitter can emit groups of particles in a synchronized manner. Such emitter then can generate the particles representing input, state and output travel at the same speed and at the same location of all three rings. This means that input, state and output particles will travel at the same time through the location of the U block.*

Particles injected are initialized to a desired input state (Figure 7, block I) and travel at constant speed to the computation device U . This in turn performs an operation on the set of qubits and the resulting qubits travel to the measuring device M . There either the computation can be terminated by sending the particles to the collector end or the particles travel back to the initialization device.

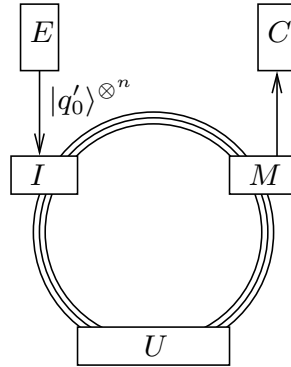


FIGURE 7

A quantum computer build as a ring.

The initialization, the unitary transformation and the measurement need to be coordinated with the speed of the particle flow. The initialization must initialize a group of quantum particles to a statistical equivalent of an input state - the same requirement is for the measurement. The Unitary transform - the computation must be done in sequence over such statistical packet of initialized particles so as the individual rotations influence only a subset of such packet and consequently allow the generation of the desired unitary transform. The preservation operation is represented by the rings; particles traveling in the ring are subject to such containment field that during each cycle of their travel (going around in the ring computer) the decoherence is almost negligible.

The reason for introducing the RQC is to explicitly show the facts that independently of whether the RQC is performing a computation on a combinatorial circuit or on a sequential circuit, the particles representing the information in the quantum circuit naturally preserve their state between each application of the Unitary transform at the location of the U block. This means that between each logic operation, the particles represent quantum memory. The re-initialization of the input particles represent a new input to the circuit. The re-initialization of the state particles represent resetting the state variables and re-initialization of the output particles represent resetting the output to some initial value. As a natural consequence of the above illustrated properties, is that not reinitializing these particles means that we store the information in the quantum memory for later usage similarly as it is done in sequential circuits.

Despite having introduced RQC as a didactic model one should consider the fact that the proposed model has several technological problems. We

briefly outline the most important ones in order to keep the didactic model as realistic as possible.

- *The particles have to travel at a constant speed*
- *As it is not possible to directly target only a particular particle p , the overall operation has a statistical behavior and as such the number of used particles will have to be quite high*
- *The ring quantum computer will behave more-likely as a tube filled with decelerated quantum particles and the results have to be statistically determined not only over the radial section of the computer but also over the radial-temporal section (the throughput) of individual rings. This double approximation allows a better control over the pulses implementing the unitary operations, initialization and the measurement.*
- *The above computer is still limited by decoherence, absorbing interference. Such phenomena will naturally make the realization of this computer even more difficult, but for the discussion in this section the decoherence is not discussed as we assume that all computation is performed within the decoherence time.*
- *Finally, in order to preserve particles in various states, the EM containment field has to be neutral with respect to the measured states. For instance assume that the EM field repels the particles based on their charge, then the measured quantity has to be the energy level or spin.*

The described operations in Example 1 are already physically possible in some of the current technologies such as Ion Trap [7, 8, 37, 38, 40, 47] or Quantum Optics [17, 18, 25, 28, 42]. However, the presented RQC model is introduced only to illustrate the separation of the classical protocol and the quantum state; the RQC is not the only architecture model we are use in this paper.

5.1 Comments on the CCQC-FSM

One could ask why CCQC and what are the advantages of the proposed model. Below the advantages and limitations are summarized:

- As will be seen in the synthesis examples, the design of a unitary matrix automatically solves the state assignment problem. The state assignment problem [23] is the problem of finding the best encoding for a set of states of an FSM. The goal of the state assignment is to find such state encoding that would minimize the logic required for the state changes.

In the case of the CCQC-FSM such technique is not required because the QFSM is represented as a unitary matrix. Designing simply a permutative circuit that satisfies the input-output mapping will automatically satisfy the state assignment such that the overall matrix specifying the QFSM is unitary. This is very unusual for the design of FSMs and thus makes the overall design much simpler.

- The representation of the QFSM as a CCQC is very advantageous because theoretically any circuit can be easily transformed into a QFSM by placing the preserve operation on the appropriate qubits. Moreover, using a single circuit various FSMS can be built according to where the preservation operation is placed. Note that in classical design, the preservation operation would represent storage, and for each added storage the state assignment must be solved.
- The time multiplexing between the logic operations and the preservation operations requires a classical control in the case that the preservation operation requires an active control signal. However, it can be relaxed to the control of only the logic operations if the preservation operation is passive (such as inert environment for instance).
- The CCQC-FSM is convenient for the design of sequence detectors but is not a general purpose model yet for arbitrary type of QFSMS.
- Under particular conditions the classical computer can be removed: if the input information is quantum and the QR is naturally in an inert environment, the machine does not need an classical controller because it will naturally preserve the state of the QR and the input information does not need to be quantized.

6 QUANTUM FLIP-FLOP AS AN EXAMPLES OF QFSM

The simplest known classical FSM is also known as flip-flop (FF). It contains one bit memory and several types exists (D-FF, T-FF and JK-FF) [23]. In this section we will look at the details of how such or equivalent device can be implemented in the CCQC-FSM model.

To analyze the implementation of some of the well known synchronous flip-flops first let's look closer at two of them; the T-FF (Figure 8a) and JK-FF (Figure 8b). The T-FF is the *toggle* flip-flop; looking at the excitation table (Figure 8a) the output of the flip-flop changes with respect to the previous stored state Q when the T input is set to '1' and remains the same otherwise. In the case of the JK-FF the excitation table (Figure 8b) shows that when $JK = 10$ the state Q is set to '1', when $JK = 01$, Q is set to '0', when $JK = 00$ the state Q remains at its previous value and when $JK = 11$ the

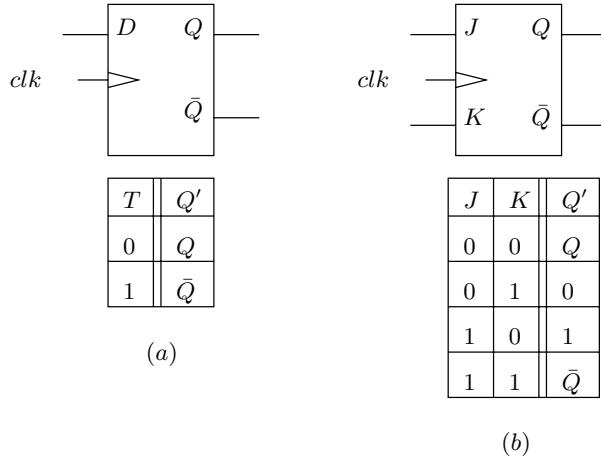


FIGURE 8
Standard synchronous Flip-Flops: (a) T Flip-Flop, (b) JK Flip-Flop.

state Q is toggled. In both Tables from Figure 8a and Figure 8b the present state is indicated by Q and the next state is indicated by Q' .

To illustrate the quantum counter part of the classical FFs introduced above the T Quantum flip-flop (T-QFF) from Figure 9a is used as an example. The device in Figure 9a has the internal state represented by the qubit q_1 and the input is given by the qubit q_0 . The output is obtained by measuring qubit q_1 ; the state qubit. For instance, let the unitary transformation U be a $[CNOT]$ quantum gate with the XOR portion of the CNOT gate acting on the bottom qubit q_1 . Then the corresponding next-state excitation function is shown in the Table 2a. The analysis of Table 2a shows that effectively the realization of a quantum equivalent of a T-QFF (Figure 9a) is implemented using a single quantum gate.

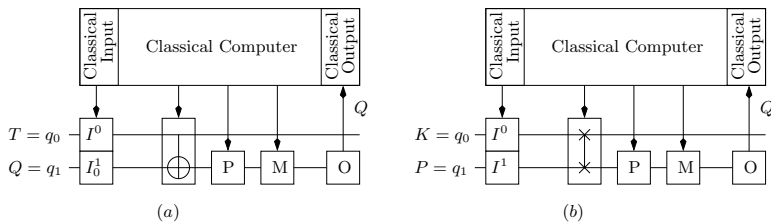


FIGURE 9
A Quantum flip-flop (QFF) implemented with two qubits. Observe that when U is $[CNOT]$ it is a standard T flip-flop, when U is the $[SWAP]$ gate it is a PK flip-flop.

q_1	$q_0(T)$	q'_1
0	0	0
0	1	1
1	0	1
1	1	0

CNOT

$q_1(K)$	$q_0(P)$	q'_1
0	0	0
0	1	1
1	0	0
1	1	1

SWAP

TABLE 2

a) Next-state excitation function of a T-QFF realized using a CNOT gate, b) Next-state excitation function of a Quantum FF using a SWAP gate

Note that the I_0^1 operator represents the initialization of the state S (qubit q_1) of the QFF and the operator I^0 represents the initialization of the QFF's input (the input qubit q_0 controls the CNOT gate). The QFF from Figure 9 is intended to accept classical input information through the classical computer which uses the initialization operators to set the QFF input and after the application of the U operator measures the output qubit to obtain classical output. Also observe that the introduced operation P is required if the state of the QFF is not observed within the decoherence time interval.

Observe that the notation used for the initialization and the measurement operations is not meant to indicate to what values is the qubit initialized or measured. Instead the focus is only to clearly indicate on what qubit and at what time during the computational process the measurement and the initialization is performed. The value that a given qubit is initialized depends on the function to be computed and on the classical input the classical computer received. Thus the I^0 initialization means that the qubit q_0 is initialized every time before the unitary computation is applied. The I_0^1 means that the qubit q_1 is initialized only at step 0 - before the overall computation is performed. Similarly, in the rest of this paper the measurement M_n^1 means that the qubit q_1 is measured after n computational steps and M^0 means that qubit q_0 is measured after every application of the computation.

Figure 9b and Table 2b shows the implementation of another type of quantum FF using a SWAP gate instead of a CNOT gate. Observe that this FF is almost equivalent to the JK-FF up to the row with the inputs '11' and thus is called the quantum PK-FF. In the traditional JK-FF when both J and K are set to value '1' the previous state stored in Q toggles its value (thus implementing the T-FF). In this case of the quantum PK-FF both the combinations of input values '00' and '11' preserves the previous quantum state. this is because the quantum PK-FF uses one of the inputs as the output Q.

To realize exact quantum equivalent to the JK-FF let's have a look on the unitary operator that could - if created - generate the same output values. The

matrix from eq. 3 shows the reversible realization of the JK-FF in a unitary operator. The inputs to this unitary operator are the JK inputs and the previous state Q and the output is the next state Q' .

$$\begin{array}{c}
 JKQ \\
 \begin{array}{cccccccc}
 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111
 \end{array} \\
 \begin{array}{c}
 000 \\
 001 \\
 010 \\
 011 \\
 100 \\
 101 \\
 110 \\
 111
 \end{array}
 \left(\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
 \end{array} \right) \quad (3)
 \end{array}$$

The unitary operator from eq. 3 could be used to built a quantum JK-FF but requires three qubits. Such device is shown in figure 10.

The QFF introduced above is different from the classical FF in the following way. First, the clock that is one of the requirement of the FF is not explicitly present. Rather the clock is integrated in the classical computer that times the required steps: initialization, computation and measurement. Also, in a flip-flop the output variable Q is in general negated in an additional output and thus a normal FF has two outputs Q and \bar{Q} (Figure 8). Such implementation might not however be always possible to realize it in quantum because the internal state of the QFF is not always known before the measurement operation.

Also observe that when another gate is used, the QFF has different states. For instance using a CV gate, will create a QFF with quantum states $|0\rangle$

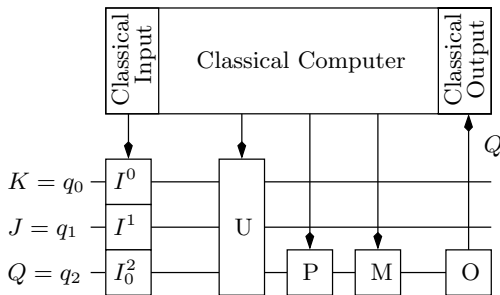


FIGURE 10

A Quantum JK Flip-Flop (JK-QFF) implemented with three qubits

and $\frac{(1+i)|0\rangle + (1-i)|1\rangle}{2}$. Then, to obtain the states $\{|0\rangle, |1\rangle\}$ an appropriate set of projective or POVM measurements must be used.

Finally, note that measurement operation is explicitly used to demonstrate the functionality of the QFF. However, it is assumed that using our model of CCQC-FSM described in Section 3 the measurement is an optional operation; if the storage is required to preserve a given quantum state it should not be measured if this information is required for later computational steps. Thus using the described computational protocol both one-way and two-way QFSM can be theoretically implemented.

7 QUANTUM LOGIC SYNTHESIS OF SEQUENCE DETECTORS

In this section, examples of design of QFSMs for sequence detection are given. The synthesis problem studied in this section is to find the simplest circuit for a sequence of given outputs. This means that the task starts from a description of some observable input-outputs of a QFSM and the goal is to design such circuit that would satisfy the known (or desired) input-output mapping.

Example 2 Synthesis of QFSM from examples of input/output sequences.

Assume the sequential oracle (representing for instance the Nature or the robot's environment) as given by a state diagram in Figure 11a. This oracle can represent partial knowledge and a deterministic or probabilistic machine of any kind. Assume that there is a clearing signal (denoted by an arrow in Figure 11a) to set the oracle into its initial state. The observer does not know the diagram of this machine - he can only observe its behavior from outside (machine in Figure 11a). By giving initial signals and input sequences and observing output sequences the observer can create a behavior tree given in Figure 11b.

Assume that the oracle from Figure 11a is represented by the sequences deduced from the diagram (Figure 11b). A set of such sequences are shown in eq. 3.

The first column of Table 3 shows the number of the input sequence, the next three columns show the three consecutive inputs to the FSM right after being initialized and the last three columns show the corresponding output sequence to the inputs. For example, the machine is initialized and then the input sequence 1 is fed to the machine. For each input at a time t^n an output with index t^n is generated. Thus the sequence of input states $|0\rangle, |0\rangle, |0\rangle$ generates the $|1\rangle, |1\rangle, |1\rangle$ output states. Such sequences can be found when starting from either the tree (Figure 11b) or from the state diagram (Figure 11a). Observe that the state qubit is not shown in (3). Naturally the

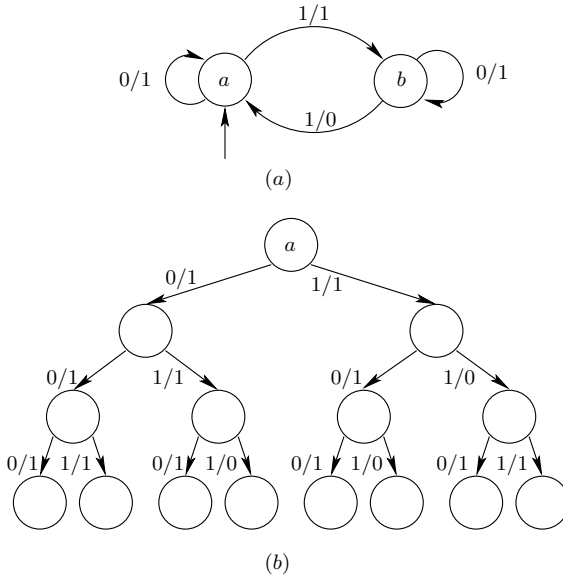


FIGURE 11
Example of a deterministic oracle and its diagnostic tree.

set of sequences is only partial as there can be an infinite number of such sequences. This is because in the oracle learning, each sequence is defined by a recursive application of the operator U . The overall procedure for the detection of a sequence of length j by the designed QFSM can be summarized as follows:

1. Initialize all qubits of the quantum register to the initial desired state,
2. repeat j times:
 - (a) Initialize the input qubit to a desired state and set the output qubit to $|0\rangle$

Sequence Number	Inputs			Outputs		
	t^0	t^1	t^2	t^0	t^1	t^2
1	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$
2	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$
3	$ 1\rangle$	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$
4	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$
5	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
6	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$

TABLE 3
Example of output sequences generated by random inputs.

		I	
		0	1
Q	a	a	b
	b	b	a

(a) q'

		I	
		0	1
Q	a	1	1
	b	1	0

(b) o

TABLE 4

State transition and Output generation table for the diagnostic tree from Figure 11. The rows represent the current states and the columns represent the current input. Table (a) shows the next state transition while Table (b) shows the output function.

(b) Apply the quantum operator on the quantum register of the QFSM

(c) Measure the output qubit and observe the result

From the diagnostic tree (Figure 11b) and the Table 3 it is possible to deduce the state transition table of QFSM (Table 4).

The state transition function is directly realizable in quantum circuit because for each combination of the input and the state a unique next state is assigned (Table 4a).

However, the Table 4b shows that while in state a the output is always 1 (no matter the input) and while in state b the output is $\bar{1}$. Because in state a the output is a constant value, the state machine cannot be directly realized on three qubits but requires one ancilla qubit that will allow to embed the non-reversible output function to a larger reversible block.

Thus the unitary transform for the output function U_o must include at least the four following input-output mappings:

$$\begin{aligned}
 |q, i, o\rangle &\rightarrow |q', i', o'\rangle \\
 |0, 0, 0\rangle &\rightarrow |0, 0, 1\rangle \\
 |0, 1, 0\rangle &\rightarrow |0, 1, 1\rangle \\
 |1, 0, 0\rangle &\rightarrow |1, 0, 1\rangle \\
 |1, 1, 0\rangle &\rightarrow |1, 1, 0\rangle
 \end{aligned} \tag{4}$$

with $|q, i, o\rangle$ representing the state, the input and the output of the QFSM respectively. Also for the eq. 4 to be correct we also assume that the output qubit is observed after each input and for each new input it is re-initialized to $|0\rangle$ (of course, this initialized value is not seen on the output qubit, and the new value is seen after the measurement).

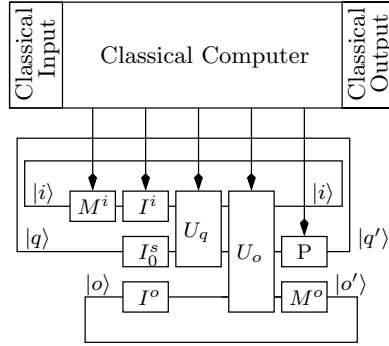


FIGURE 12

The circuit obtained from the analysis of the diagnostic tree (Figure 11).

The unitary transform used for the state transition U_q is a Controlled-NOT gate as can be seen in the eq. 5.

$$\begin{aligned}
 |i, q\rangle &\rightarrow |i', q'\rangle \\
 |0, 0\rangle &\rightarrow |0, 0\rangle \\
 |0, 1\rangle &\rightarrow |0, 1\rangle \\
 |1, 0\rangle &\rightarrow |1, 1\rangle \\
 |1, 1\rangle &\rightarrow |1, 0\rangle
 \end{aligned} \tag{5}$$

The design of these two unitary transforms conclude this example. The schematic of the circuit is shown in the Figure 12.

Observe that in both eq. 4 and eq. 5 the state qubit takes value $|0\rangle$ and $|1\rangle$ corresponding to the state a and b respectively. Moreover, notice that the state transfer function and the output generation functions are separated. This is contrary to classical decription of QFSMs where only a single function is given. This is natural because as both the state transfer and output generation functions are both reversible they can be combined into a single function. Thus combining eq. 4 and eq. 5 a combined state transfer/output function must contains at least the following transitions:

$$\begin{aligned}
 |q, i, o\rangle &\rightarrow |q', i', o'\rangle \\
 |0, 0, 0\rangle &\rightarrow |0, 0', 1\rangle \\
 |0, 1, 0\rangle &\rightarrow |1, 1', 1\rangle \\
 |1, 0, 0\rangle &\rightarrow |1, 0', 1\rangle \\
 |1, 1, 0\rangle &\rightarrow |0, 1', 0\rangle
 \end{aligned} \tag{6}$$

The approach illustrated in Example 2 can be summarized in the following pseudo-algorithm:

1. From a set of input-output pairs create a FSM table.
2. Minimize it using well-known minimization methods [23,24,49].
3. Do the state assignment using well-known methods [23,24,49].
4. Find excitation functions like for a D Flip-Flop which means the encoded transition function becomes directly the excitation function.
5. Decide what are the input, output and state qubits. Realize the excitation functions using the quantum array.
6. Add measurement and initialization devices to input, output and state qubits as discussed earlier.
7. Draw final schematic.

8 REMARKS NO ONE-WAY AND TWO-WAY QUANTUM FINITE STATE MACHINES

Finally, in the presented circuit notation, the difference between the first and the second class of the QFSM can be simply represented. The QFSM presented in this paper, if implemented without measurements on output and input qubits (the measurement is executed only after l computational steps) the QFSM becomes the well-known two-way QFSM [26] because the machine quantum state $|q\rangle$ can be in superposition with both the input and the output states. This is equivalent to stating that the reading head of a QFSM is in a superposition with the input tape as required for the time-quadratic recognition of the $\{a^n b^n\}$ language [26].

Thus, to represent the 1-way and the 2-way QFSM in the circuit notation the main difference is in the missing measurement operations between the application of the different CU (Controlled-U) operations. This is represented in Figures 13 and 14 for the 1-way and the 2-way QFSMs, respectively.

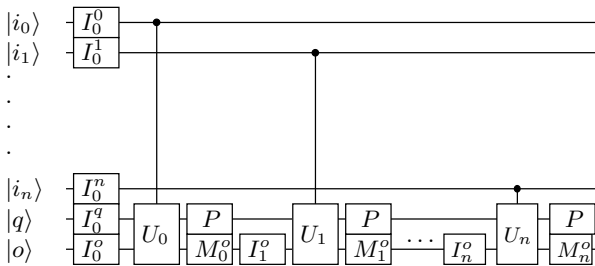


FIGURE 13
Example of circuit implementing 1-way QFSM.

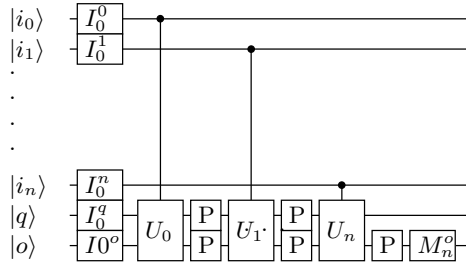


FIGURE 14
Example of circuit implementing 2-way QFSM.

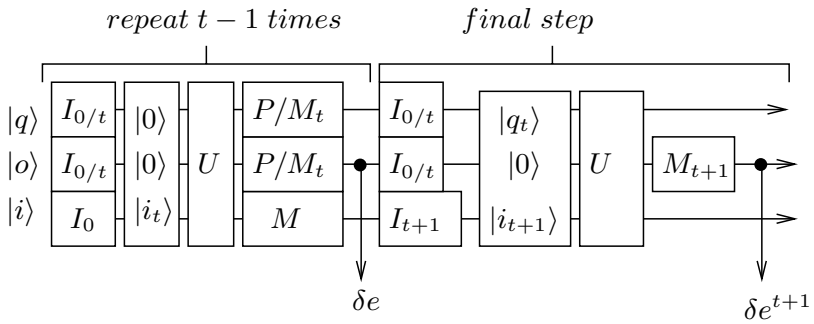


FIGURE 15
The general step by step implementation of QFSM.

To summarize, a general schema for implementing the One-way or the Two-way QFSM can be represented by a general schema shown in Figure 15. Observe that the blocks P/M represent the disinction between the One-way and Two-way QFSMs: when the block is set to perform measurement after each operation it is the One-Way QFSM and Two-Way QFSM otherwise.

9 CONCLUSIONS

In this paper we introduced the Quantum Logic Synthesis for Quantum Finite State Machines based directly on quantum combinational circuits. We showed that in essence these synthesis problems of quantum combinational arrays and automata can be solved using similar QLS methodology; by specifying either the desired machine by a state transition matrix (Unitary operator) or by an observable sequence of output states, the QLS developed for synthesis of quantum logic circuits [3, 20, 29, 30, 32, 45] can be used as well

for the synthesis of QFSMs. Unlike in other approaches to QFSMs, we provided a direct technology related methods and theory about how the QFSMs could be implemented in the future.

The provided examples illustrate the fact that designing QFSMs in the circuit model can be advantageous from at least the following two points of view:

- The synthesis design of reversible quantum circuits can be easily adapted to the synthesis of QFSMs
- The technological implementation does not require new devices or approaches when dealing with QFSMs; the same technology that is used to operate circuits can be used to operate QFSMs with the only difference is how to go about treating the individual (or groups) of elementary particles

Future works include an implementation of a fully automated algorithm in order to compare the synthesis on some selected benchmarks as well as the formulation of a complete theory on the relation between quantum circuits and QFSMs. Practical synthesis and simulation of various classes of sequential automata will be also done for several examples of language acceptors and generators from the literature [19].

One of the open problem that remains to be solved, and not only for the CCQC-FSM model is the diagnostic and analysis of arbitrary QFSMs. In this paper we provided a starting point for the analysis of QFSMs used for sequence detection. However it is not yet clear how for instance machines such as proposed in [26] could be diagnosed and verified for correctness. In classical FSM theory and practice, the sequential devices are analyzed using the homing and distinguishing input sequences. It is an open problem whether such sequences can be used successfully on QFSMs and what would be the complexity of such diagnostic process.

REFERENCES

- [1] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In *39th Annual Symposium on Foundations of Computer Science*, pages 332–341, Nov 1998.
- [2] A. Ambainis and J. Watrous. Two-way finite automata with quantum and classical states. *Theoretical Computer Science*, 287(1):299–311, 2002.
- [3] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and Weinfurter H. Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995.

- [4] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal of computing*, 26:1411–1473, 1997.
- [5] A. Brodsky and N. Pippenger. Characterization of 1-way quantum finite automata. *SIAM Journal on Computing*, 31:1456–1478, 2002.
- [6] Daniel E. Browne and Terry Rudolph. Resource-efficient linear optical quantum computation. *Phys. Rev. Lett.*, 95(1):010501, Jun 2005.
- [7] J. I. Cirac and P. Zoller. Quantum computation with cold trapped ions. *Physical Review letters*, 74(20):4091, 1995.
- [8] J. I. Cirac and P. Zoller. A scalable quantum computer with ions in an array of microtraps. *Nature*, 404:579–581, 2000.
- [9] G. De Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill, 1994.
- [10] D. Dhar, L. K. Grover, and S. M. Roy. Preserving quantum states using inverting pulses: A super-zeno effect. *Phys. Rev. Lett.*, 96:100405, Mar 2006.
- [11] P. A. M. Dirac. *The principles of quantum mechanics*. Clarendon, Oxford, 1984.
- [12] L. M. Duan, A. Kuzmich, and H. J. Kimble. Cavity QED and quantum-information processing with ‘hot’ trapped atoms. *Physical Review A*, 67:032305, 2003.
- [13] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47(10):777–780, May 1935.
- [14] H. T. Elze. Vacuum-induced quantum decoherence and the entropy puzzle. In *Workshops on “Quantum Infrared Physics”*, Paris, France, 1994.
- [15] R. P. Feynmann. Quantum mechanical computers. *Optic News*, 11:11–20, 1985.
- [16] P. T. H. Fisk, M. J. Sellars, M. A. Lawn, C. Coles, A. G. Mann, and D. G. Blair. Very high q microwave spectroscopy on trapped $^{171}\text{Yb}^+$ ions: Application as frequency standard. *IEEE Transactions on Instrumentation and Measurement*, 44:113–116, 1995.
- [17] Jaromír Fiurášek. Linear optical fredkin gate based on partial-swap gate. *Phys. Rev. A*, 78(3):032317, Sep 2008.
- [18] Yan-Xiao Gong, Guang-Can Guo, and Timothy C. Ralph. Methods for a linear optical quantum fredkin gate. *Phys. Rev. A*, 78(1):012305, Jul 2008.
- [19] J. Gruska. *Quantum computing*. Osborne/McGraw-Hill, U.S., 1999.
- [20] W.N.N. Hung, X. Song, G. Yang, J. Yang, and M Perkowski. Quantum logic synthesis by symbolic reachability analysis. In *Proceedings of DAC*, 2004.
- [21] A. Kay. Unifying quantum state transfer and state amplification. *Phys. Rev. Lett.*, 98:010501, Jan 2007.
- [22] M. H. A Khan, N. K. Siddika, and M. A. Perkowski. Minimization of quaternary galois field sum of products expression for multi-output quaternary logic function using quaternary galois field decision diagram. In *Proceedings of the International Symposium on Multiple-Valued Logic 2008*, 2008.
- [23] Z. Kohavi. *Switching and Finite Automata Theory*. McGraw-Hill, 1978.
- [24] Z. Kohavi and N.K. Jha. *Switching and Finite Automata Theory, 3rd edition*. Cambridge University Press, 2009.
- [25] Pieter Kok, W. J. Munro, Kae Nemoto, T. C. Ralph, Jonathan P. Dowling, and G. J. Milburn. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.*, 79(1):135–174, Jan 2007.
- [26] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *IEEE Symposium on Foundations of Computer Science*, pages 66–75, 1997.

- [27] M. Kumar. Engineering models and design methods for quantum state machines, Master's Thesis. Departement ECE, Portland State University, September 2008.
- [28] X. Li, Y. Wu, D. Steel, D. Gammon, T.H. Stievater, D.S. Katzer, D. Park, C. Piermarocchi, and L.J. Sham. An all-optical quantum gate in a semiconductor quantum dot. *Science*, 301(5634):809 – 811, 2003.
- [29] M. Lukac and M. Perkowski. Combining evolutionary and exhaustive search to find the least expensive quantum circuits. In *Proceedings of ULSI symposium*, 2005.
- [30] M. Lukac and M. Perkowski. Quantum mechanical model of emotional robot behaviors. In *Proceedings of the International Symposium on Multiple-Valued Logic*, 2007.
- [31] M. Lukac and M. Perkowski. Quantum finite state machines as sequential quantum circuits. In *Proceedings of ISMVL*, 2009.
- [32] M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C. H. Yu, K. Chung, H. Jee, B.-G. Kim, and Y.-D. Kim. Evolutionary approach to quantum reversible circuit synthesis. *Artif. Intell. Review.*, 20(3-4):361–417, 2003.
- [33] M. Lukac, M. Perkowski, and M. Kameyama. Evolutionary quantum logic synthesis of boolean reversible logic circuits embedded in ternary quantum space using structural restrictions. In *Proceedings of the WCCI 2010*, 2010.
- [34] D. Maslov, G. W. Dueck, and D. M. Miller. Synthesis of Fredkin-Toffoli reversible networks. *IEEE Transactions on VLSI*, 13(6):765–769, 2005.
- [35] D. Maslov, G. W. Dueck, and D. M. Miller. Techniques for the synthesis of reversible Toffoli networks. *ACM Transactions on Design Automation of Electronic Systems*, 12(4):article 41, 2007.
- [36] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In *Proceedings of DAC*, 2003.
- [37] C. Monroe, B. E. Meekhof, D. M. ind King, D. Leibriefd, W. M. Itano, and D. J. Wineland. Manipulating the motion of a single trapped atom. *Acc. Chem. Res.*, 29(12):585590, 1998.
- [38] C. Monroe, D. M. Meekhof, B. E. King, W.M. Itano, and D.J. Wineland. Demonstration of a fundamental quantum logic gate. *Physical Review Letters*, 75:4717–4717, 1995.
- [39] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237:275–306, 2000.
- [40] H. C. Nägerl, R. G. Hulet, W. M. Itano, and D. J. Wineland. Laser addressing of individual ions in a linear ion trap. *Physical Review A*, 60:145–148, 1999.
- [41] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [42] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning. Demonstration of an all-optical quantum controlled-not gate. *Nature*, 426:264–267, 2004.
- [43] Arun Kumar Pati and Samuel L. Braunstein. Impossibility of deleting an unknown quantum state. *Nature*, 404:164–165, 2000.
- [44] A. Raghuvanshi and M. Perkowski. Using fuzzy quantum logic to learn facial gestures of a schrödinger cat puppet for robot theatre. In *Proceedings of the International Symposium on Multiple-Valued Logic*, 2008.
- [45] V. V. Shende, S. S. Bullock, and I. L. Markov. A practical top-down approach to quantum circuit synthesis. In *Proceedings of Asia Pacific DAC*, 2005.
- [46] J. Smolin and D. P. DiVincenzo. Five two-qubit gates are sufficient to implement the quantum fredkin gate. *Physical Review A*, 53(4):2855–2856, 1996.

- [47] A. Sørensen and K. Mølmer. Entanglement and quantum computation with ions in thermal motion. *Physical Review A*, 62:022311, 2000.
- [48] H. P. Specht, Ch. Nilleke, A. Reiserer, M. Uphoff, E. Figueroa, S. Ritter, and G. Rempe. A single-atom quantum memory. *Nature*, 473:190193, 2011.
- [49] J. F. Wakerly. *Digital Design: Principles and Practices Package, 4th edition*. Prentice Hall, 2005.
- [50] J. Watrous. On one-dimensional quantum cellular automata. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 528–537, 1995.
- [51] J. Watrous. On the power of 2-way quantum finite state automata. Technical Report CS-TR-1997-1350, 1997.
- [52] D. Wineland and T. Heinrichs. Ion trap approaches to quantum information processing and quantum computing. A *Quantum Information Science and Technology Roadmap*, N/A, 2004.
- [53] D. Wineland, C. Monroe, W. Itano, B. King, D. Leibfried, D. Meekhof, C. Myatt, and C. Wood. Experimental primer on the trapped ion quantum computer. In *Quantum Computing: Where Do We Want to Go Tomorrow?* (ed S. L. Braunstein). Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, FRG., 2005.
- [54] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, , and D. M. Meekhof. Experimental issues in coherent quantum-state manipulation of trapped atomic ions. *Journal of Research of the National Institute of Standards and Technology*, 103(259):259, 1998.