



(19) **United States**

(12) **Patent Application Publication**
Ozfidan

(10) **Pub. No.: US 2019/0324943 A1**

(43) **Pub. Date: Oct. 24, 2019**

(54) **SYSTEMS AND METHODS FOR CALCULATING THE GROUND STATE OF NON-DIAGONAL HAMILTONIANS**

(52) **U.S. Cl.**
CPC *G06F 15/82* (2013.01); *G06N 10/00* (2019.01)

(71) Applicant: **D-WAVE SYSTEMS INC.**, Burnaby (CA)

(57) **ABSTRACT**

(72) Inventor: **A. Isil Ozfidan**, Burnaby (CA)

(21) Appl. No.: **16/375,567**

(22) Filed: **Apr. 4, 2019**

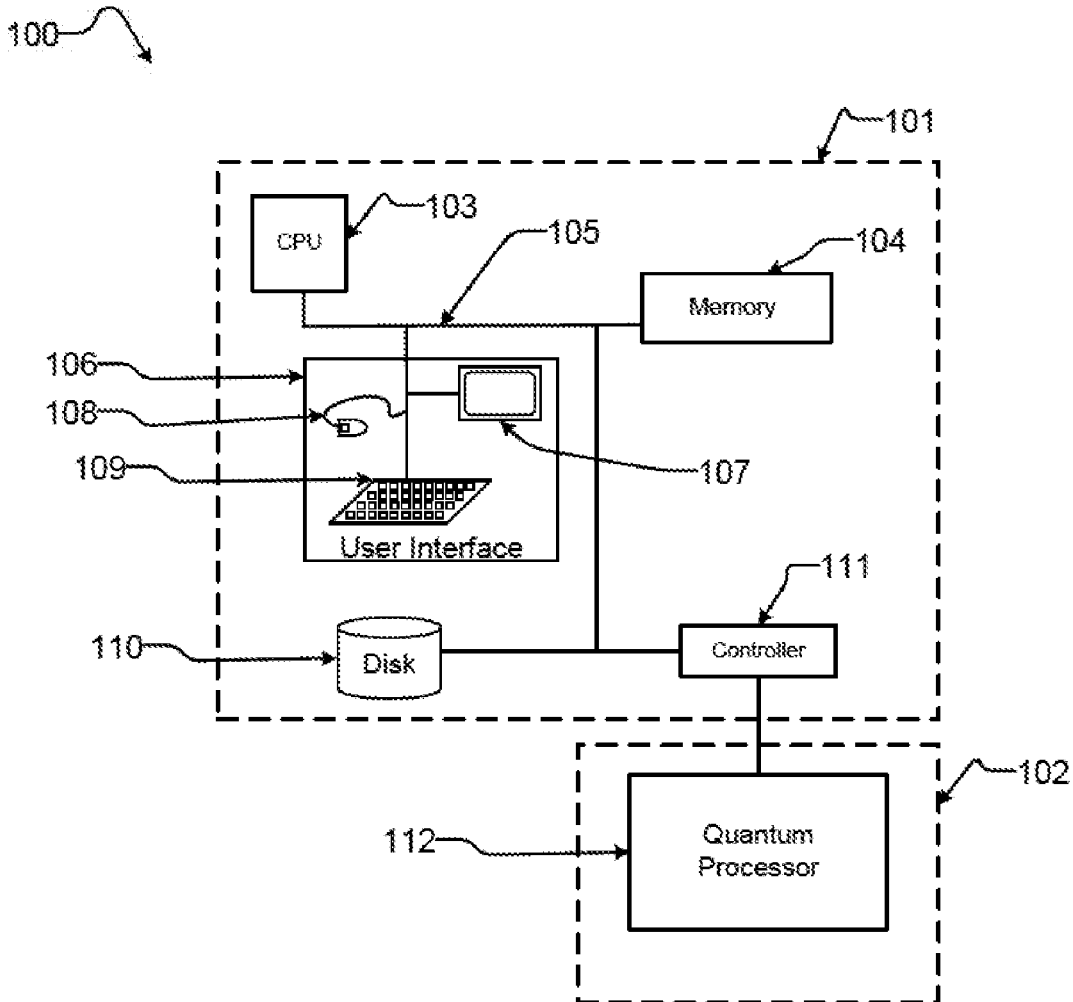
Related U.S. Application Data

(60) Provisional application No. 62/661,986, filed on Apr. 24, 2018.

Publication Classification

(51) **Int. Cl.**
G06F 15/82 (2006.01)
G06N 10/00 (2006.01)

A hybrid computing system comprising a digital and an analog processor calculates the ground energy state of a non-diagonal Hamiltonian via diagonalization of the Hamiltonian in different bases and reverse annealing. A first basis is rotated to render part of the Hamiltonian diagonal, then the quantum processor evolves backwards until a value s^* of the normalized evolution coefficient. Another basis is rotated to render another part of the Hamiltonian diagonal and the quantum processor evolves backwards again until s^* . The bases can be rotated via discrete Fourier transform. The quantum processor may pause for a time t after each backward evolution. The ground state energy is calculated using the final ground states.



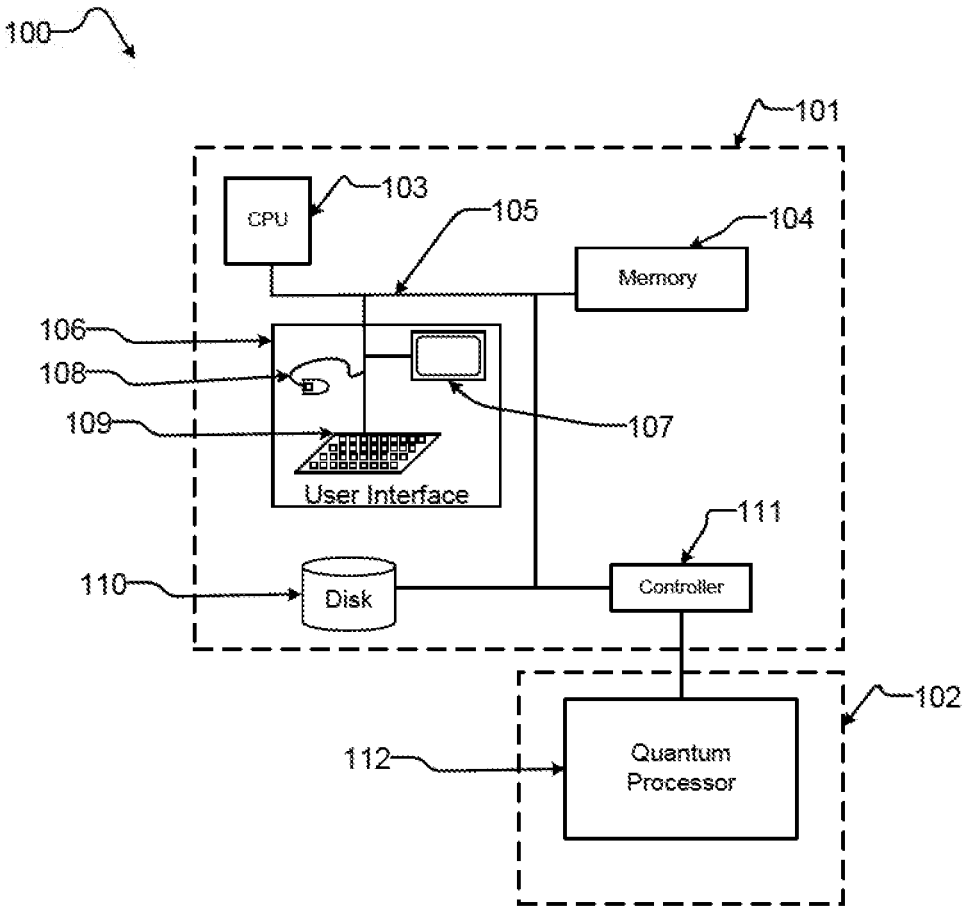


FIGURE 1

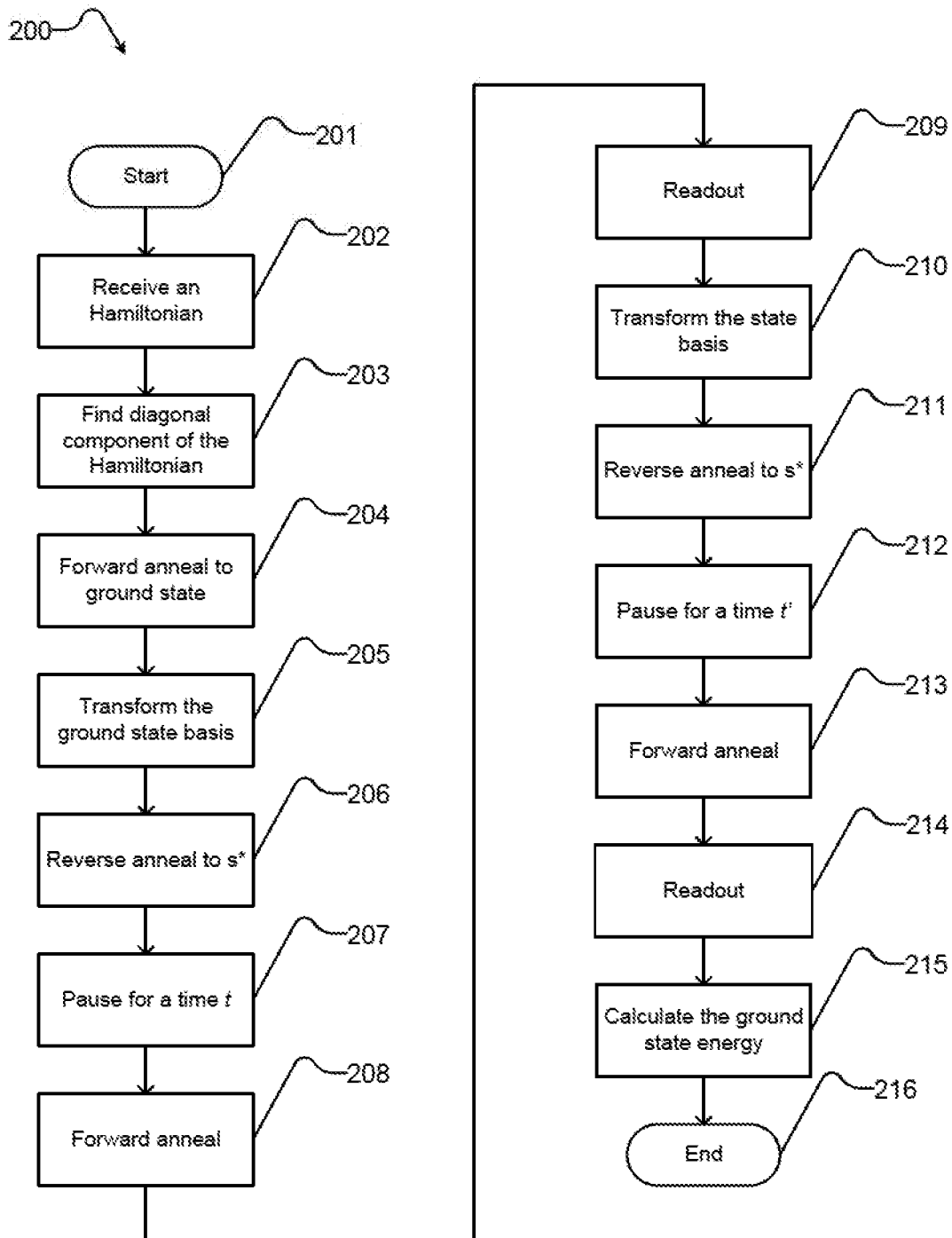


FIGURE 2

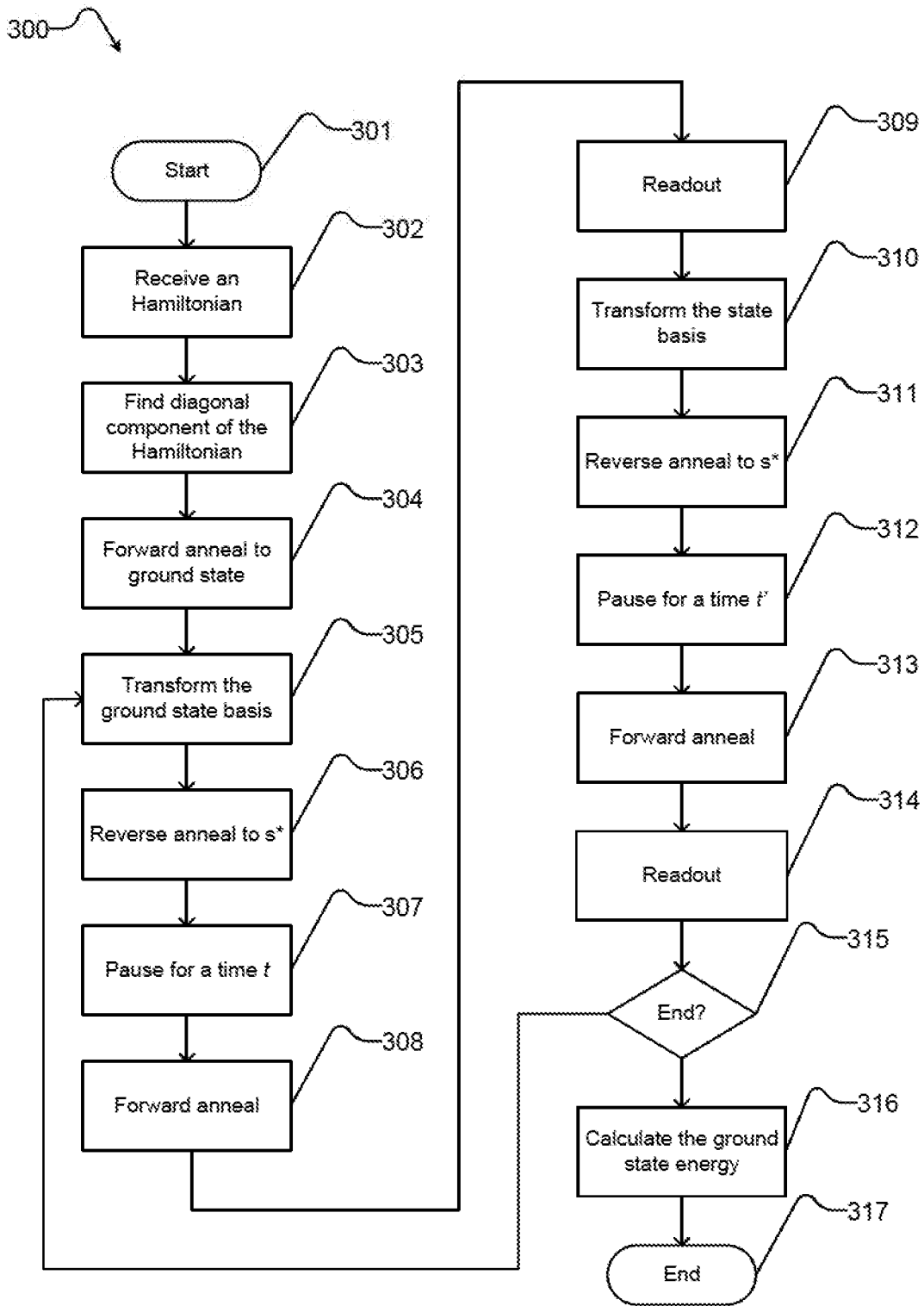


FIGURE 3

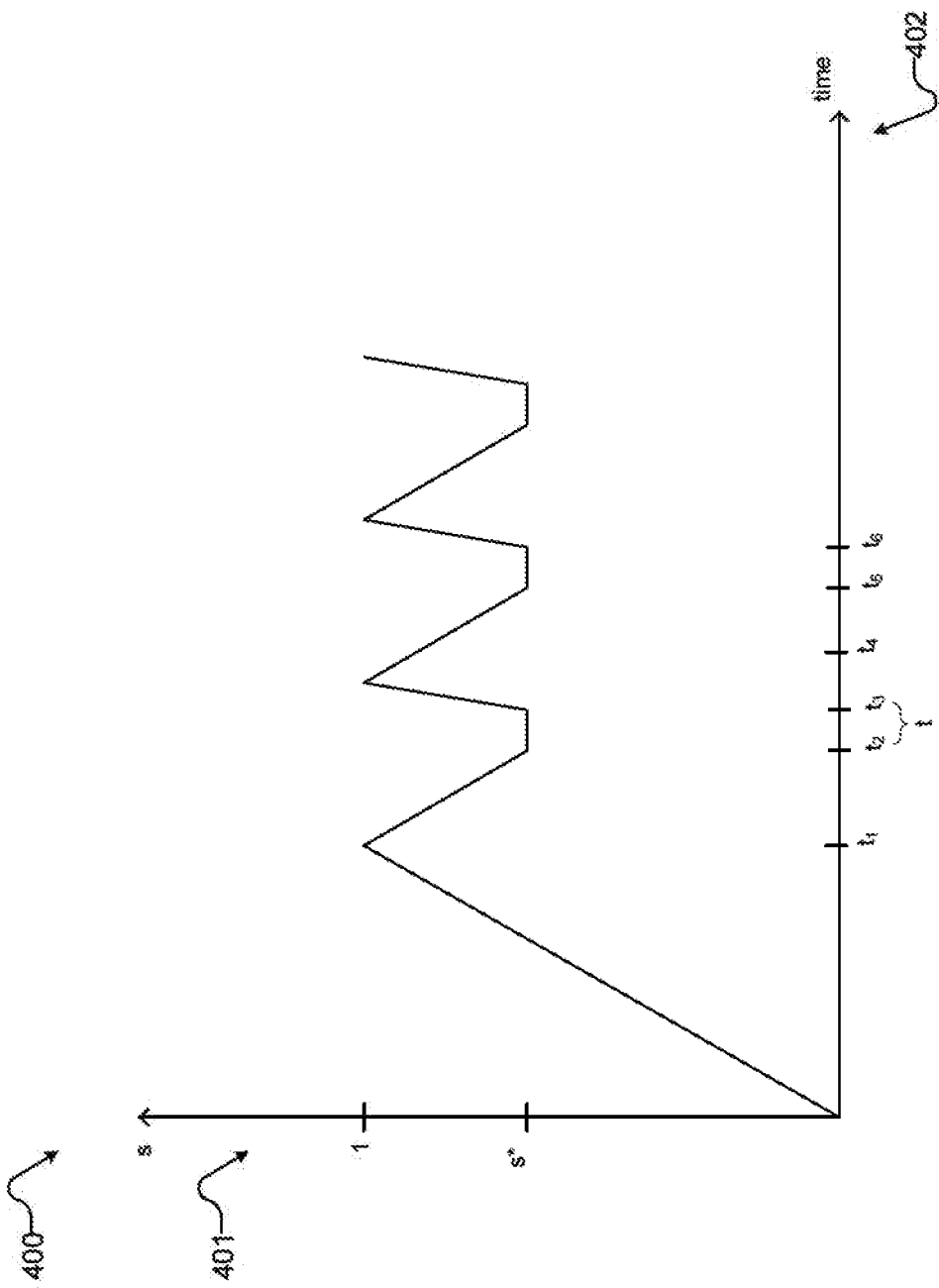


FIGURE 4

**SYSTEMS AND METHODS FOR
CALCULATING THE GROUND STATE OF
NON-DIAGONAL HAMILTONIANS**

FIELD

[0001] This disclosure generally relates to calculating the ground state of non-diagonal Hamiltonians in adiabatic quantum computers.

BACKGROUND

[0002] Quantum annealing is a computation method that may be used to find a low-energy state, typically preferably the ground state, of a system. Similar in concept to classical simulated annealing, the method relies on the underlying principle that natural systems tend towards lower energy states because lower energy states are more stable. However, while classical annealing uses classical thermal fluctuations to guide a system to a low-energy state and ideally its global energy minimum, quantum annealing may use quantum effects, such as quantum tunneling, as a source of delocalization to reach a global energy minimum more accurately and/or more quickly than classical annealing. In quantum annealing thermal effects and other noise may be present. The final low-energy state may not be the global energy minimum. Adiabatic quantum computation may be considered a special case of quantum annealing for which the system, ideally, begins and remains in its ground state throughout an adiabatic evolution. Thus, those of skill in the art will appreciate that quantum annealing systems and methods may generally be implemented on an adiabatic quantum computer. Throughout this specification and the appended claims, any reference to quantum annealing is intended to encompass adiabatic quantum computation unless the context requires otherwise.

[0003] Quantum annealing uses quantum mechanics as a source of disorder during the annealing process. An objective function, such as an optimization problem, is encoded in a Hamiltonian H_p , and the algorithm introduces quantum effects by adding a disordering Hamiltonian H_D that does not commute with H_p . An example case is:

$$H_E \propto A(t)H_D + B(t)H_p$$

[0004] where $A(t)$ and $B(t)$ are time dependent envelope functions. For example, $A(t)$ can change from a large value to substantially zero during the evolution and H_E can be thought of as an evolution Hamiltonian. The disorder is slowly removed by removing H_D (i.e., by reducing $A(t)$).

[0005] Thus, in quantum annealing the system starts with an initial Hamiltonian and evolves through an evolution Hamiltonian to a final “problem” Hamiltonian H_p whose ground state encodes a solution to the problem. If the evolution is slow enough, the system may settle in the global minimum (i.e., the exact solution), or in a local minimum close in energy to the exact solution. The performance of the computation may be assessed via the residual energy (difference from exact solution using the objective function) versus evolution time. The computation time is the time required to generate a residual energy below some acceptable threshold value. In quantum annealing, H_p may encode an optimization problem and therefore H_p may be diagonal in the subspace of the qubits that encode the solution, but the system does not necessarily stay in the ground state at all times. However a problem Hamiltonian that is non-diagonal in the subspace of the qubits may not be readily mapped to

an adiabatic quantum system. There is thus a general desire for systems and methods for method to apply non-diagonal Hamiltonians to adiabatic quantum systems.

[0006] The energy landscape of H_p may be crafted so that its global minimum is the answer to the problem to be solved, and low-lying local minima are good approximations.

[0007] The foregoing examples of the related art and limitations related thereto are intended to be illustrative and not exclusive. Other limitations of the related art will become apparent to those of skill in the art upon a reading of the specification and a study of the drawings.

BRIEF SUMMARY

[0008] A hybrid computing system comprises a quantum processor and a digital processor. A non-diagonal Hamiltonian comprises a first term and a second term, the first term diagonal in a first basis and the second term diagonal in a second basis, the second basis different from the first basis. A method of operating a hybrid computing system to calculate the ground state of a non-diagonal Hamiltonian comprises programming the quantum processor to evolve forwards with the first term in the first basis to a first ground state; applying a first transformation to the first basis; programming the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ; pausing the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis; applying a second transformation to the second basis; programming the quantum processor to evolve backwards with the first terms in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ; pausing the evolution of the quantum processor for a second period of time during the evolving backwards with the first term in the first basis; programming the quantum processor to evolve forwards with the first term in the first basis to a second ground state; reading out the second ground state; and using the second ground state to calculate the ground energy of the non-diagonal Hamiltonian.

[0009] The second transformation may be the inverse of the first transformation. The first transformation can be a discrete Fourier transform. The non-diagonal Hamiltonian may include a plurality of terms and the first transformation may render more than one term of the non-diagonal Hamiltonian diagonal in the first basis. The method may further comprise programming the quantum processor to evolve forward with the second term in the second basis to a third ground state after pausing the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis; and reading out the third ground state. The quantum processor may evolve forwards with the first term in the first basis to a first set of ground states; and may evolve forwards with the first term in the first basis to a second set of ground states. The value s_2^* of the normalized evolution coefficient s may be equal to s_1^* . The non-diagonal Hamiltonian may describe the electronic structure of a period lattice.

[0010] The method may further comprise until an exit condition is met, iteratively repeating: applying a first transformation to the first basis; programming the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ; pausing the evolution of the quantum

processor a first period of time during the evolving backwards with the second term in the second basis; applying a second transformation to the second basis; programming the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ; pausing the evolution of the quantum processor **100** for a second period of time during the evolving backwards with the first term in the first basis; programming the quantum processor to evolve forwards with the first term in the first basis to a n^{th} ground state; and reading out the n^{th} ground state. The method may further comprise determining whether an exit condition has been met. The exit condition may be at least one of: an error condition is below an error **105** threshold and a number of evolution iterations has been completed.

[0011] A hybrid computing system comprises a quantum processor and a digital processor, and the digital processor operable to program the quantum processor with a first term of a non-diagonal Hamiltonian, the non-diagonal Hamiltonian comprising the first term and a second term, the first term diagonal in a first basis and the second term diagonal in a second basis different from the first basis, to evolve forwards in the first basis to a first ground state; apply a first transformation to the first basis; program the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ; pause the evolution of the quantum processor **115** for a first period of time during the evolving backwards with the second term in the second basis; apply a second transformation to the second basis; program the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ; pause the evolution of the quantum processor for a second period of time during the evolving backwards with the first term in the first basis; program the quantum processor to evolve forwards with the first term in the first basis to a second ground state; read out the second ground state; and use the second ground state to calculate the ground energy of the non-diagonal Hamiltonian.

[0012] The second transformation may be the inverse of the first **125** transformation. The first transformation may be a discrete Fourier transform. The non-diagonal Hamiltonian may include a plurality of terms and the digital processor may be operable to apply a first transformation to the first basis so that more than one term of the non-diagonal Hamiltonian becomes diagonal. The digital processor may be further operable to program the quantum processor to evolve forward with **130** the second term in the second basis to a third ground state after pausing the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis and to read out the third ground state.

[0013] The digital processor is further operable to program the quantum processor **135** to evolve forwards in the first basis to a first set of ground states and to cause the quantum processor to evolve forwards in the first basis to a second set of ground states. s_2^* may be equal to s_1^* . The non-diagonal Hamiltonian may describe the electronic structure of a period lattice.

[0014] The digital processor may be further operable to, until an exit condition is met, iteratively repeat apply a first transformation to the first basis; program the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized

evolution coefficient; pause the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis; apply a second transformation to the second basis; program the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient; pause the evolution of the quantum processor for a second period of time during the evolving backwards with the first term in the first basis; program the quantum processor to evolve forwards with the first term in the first basis to a n^{th} ground state; and read out the n^{th} ground state.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S)

[0015] In the drawings, identical reference numbers identify similar elements or acts. The sizes and relative positions of elements in the drawings are not necessarily drawn to scale. For example, the shapes of various elements and angles are not necessarily drawn to scale, and some of these elements may be arbitrarily enlarged and positioned to improve drawing legibility. Further, the particular shapes of the elements as drawn, are not necessarily intended to convey any information regarding the actual shape of the particular elements, and may have been solely selected for ease of recognition in the drawings.

[0016] FIG. 1 is a schematic diagram showing a hybrid processor including at least one quantum processor and at least one digital processor according to the present system, devices, articles and methods.

[0017] FIG. 2 is a flow diagram showing an example method of operating a hybrid processor system to calculate a ground state of a non-diagonal Hamiltonian with the hybrid computing system.

[0018] FIG. 3 is a flow diagram showing an example iterative method of operating a hybrid processor system to calculate a ground state of a non-diagonal Hamiltonian with the hybrid computing system.

[0019] FIG. 4 is a chart showing an example evolution of an analog processor performed during an example implementation of the methods of FIG. 2 and FIG. 3.

DETAILED DESCRIPTION

[0020] In the following description, certain specific details are set forth in order to provide a thorough understanding of various disclosed implementations. However, one skilled in the relevant art will recognize that implementations may be practiced without one or more of these specific details, or with other methods, components, materials, etc. In other instances, well-known structures associated with computer systems, server computers, and/or communications networks have not been shown or described in detail to avoid unnecessarily obscuring descriptions of the implementations.

[0021] Unless the context requires otherwise, throughout the specification and claims that follow, the word “comprising” is synonymous with “including,” and is inclusive or open-ended (i.e., does not exclude additional, unrecited elements or method acts).

[0022] Reference throughout this specification to “one implementation” or “an implementation” means that a particular feature, structure or characteristic described in connection with the implementation is included in at least one

implementation. Thus, the appearances of the phrases “in one implementation” or “in an implementation” in various places throughout this specification are not necessarily all referring to the same implementation. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more implementations.

[0023] As used in this specification and the appended claims, the singular forms “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise. It should also be noted that the term “or” is generally employed in its sense including “and/or” unless the context clearly dictates otherwise.

[0024] The headings and Abstract of the Disclosure provided herein are for convenience only and do not interpret the scope or meaning of the implementations.

[0025] FIG. 1 illustrates a hybrid computing system **100** including a digital computer **101** coupled to an analog computer **102**. In some implementations the analog computer **102** is a quantum computer. The example digital computer **101** includes a digital processor (CPU) **103** that may be used to perform classical digital processing tasks.

[0026] Digital computer **101** may include at least one digital processor (such as central processor unit **103** with one or more cores), at least one system memory **104**, and at least one system bus **105** that couples various system components, including system memory **104** to central processor unit **103**. The digital processor may be any logic processing unit, such as one or more central processing units (“CPUs”), graphics processing units (“GPUs”), digital signal processors (“DSPs”), application-specific integrated circuits (“ASICs”), programmable gate arrays (“FPGAs”), programmable logic controllers (PLCs), etc.

[0027] Digital computer **101** may include a user input/output subsystem **106**. In some implementations, the user input/output subsystem includes one or more user input/output components such as a display **107**, mouse **108**, and/or keyboard **109**.

[0028] System bus **105** can employ any known bus structures or architectures, including a memory bus with a memory controller, a peripheral bus, and a local bus. System memory **104** may include non-volatile memory, such as read-only memory (“ROM”), static random-access memory (“SRAM”), Flash NANO; and volatile memory such as random access memory (“RAM”) (not shown).

[0029] Digital computer **101** may also include other non-transitory computer or processor-readable storage media or non-volatile memory **110**. Non-volatile memory **110** may take a variety of forms, including: a hard disk drive for reading from and writing to a hard disk, an optical disk drive for reading from and writing to removable optical disks, and/or a magnetic disk drive for reading from and writing to magnetic disks. The optical disk can be a CD-ROM or DVD, while the magnetic disk can be a magnetic floppy disk or diskette. Non-volatile memory **110** may communicate with digital processor via system bus **105** and may include appropriate interfaces or controllers **111** coupled to system bus **105**. Non-volatile memory **110** may serve as long-term storage for processor- or computer-readable instructions, data structures, or other data (sometimes called program modules) for digital computer **101**.

[0030] Although digital computer **101** has been described as employing hard disks, optical disks and/or magnetic disks, those skilled in the relevant art will appreciate that

other types of non-volatile computer-readable media may be employed, such magnetic cassettes, flash memory cards, Flash, ROMs, smart cards, etc. Those skilled in the relevant art will appreciate that some computer architectures employ volatile memory and non-volatile memory. For example, data in volatile memory can be cached to non-volatile memory. Or a solid-state disk that employs integrated circuits to provide non-volatile memory.

[0031] Various processor- or computer-readable instructions, data structures, or other data can be stored in system memory **104**. For example, system memory **104** may store instruction for communicating with remote clients and scheduling use of resources including resources on the digital computer **101** and analog computer **102**.

[0032] In some implementations system memory **104** may store processor- or computer-readable calculation instructions to perform pre-processing, coprocessing, and post-processing to analog computer **102**. System memory **104** may store at set of analog computer interface instructions to interact with the analog computer **102**.

[0033] Analog computer **102** may include an analog processor such as quantum processor **112**. The analog computer **102** can be provided in an isolated environment, for example, in an isolated environment that shields the internal elements of the quantum computer from heat, magnetic field, and other external noise (not shown).

[0034] Hybrid computing system **100** may be used to calculate the ground state of non-diagonal Hamiltonians, for example Hamiltonians defining the electronic structure of matter. The systems and methods described in the present disclosure and the appended claims may be used in the field of quantum chemistry.

[0035] Electronic structure Hamiltonians have two parts, kinetic (T) and potential (U). In the plane wave basis, the kinetic part T of the Hamiltonian is diagonal and can be represented by local σ_z terms. The potential part U of the Hamiltonian can be diagonalized by taking the Fourier transform of the basis, thus allowing U to be represented by $\sigma_z^i \sigma_z^j$ and σ_z^i terms, making T non-diagonal.

[0036] A Hamiltonian $H = \sum H_i$ can be decomposed via Trotterization into fundamental gates $e^{iH\tau}$ for use in gate model quantum computing. Switching gates on and off quickly will let the system represent the full Hamiltonian.

[0037] Instead of Trotterization, reverse annealing may be used in a hybrid computing system where the quantum processor is a quantum annealer. Using a hybrid computing system, for example hybrid computing system **100**, the digital processor can rotate the basis and the resulting states can be evolved backwards by the quantum processor. Examples of reverse annealing are discussed in more detail in International Patent Publication No. WO2017214293A1.

[0038] A Hamiltonian H may be represented as

$$H = T + U \quad (1)$$

where T, the kinetic energy term, is diagonal in the plane wave basis (pw), and U, the potential energy term, is diagonal in the plane wave dual basis (pwd), i.e., the dual of the plane wave basis obtained by a unitary rotation. A basis is then chosen to evolve the quantum processor to an initial state. For example, the plane wave basis may be chosen.

[0039] The quantum processor is programmed with the part of the Hamiltonian that is diagonal in the plane wave basis (T)

$$H_p = T \quad (2)$$

[0040] where H_p is the problem Hamiltonian programmed into the quantum processor. The quantum processor evolves to an initial ground state(s) ψ_0 .

[0041] Then the digital processor rotates the ground state basis from plane wave to plane wave dual, for example via a discrete Fourier transform (DFT)

$$DFT\psi_0 = \sum_i c_{ix} \phi_i \quad (3)$$

[0042] where the coefficients c_{ix} can be complex and negative, and ϕ_i are eigenstates of the new problem Hamiltonian.

[0043] The ϕ_i are used to program the quantum processor for reverse annealing (RA) under the problem Hamiltonian $H_p=U$. The quantum processor evolves backwards until a value s^* of the normalized evolution coefficient. The digital processor reads out new states

$$RA\phi_i = \sum_x a_{ix} \phi_x \quad (4)$$

[0044] where all a_{ix} representing the frequency of the individual samples, are normalized positive real numbers ≤ 1 and x represents the individual samples read out.

[0045] The digital processor may store the coefficients a_{ix} and c_{ix} for future calculations. For example, the ground state after the first evolution $RA(DFT\Psi_0)=\Phi_1$ is approximated as a superposition of the form $\Phi_1=\sum_i c_{i0} \sum_x a_{ix} \phi_x$.

[0046] The digital processor then rotates the states ϕ_x obtained above into the plane basis, for example using an inverse discrete Fourier transform (IDFT).

$$IDFT\phi_x = \sum_x b_{xi} \psi_x \quad (5)$$

[0047] The digital processor programs the quantum processor for reverse anneal with each ψ_x under the problem Hamiltonian $H_p=T$. The quantum processor evolves backwards until s^* and the digital processor reads out new states

$$RA\psi_i = \sum_x a_{ix} \psi_x \quad (6)$$

[0048] The digital processor can then calculate the ground states of the Hamiltonian H using the final ground states wavefunction obtained, either in the plane wave or the plane wave dual basis:

$$H(pw)\Psi_n = \epsilon_n \Psi_n \quad (7)$$

$$H(pwd)\Psi_n = \epsilon_n \Phi_n \quad (8)$$

[0049] FIG. 2 is a flow diagram showing an example method 200 for calculating the ground state of a non-diagonal Hamiltonian with a hybrid computing system. The hybrid computing system comprises a digital processor and

a quantum processor. An example hybrid computing system that can be employed for method 200 is described in greater detail in FIG. 1.

[0050] Method 200 comprises acts 201 to 216; however, a person skilled in the art will understand that the number of acts is an example, and, in some implementations, certain acts may be omitted, further acts may be added, and/or the order of the acts may be changed.

[0051] Method 200 starts at 201, for example in response to a call from another routine.

[0052] At 202, the digital processor receives a Hamiltonian. The Hamiltonian may be non-diagonal and comprising a plurality of terms. The Hamiltonian may represent the electronic structures of periodic lattices, for example homogeneous or uniform electron gas (jellium) or other chemical structures.

[0053] At 203, the digital processor applies a transformation to render the terms of the non-diagonal Hamiltonian diagonal in different bases. For example, a non-diagonal Hamiltonian H will be transformed in a non-diagonal Hamiltonian of the form $H=T+U$, where the two terms T and U are diagonal in either one of two bases. In one implementation T may be diagonal in the plane wave basis and U may be diagonal in the plane wave dual basis, i.e., the dual of the plane wave basis obtained by a unitary rotation.

[0054] Several methods may be applied at act 103. In at least one implementation, methods described in Babbush et al. (see <https://arxiv.org/abs/1706.00023>) may be employed.

[0055] At 204, the quantum processor is programmed to evolve forward in a basis where one term of the Hamiltonian H is diagonal. For example, in one implementation, the quantum processor may be programmed with a problem Hamiltonian $H_p=T$ in the plane wave basis. The quantum processor will reach a ground state ψ_0 .

[0056] At 205, the digital processor applies a transformation to the ground state basis (e.g., the plane wave basis) so that another term of the non-diagonal Hamiltonian (e.g., U) is diagonal in the transformed ground state basis (e.g., the plane wave dual basis). In the case of Hamiltonians with a plurality of terms (e.g., more than two terms), more than one term may be diagonal in the transformed ground state basis.

[0057] In at least one implementation a discrete Fourier transform (DFT) may be applied

$$DFT\psi_s = \sum_i c_{ix} \phi_i \quad (3)$$

[0058] where the coefficients may be complex and/or negative and ϕ_i are eigenstates of the new problem Hamiltonian.

[0059] At 206, the quantum processor is programmed for reverse (backwards) evolution, or reverse anneal (RA), with another term of the non-diagonal Hamiltonian. For example, the quantum processor is programmed with ϕ_i from EQ (3) under the problem Hamiltonian $H_p=U$. The quantum processor evolves backwards until a value s_1^* of the normalized evolution coefficient s . The value s_1^* may be received by the digital processor as part of a set of inputs.

[0060] At 207, the digital processor causes the quantum processor to pause the evolution for a time t . Time t may be received by the digital processor as part of a set of inputs.

[0061] At **208**, the digital processor causes the quantum processor to resume evolving forward, thus reaching a ground state or a set of ground states.

[0062] At **209**, the digital processor reads out the states quantum processor.

$$RA\phi_i = \sum_{\chi} a_{i\chi} \phi_{\chi} \quad (4)$$

[0063] where χ represents the individual samples read out and all $a_{i\chi}$ representing the frequency of the individual samples, may be normalized positive real numbers. The digital processor may store the coefficients c_{ix} and $a_{i\chi}$ in a memory system for future calculations. In at least one implementation, act **209** is optional and method **200** proceeds from **208** to **210**.

[0064] At **210**, the digital processor applies a transformation to the state basis (e.g., the plane wave dual basis) so that another term of the non-diagonal Hamiltonian (e.g., T) is diagonal in the transformed basis (e.g., the plane wave basis). In one implementation an Inverse Fourier transform (IDFT) may be applied

$$IDFT(\phi_{\chi}) = \sum_x b_{x\chi} \psi_x \quad (5)$$

[0065] where the coefficients $b_{x\chi}$ may be complex and/or negative.

[0066] At **211**, the quantum processor is programmed for reverse (backwards) anneal (RA) with another term of the non-diagonal Hamiltonian. For example, the quantum processor is programmed with each ψ_x under the problem Hamiltonian $H_p=T$. The quantum processor evolves backwards until a value s_2^* of the normalized evolution coefficient s . The value s_2^* may be received by the digital processor as part of a set of inputs. In at least one implementation s_2^* is equal s_1^* .

[0067] At **212**, the digital processor causes the quantum processor to pause the evolution for a time t' . Time t' may be received by the digital processor as part of a set of inputs. In at least one implementation, $t=t'$.

[0068] At **213**, the digital processor causes the quantum processor to resume evolving forward, thus reaching a ground state or a set of ground states.

[0069] At **214**, the digital processor reads out the states of the quantum processor

$$RA\psi_i = \sum_{\chi} a_{i\chi} \psi_{\chi} \quad (6)$$

[0070] The digital processor may store the coefficients c_{ix} , $a_{i\chi}$ and $b_{x\chi}$ in a memory system for future calculations.

[0071] At **215**, the digital processor calculates the ground energy of the non-diagonal Hamiltonian H using the final ground state function read out at **214**. For example, if at act **211** the quantum processor is programmed in the plane wave basis (pw), the digital processor performs the calculation in the plane wave basis ($H_{pw}\psi_n=\epsilon_n\psi_n$, EQ (7)), otherwise if at act **211** the quantum processor is programmed in the plane

wave dual basis (pwd), the digital processor performs the calculation in the plane wave dual basis ($H_{pwd}\phi_n=\epsilon_n\phi_n$, EQ(8)).

[0072] Method **200** terminates at **216**, until it is, for example, invoked again.

[0073] FIG. 3 is a flow diagram showing an example iterative method **300** for calculating the ground state of a non-diagonal Hamiltonian with a hybrid computing system. The hybrid computing system comprises a digital processor and a quantum processor.

[0074] Method **300** comprises acts **301** to **317**; however, a person skilled in the art will understand that the number of acts is an example, and, in some implementations, certain acts may be omitted, further acts may be added, and/or the order of the acts may be changed.

[0075] Method **300** starts at **301**, for example in response to call from another routine.

[0076] At **302**, the digital processor receives a Hamiltonian as described above with reference to act **202** of method **200**.

[0077] At **303**, the digital processor applies a transformation to the Hamiltonian received at **302**, as described above with reference to act **103** of method **200**.

[0078] At **304**, the quantum processor is programmed to evolve forward, as described above with reference to act **204** of method **200**.

[0079] At **305**, the digital processor applies a transformation to the state basis, as described above with reference to act **205** of method **200**.

[0080] At **306**, the quantum processor is programmed for reverse (backwards) anneal, as described above with reference to act **206** of method **200**.

[0081] At **307**, the digital processor causes the quantum processor to pause, as described above with reference to act **207** of method **200**.

[0082] At **308**, the digital processor causes the quantum processor to resume evolving forward, as described above with reference to act **208** of method **200**.

[0083] At **309**, the digital processor reads out the states of the quantum processor, as described above with reference to act **209** of method **200**. In at least one implementation, act **309** is optional and method **300** proceeds to **310**.

[0084] At **310**, the digital processor applies a transformation to the state basis, as described above with reference to act **210** of method **200**.

[0085] At **311**, the quantum processor is programmed for reverse (backwards) anneal, as described in act **211** of method **200**.

[0086] At **312**, the digital processor causes the quantum processor to pause the evolution, as describe above with reference to act **212** of method **200**.

[0087] At **313**, the digital processor causes the quantum processor to resume evolving forward, as describe above with reference to act **213** of method **200**.

[0088] At **314**, the digital processor reads out the states quantum processor, as described in act **214** of method **200**.

[0089] At **315**, the digital processor determines whether an exit condition has been met. In one implementation the exit condition is the completion of a determined number of iterations. In another implementation the exit condition is determining that an error factor is lower than an established threshold. In yet another implementation, the exit condition is the first to occur of: i) determining that an error factor is lower than an established threshold, and ii) completion of a

determined number of iterations. If the exit condition is not met, method **300** continues to act **305**. Otherwise to **316**.

[0090] At **316**, the digital processor calculates the ground energy of the non-diagonal Hamiltonian, as described above with reference to act **215** of method **200**.

[0091] Method **300** terminates at **317**, until it is, for example, invoked again.

[0092] Method **200** and **300** may be employed to calculate the ground energy of non-diagonal Hamiltonians. In one implementation, method **200** and/or may be employed to simulate electronic structures of periodic lattices, for example homogeneous or uniform electron gas (jellium) or other chemical structures.

[0093] FIG. 4 is a chart showing an example evolution **400** of an analog processor performed during an example implementation of method **200** of FIG. 2 or method **300** of FIG. 3. An analog processor may be a quantum processor.

[0094] Vertical axis **401** represents the normalized evolution coefficient s and horizontal axis **402** represents the time of the evolution of the analog processor. The normalized evolution coefficient s may represent the normalized flux applied to a compound Josephson junction or the normalized persistent current I_p of a flux qubit. The normalized evolution coefficient s changes monotonically over time, increasing from 0 to a maximum value of 1. A person skilled in the art will understand that the rate of change of the normalized evolution coefficient over time is shown in FIG. 4 for illustration purposes only and in other implementations the normalized evolution coefficient can increase at a slower or faster rate. In some implementations the normalized evolution coefficient s can change non-linearly. Examples of evolution schedules of analog processors are described in great details in Patent Publication No. US20180330264A1, which is here incorporated by reference. Techniques described herein are used to operate a hybrid processor comprising an analog processor and a digital processor where the normalized evolution coefficient s may increase and/or decrease over the course of the operation of the hybrid processor.

[0095] Before the start of example evolution **400** the hybrid processor may determine a classical spin state and apply preparatory bias to the analog processor to target the evolution of the analog processor towards the classical spin state. Preparatory bias may be applied via the analog processor's circuitry components, for example via on-chip DACs or analog lines. Preparatory bias may influence the evolution of the analog processor towards a classical state. When the analog processor is a quantum processor with n qubits, there are 2^n classical states.

[0096] With reference to method **300** of FIG. 3, the normalized evolution coefficient s increases from 0 to 1 until time t_1 as described with reference to act of method **300**.

[0097] From t_1 to t_2 the evolution of the analog processor resumes in a direction opposite the direction before time t_1 , i.e., backwards and the normalized evolution coefficient s decreases from 1 to a value s^* , as described before with reference to act **306** of method **300**. The digital processor may determine the value of s^* before the start of example evolution **400**. A person skilled in the art may understand that a time interval (not shown in FIG. 4) may occur before the evolution of the analog processor resumes at t_1 .

[0098] Where the analog processor is a quantum processor, the energy barriers of the qubits are lowered until an intermediate transverse field and/or tunneling energy is

reached. The intermediate transverse field and/or tunneling energy may be determined by the digital processor.

[0099] After time t_2 the evolution of the analog processor is paused for a time interval t_2-t_3 (also indicated in FIG. 4 as t for simplicity). The time interval t may be determined by the digital processor before the start of example evolution **400**.

[0100] A person skilled in the art will recognize that the rate of change of the normalized evolution coefficient between 0 and time t_1 may be the same as the rate of change between t_1 and t_2 or it may be different. The digital processor may determine the rate of change of s .

[0101] After time interval t , the evolution of the analog processor resumes in the same direction as the evolution from 0 to time t_1 , i.e., the normalized evolution coefficient s increases from value s^* to 1 until the analog processor reaches a classical spin state at t_4 . Where the analog processor is a quantum processor the digital processor may raise the energy barriers of the qubits to reach a classical spin state. At t_4 the digital processor may read out this classical spin state, as described with reference to act **309** of method **300**. The time interval during which the digital processor reads out the state of the analog processor is not indicated in FIG. 4.

[0102] After time t_4 the evolution of the analog processor resumes in an opposite direction until a value s^* of the normalized evolution coefficient at time t_5 and then paused until time t_6 , as described above with reference to acts **311** and **312** of method **300**.

[0103] A person skilled in the art will recognize that the analog process may be programmed to repeat this evolution pattern one or more time as determined by the digital processor, as described above with reference to act **315** of method **300**.

[0104] The above described method(s), process(es), or technique(s) could be implemented by a series of processor readable instructions stored on one or more non-transitory processor-readable media. Some examples of the above described method(s), process(es), or technique(s) method are performed in part by a specialized device such as an adiabatic quantum computer or a quantum annealer or a system to program or otherwise control operation of an adiabatic quantum computer or a quantum annealer, for instance a computer that includes at least one digital processor. The above described method(s), process(es), or technique(s) may include various acts, though those of skill in the art will appreciate that in alternative examples certain acts may be omitted and/or additional acts may be added. Those of skill in the art will appreciate that the illustrated order of the acts is shown for example purposes only and may change in alternative examples. Some of the example acts or operations of the above described method(s), process(es), or technique(s) are performed iteratively. Some acts of the above described method(s), process(es), or technique(s) can be performed during each iteration, after a plurality of iterations, or at the end of all the iterations.

[0105] The above description of illustrated implementations, including what is described in the Abstract, is not intended to be exhaustive or to limit the implementations to the precise forms disclosed. Although specific implementations of and examples are described herein for illustrative purposes, various equivalent modifications can be made without departing from the spirit and scope of the disclosure, as will be recognized by those skilled in the relevant art. The

teachings provided herein of the various implementations can be applied to other methods of quantum computation, not necessarily the example methods for quantum computation generally described above.

[0106] The various implementations described above can be combined to provide further implementations. All of the commonly assigned US patent application publications, US patent applications, foreign patents, and foreign patent applications referred to in this specification and/or listed in the Application Data Sheet are incorporated herein by reference, in their entirety, including but not limited to U.S. provisional Patent Application No. 62/661,986; U.S. Patent Publication No. US20180330264A1; and International Patent Publication No. WO2017214293A1.

[0107] These and other changes can be made to the implementations in light of the above-detailed description. In general, in the following claims, the terms used should not be construed to limit the claims to the specific implementations disclosed in the specification and the claims, but should be construed to include all possible implementations along with the full scope of equivalents to which such claims are entitled. Accordingly, the claims are not limited by the disclosure.

1. A method of operating a hybrid computing system to calculate the ground state of a non-diagonal Hamiltonian, the hybrid computing system comprising a quantum processor and a digital processor, wherein the non-diagonal Hamiltonian comprises a first term and a second term, the first term diagonal in a first basis and the second term diagonal in a second basis, the second basis different from the first basis, the method comprising:

- programming the quantum processor to evolve forwards with the first term in the first basis to a first ground state;
- applying a first transformation to the first basis;
- programming the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ;
- causing the evolution of the quantum processor to pause for a first period of time during the evolving backwards with the second term in the second basis;
- applying a second transformation to the second basis;
- programming the quantum processor to evolve backwards with the first terms in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ;
- causing the evolution of the quantum processor to pause for a second period of time during the evolving backwards with the first term in the first basis;
- programming the quantum processor to evolve forwards with the first term in the first basis to a second ground state;
- reading out the second ground state; and
- using the second ground state to calculate the ground energy of the non-diagonal Hamiltonian.

2. The method of claim 1 wherein applying a second transformation to the second basis includes applying to the second basis a second transformation that is the inverse of the first transformation.

3. The method of claim 2 wherein applying a first transformation to the first basis includes applying a discrete Fourier transform to the first basis.

4. The method of claim 1 wherein applying a first transformation to the first basis includes applying a first transformation to the first basis of a non-diagonal Hamiltonian

that includes a plurality of terms and applying a first transformation to the first basis includes applying a first transformation to the first basis so that more than one term of the non-diagonal Hamiltonian becomes diagonal in the first basis.

5. The method of claim 1 further comprising programming the quantum processor to evolve forward with the second term in the second basis to a third ground state after pausing the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis; and reading out the third ground state.

6. The method of claim 1 wherein programming the quantum processor to evolve forwards with the first term in the first basis to a first ground state includes programming the quantum processor to evolve forwards with the first term in the first basis to a first set of ground states; and further comprising programming the quantum processor to evolve forwards with the first term in the first basis to a second set of ground states before reading out the second set of ground states.

7. The method of claim 1 wherein programming the quantum processor to evolve backwards with the first term in the first basis until a value s_2^* of the normalized evolution coefficient s includes programming the quantum processor to evolve backwards with the first term in the first basis until a value s_2^* of the normalized evolution coefficient s is equal to s_1^* .

8. The method of claim 1 wherein calculating the ground energy of the non-diagonal Hamiltonian includes calculating the ground energy of a non-diagonal Hamiltonian describing the electronic structure of a period lattice.

9. The method of claim 1 further comprising until an exit condition is met, iteratively repeating:

- applying a first transformation to the first basis;
- programming the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ;
- causing the evolution of the quantum processor to pause for a first period of time during the evolving backwards with the second term in the second basis;
- applying a second transformation to the second basis;
- programming the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ;
- causing the evolution of the quantum processor to pause for a second period of time during the evolving backwards with the first term in the first basis;
- programming the quantum processor to evolve forwards with the first term in the first basis to a n^{th} ground state; and
- reading out the n^{th} ground state.

10. The method of claim 8 further comprising determining whether an exit condition has been met.

11. The method of claim 10 wherein determining whether an exit condition has been met includes determining at least one of: whether an error condition is below an error threshold and whether a number of evolution iterations has been completed.

12. A hybrid computing system, the hybrid computing system comprising a quantum processor and a digital processor, the digital processor operable to:

- program the quantum processor with a first term of a non-diagonal Hamiltonian, the non-diagonal Hamiltonian

nian comprising the first term and a second term, the first term diagonal in a first basis and the second term diagonal in a second basis different from the first basis, to evolve forwards in the first basis to a first ground state;

apply a first transformation to the first basis;

program the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient s ;

pause the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis;

apply a second transformation to the second basis;

program the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient s ;

pause the evolution of the quantum processor for a second period of time during the evolving backwards with the first term in the first basis;

program the quantum processor to evolve forwards with the first term in the first basis to a second ground state;

read out the second ground state; and

use the second ground state to calculate the ground energy of the non-diagonal Hamiltonian.

13. The hybrid computing system of claim **12** wherein the second transformation is the inverse of the first transformation.

14. The hybrid computing system of claim **13** wherein the first transformation is a discrete Fourier transform.

15. The hybrid computing system of claim **12** wherein the non-diagonal Hamiltonian includes a plurality of terms and the digital processor is operable to apply a first transformation to the first basis so that more than one term of the non-diagonal Hamiltonian becomes diagonal.

16. The hybrid computing system of claim **12** wherein the digital processor is further operable to program the quantum processor to evolve forward with the second term in the

second basis to a third ground state after pausing the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis; and to read out the third ground state.

17. The hybrid computing system of claim **12** wherein the digital processor is further operable to program the quantum processor to evolve forwards in the first basis to a first set of ground states; and to cause the quantum processor to evolve forwards in the first basis to a second set of ground states before reading out the second set of ground states.

18. The hybrid computing system of claim **12** wherein s_2^* is equal to s_1^* .

19. The hybrid computing system of claim **12** wherein the non-diagonal Hamiltonian describes the electronic structure of a period lattice.

20. The hybrid computing system of claim **12** wherein the digital processor is further operable to, until an exit condition is met, iteratively repeatedly:

apply a first transformation to the first basis;

program the quantum processor to evolve backwards with the second term in the second basis until reaching a value s_1^* of the normalized evolution coefficient;

pause the evolution of the quantum processor for a first period of time during the evolving backwards with the second term in the second basis;

apply a second transformation to the second basis;

program the quantum processor to evolve backwards with the first term in the first basis until reaching a value s_2^* of the normalized evolution coefficient;

pause the evolution of the quantum processor for a second period of time during the evolving backwards with the first term in the first basis;

program the quantum processor to evolve forwards with the first term in the first basis to a n^{th} ground state; and read out the n^{th} ground state.

* * * * *