Section 10.2: Introduction to Quantum Computing

(under construction)

The size of electronic components and the systems continue to decrease. Thus far, these components generally obey the laws of classical physics. Inevitably, the reduced sizes will require new quantum operating principles. This translates to new operating principles for computers as well. The new principles must address and incorporate the ultimate probabilistic nature of the elementary particle.

Quantum computing is an interdisciplinary endeavor. It encompasses theoretical computer sicence, physics and engineering. The reader will find a wealth of information and simulation software in the book "Explorations in Quantum Computing" by C. P. Williams and S. H. Clearwater with over 300 references.

Topic 10.2.1: Turing Machines

The Turing machine originated as a conceptual means to reduce mathematical proofs to a mechanical computation. The results apply to all modern machines regardless of size and speed.

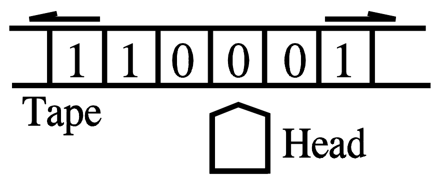

Figure 10.2.1: The classical Turing machine.

The classical (deterministic) Turing machine consists of a "tape" as a type of memory that moves forward and backwards across a read-write head as shown in Figure 10.2.1. The tape contains 0s and 1s arranged in sequential order. These bits can represent program steps or data bits. The head has the responsibility to interpret the "meaning" of the bits. For example, if the head is in such a "state" that it must read in 8 data bits then it will interpret the next sequence of 8 bits as data. The history of the calculations performed and the program steps executed determine the "state of the head" which gives meaning to the sequence of bits on the tape. With these machines, there is a tradeoff between computational accuracy and the length of time required to perform a calculation.

The probabilistic Turing Machine differs from the classical one in that the machine can assume a number of possible states controlled by probability. Basically, the result represents a possible path through a calculation as controlled by a probability distribution. The resulting state will be a probability that is the sum of probabilities for all possible past states. However, only one path is actually followed which distinguishes this machine from its quantum counterpart. Any problem solvable on the probabilistic Turing machine can also be solved on the classical one (and vice versa).

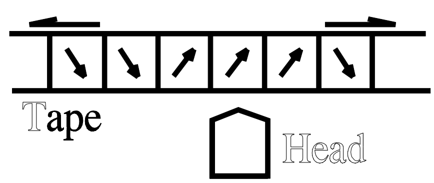

Figure 10.2.2: Bennett's orignal quantum Turing machine replaces bits with quantum bits characterized by a 2-D Hilbert space.

The quantum Turing machine replaces the "bit" with "quantum bits" (qbits). The qbits most often represent quantum properties that can assume two possible configurations although an observable with any number of discrete states will work. For the purposes of this chapter, we envision the qbit as representing the up or down spin of an electron confined to a trap. When the electron occupies the "up state" denoted by $|0\rangle$

then this will correspond to a logical 0 or false. The down state, denoted by $|1\rangle$, represents the logical 1 or true. The bits can encode a range of values between 0 and 1 since the actual quantum mechanical state of the spin particle can have the form $|\psi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$ where β_i represents a complex number.

The original quantum Turing machine considered the head to be interacting with a given qbit for a fixed period time but leaves it in a collapsed state (i.e., in one of the basis states $|0\rangle$ or $|1\rangle$). The quantum Turing machine attempts to use the fact that an electron will sample all possible trajectories through Hilbert space similar to the idea behind the Feynman path integral only applied to spin space in this case. Therefore, the particle reaches time t bearing the influence of all possible paths represented by a superposition of basis states. Making an observation forces the particle wave function to collapse to one of the basis states with a probability determined by its history. This process hasn't any classical analog.

In the section on the relation between linear algebra and quantum theory, we discuss the collapse of the wave function. The quantum mechanical system without outside influences and observers evolves according to the dynamics in the Schrodinger equation. This evolution causes the system, perhaps initially in an energy basis state, to evolve to some superposition of the basis states. We view the particle as simultaneously in these states. Making an observation on the system causes the wave function to instantaneously and randomly collapse to one of the basis sets without following the evolution described by the Schrodinger Equation. Making such an observation is the same as "checking the answer" from the computer. So long as we don't check for an answer, the quantum computer can be reversed at any time since the evolution operator $\hat{U} = e^{\frac{\hat{\mathscr{H}} t}{i\hbar}}$ is unitary so that

$$|\psi(t)\rangle = \hat{U}|\psi(0)\rangle \quad \Leftrightarrow \quad |\psi(0)\rangle = \hat{U}^{+}|\psi(t)\rangle \tag{10.2.1}$$

The original Turing machine only allows the qbit to evolve according to the evolution operator only during the time that the head interacts with it. Therefore this machine could not make full use of the ability of the electron to make large superpositions with many different qbits.

Topic 10.2.2: Block Diagrams for the Quantum Computer

We now fix our ideas on how a quantum computer might physically appear. In classical computer, logic gates have an input and an output. The input signal might come from a register of bits. The output usually comes to a separate location as transformed bits. Applying this classical view to the quantum gate results in Figure 10.2.3. In this case, the gate transforms the qbit into another separate qbit. Several present designs for the quantum computer do not allow for this capability. In fact, a register of qbits might be pictures as a series of electrons confined to traps. The quantum computer has an input starting with this register of qbits and an output ending with these qbits. A scheme similar to Figure 10.2.3 might become viable for the quantum computer if the teleportation technology becomes viable. This

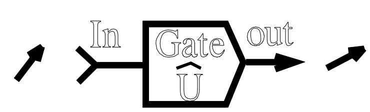

Figure 10.2.3: Classical view of a quantum gate.

technology might one-day be able to extract all of the quantum information from a particle, modify and transmit the information through a quantum gate, and reconstruct the state at a new location.

For now, we use a register consisting of spin particles. We design a Hamiltonian to evolve the spins. The Hamiltonian represents the program. An interaction begins at t=0 and evolves the qbit in time according to the evolution operator

$$\hat{U}(t) = e^{\frac{\mathscr{H} t}{i\hbar}} \tag{10.2.2}$$

This form of the evolution operator requires a closed system. A time independent Hamiltonian therefore represents a type of "hardwired" gate. In order to change the programming, the Hamiltonian would need to depend on time and the evolution operator would use the time ordered product discussed in the quantum mechanical representation theory.

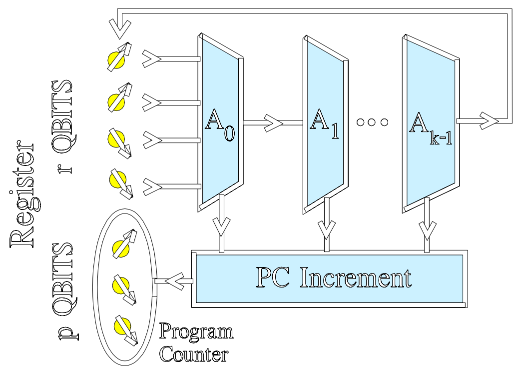

Figure 10.2.4: Idea behind the Feynman processor. In actuality, the depicted gates are part of the Hamiltonian. The evolution operator actually operates on the register.

The Feynman processor uses a closed system. The design starts with logic operations. In order to determine when to stop the processor, the register of qbits is divided into two sections. The r-qbits make up the data and the p-qbits serve as a program step counter. The r-qbits (r=register, number of bits = r) store the data and interact with the processor in parallel fashion. The p-qbits (p=program counter, number of bits = p = k+1) keep track of the number of steps that the computer has executed. The number of p-qbit corresponds to the number of "gates" in Figure 10.2.4 (plus one). When the cursor resides in the k+1 qbit then the calculation is complete.

The Feynman computer cannot be reprogrammed once the circuitry has been set since it uses the time independent Hamiltonian. Figure 10.2.4 sets the basic computer architecture. Once having decided on the computation to be performed, the basic block diagram can be laid out using quantum gates. The machine performs the function $\hat{A}_{k-1}\hat{A}_{k-2}\cdots\hat{A}_1\hat{A}_0$. Next, the Hamiltonian and evolution operator can be calculated. For a closed system, the product is implemented using a Hamiltonian of the form

$$\hat{H} = \frac{1}{2}\sum_{i=0}^{k-1}\left[\hat{a}_{i+1}^{+}\hat{a}_i\hat{A}_i + \left(\hat{a}_{i+1}^{+}\hat{a}_i\hat{A}_i\right)^{+}\right] \tag{10.2.3}$$

where $\hat{a}^{+}, \hat{a}$ represent creation and annihilation operators, respectively. The adjoint operator appears in Equation 10.2.3 to ensure the Hamiltonian is Hermitian. Each operator acts on a separate Hilbert space and therefore the products must be direct products. The creation and annihilation operators change the state of the program counter. Once knowing the number of gates, the number p-qbits can be determined. Once the mechanics have been built, we can initialize the data in the r-qbits (i.e. memory register) and let the computer run. We periodically check the p-qbits until the (k+1) qbit sets and we then read off the answer from the memory register.

Alternate version of the quantum computer can be envisioned. One radically different model uses the Feynman path integral for coordinate space rather than for the configuration space used above. A person might imagine an electron entering a region of space with a number of obstacles and gates. The Feynman path integral indicates that the electron arriving at the output of the box, must carry with it information from all possible paths through the box. By an appropriate choice of "innards" (i.e., interactions), the resulting electron will carry the results of a computation. One advantage of this scheme would be that the "box" could be reduced to 100s of Angstroms and the computer would have separate inputs and outputs.

We continue with the Feynman computer in what follows.

Topic 10.2.3: The Memory Register with Multiple Spins

In this section, we model the memory qbits after 2-state spin but realize that memory can be implemented using any number quantized levels. Rather than use the notation of $|1\rangle, |2\rangle$ for spin up and spin down, we use $|0\rangle, |1\rangle$ as a reminder of logic 0 and logic 1, respectively. The superposition wave function has the form

$$\left|\psi^{(1)}\right\rangle = \beta_0^{(1)} |0\rangle^{(1)} + \beta_1^{(1)} |1\rangle^{(1)} \quad \to \quad \underline{\psi}^{(1)} = \begin{pmatrix} \beta_0^{(1)} \\ \beta_1^{(1)} \end{pmatrix}$$

$$\left|\psi^{(2)}\right\rangle = \beta_0^{(2)} |0\rangle^{(2)} + \beta_1^{(2)} |1\rangle^{(2)} \quad \to \quad \underline{\psi}^{(2)} = \begin{pmatrix} \beta_0^{(2)} \\ \beta_1^{(2)} \end{pmatrix}$$

The linear algebra shows that the direct product of the two wave functions has the form

$$\left|\psi^{(1)}\right\rangle \otimes \left|\psi^{(2)}\right\rangle = \beta_0^{(1)}\beta_0^{(2)} |0\rangle^{(1)} |0\rangle^{(2)} + \beta_1^{(1)}\beta_0^{(2)} |1\rangle^{(1)} |0\rangle^{(2)} + \beta_0^{(1)}\beta_1^{(2)} |0\rangle^{(1)} |1\rangle^{(2)} + \beta_1^{(1)}\beta_1^{(2)} |1\rangle^{(1)} |1\rangle^{(2)}$$

which produces the matrix

$$\underline{\psi} = \underline{\psi}^{(1)} \otimes \underline{\psi}^{(2)} = \begin{pmatrix} \beta_0^{(1)} \\ \beta_1^{(1)} \end{pmatrix} \otimes \begin{pmatrix} \beta_0^{(2)} \\ \beta_1^{(2)} \end{pmatrix} = \begin{pmatrix} \beta_0^{(1)}\beta_0^{(2)} \\ \beta_0^{(1)}\beta_1^{(2)} \\ \beta_1^{(1)}\beta_0^{(2)} \\ \beta_1^{(1)}\beta_1^{(2)} \end{pmatrix}$$

The basis vectors for the direct product space becomes

$$|00\rangle = |0\rangle^{(1)} |0\rangle^{(2)} \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = |0\rangle^{(1)} |1\rangle^{(2)} \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = |1\rangle^{(1)} |0\rangle^{(2)} \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle = |1\rangle^{(1)} |1\rangle^{(2)} \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

In general, can write a sequence of memory qbits as $|011010001...\rangle$ where each location in the ket corresponds to a different spin particle. We anticipate the basis vector $|\cdots b_3 b_2 b_1 b_0\rangle$ produces a 1 in location $\sum_{n=0}^{N-1} 2^n b_n = \cdots 2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0$.

For multiple spins that interact with each other (or other multiple systems that interact with each other), the wave function becomes a coherent state that cannot be factored. Any measurement will destroy the state. These are entangled states. Classical computing does not incorporate this feature.

Topic 10.2.4: The Feynman Computer for Negation without a Program Counter

One of the simplest examples of the Feynman computer calculates the "negation" of an input bit as shown in Figure 10.2.5. For this example, we do not include the program counter in order to make the computation machinery as evident as possible. We compute the negation of a single qbit initially assumed to be in a zero state $|0\rangle$ corresponding to spin up. The book "Explorations in Quantum Computing" by C. P. Williams and S. H. Clearwater discusses the case of $\sqrt{\text{NOT}}$ as a purely quantum mechanical operation and provides references for Feynman's two bit adder.

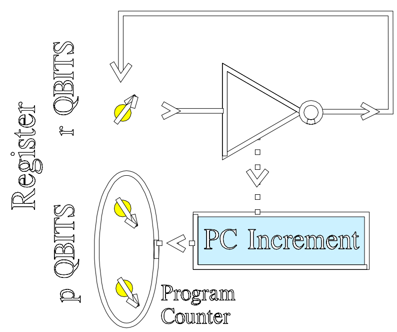

Figure 10.2.5: The Feynman processor for calculating the "NOT" of a qbit.

The "not" operator has the form $\hat{N} = |1\rangle\langle 0| + |0\rangle\langle 1|$ which should be recognized as the Pauli x-component spin operator $\hat{\sigma}_x$. The Hamiltonian in Equation 10.2.3 reduces to

$$\hat{H} = \frac{1}{2}\left(\hat{\sigma}_x + \hat{\sigma}_x^+\right) = \hat{\sigma}_x \tag{10.2.4}$$

since we only need the single "NOT" gate defined by $\hat{\sigma}_x$ which is already Hermitian. We do not include the Planck's constant $\hbar$. The unitary operator in Equation 10.2.2 becomes

$$\hat{U}(t) = e^{-i\mathscr{H}t} = e^{-i\hat{\sigma}_x t} \tag{10.2.5a}$$

We can see that this operator rotates an up spin to a down spin in Hilbert space by making a Taylor series expansion and using

$$\underline{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{10.2.5b}$$

Expanding the evolution operator gives

$$\underline{U}(t) = e^{-i\underline{\sigma}_x t} = 1 + \frac{(-i)}{1!}\underline{\sigma}_x t + \frac{(-i)^2}{2!}\underline{\sigma}_x^2 t^2 + \frac{(-i)^3}{3!}\underline{\sigma}_x^3 t^3 + \cdots$$

Next, separate the real and imaginary parts and note

$$\underline{\sigma}_x^n = \begin{cases} \underline{1} & n = \text{even} \\ \underline{\sigma}_x & n = \text{odd} \end{cases} \tag{10.2.6}$$

to find

$$\underline{U}(t) = e^{-i\underline{\sigma}_x t} = \underline{1}\left(1 - \frac{1}{2!}t^2 + \cdots\right) - i\underline{\sigma}_x\left(\frac{1}{1!}t - \frac{1}{3!}t^3 + \cdots\right) = \underline{1}\text{Cos}(t) - i\underline{\sigma}_x\text{Sin}(t) \tag{10.2.7}$$

Now if we could monitor the progress of the interaction, we would find that near $t = \pi/2$

$$\underline{U}\left(\frac{\pi}{2}\right)\begin{pmatrix}1\\0\end{pmatrix} = -i\underline{\sigma}_x\begin{pmatrix}1\\0\end{pmatrix} = -i\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = -i\begin{pmatrix}0\\1\end{pmatrix} \tag{10.2.8}$$

which shows the qbit is inverted apart from an unimportant phase factor -i.

We can show how this inverter can be physically implemented. The discussion of spin from Chapter 3 shows the spin Hamiltonian has the form

$$\hat{\mathscr{H}}_s = -\frac{q}{m}\vec{B}\cdot\hat{\vec{S}} = \mu_B\vec{B}\cdot\hat{\vec{\sigma}}$$

We want the x Pauli spin matrix to appear in the unitary operator, so choose the magnetic field to point along the positive x-direction

$$\hat{\mathscr{H}}_s = \mu_B\vec{B}\cdot\hat{\vec{\sigma}} = \mu_B B_x\hat{\sigma}_x \tag{10.2.9}$$

The unitary operator in Equation becomes

$$\hat{U}(t) = e^{\frac{\hat{\mathscr{H}}t}{i\hbar}} = \text{Exp}\left\{\frac{\mu_B B_x\hat{\sigma}_x t}{i\hbar}\right\} \tag{10.2.10}$$

where we now work with a physical Hamiltonian and we keep Planck's constant. We expand the exponential using the results from Equation 10.2.7 with

$$t \to \frac{\mu_B B_x t}{\hbar}$$

Therefore Equation 10.2.7 becomes

$$\underline{U}(t) = \underline{1}\text{Cos}\left(\frac{\mu_B B_x t}{\hbar}\right) - i\underline{\sigma}_x\text{Sin}\left(\frac{\mu_B B_x t}{\hbar}\right) \tag{10.2.11}$$

When

$$\frac{\mu_B B_x t}{\hbar} = \frac{\pi}{2} \quad \to \quad t = \frac{\pi\hbar}{2\mu_B B_x} \tag{10.2.12}$$

we find the spin has flipped. Notice that we can control the rate at which the spin flips by adjusting the magnitude of the magnetic field.

Figure 10.2.6 shows why the magnetic field B_x causes the spin to flip. The external magnetic field produces a torque on the spin particle in order to align the two magnetic fields. The Hamiltonian does not include any damping. From a classical point of view, the spin will overshoot the lowest energy configuration and point downward at the time given in Equation 10.2.12. If left to itself, the spin would return to its original configuration. The process explains the Sine and Cosine in Equation 10.2.11.

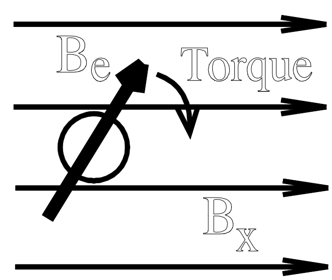

Figure 10.2.6: The external field causes the spin to flip.

Topic 10.2.5: Notes on the Feynman Computer with a Program Counter

Topic 10.2.6: Physical Realizations of Quantum Computers

We now very briefly summarize several physical implementations of quantum computers and logic gates. The interested reader can find in-depth information in the Nielsen and Chuang book "Quantum Computation and Quantum Information," published by Cambridge University Press in 2000. An abbreviated version appears in the Willams and Clearwater book "Explorations in Quantum Computing," published by Springer in 1997. Also check the references in these books. We briefly present the heteropolymer-based, Ion-trap based, QED-based, and NMR based computers.

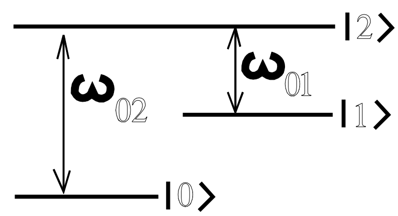

Figure 12.2.7: The three level atom with the angular frequency ω given by the relation $E = \hbar\omega$

The heteropolymer-based computer uses an array of atoms for the memory register. The atoms have three levels as shown in Figure 12.2.7. The ground state $|0\rangle$ is stable. The highest state $|2\rangle$ decays rapidly to either the ground state $|0\rangle$ or to the metastable first excited state $|1\rangle$. A pulse of light with center optical frequency of ω_{02} will transition an electron to state $|2\rangle$. The excited electron can decay to either state $|0\rangle$ or state $|1\rangle$. Whether it relaxes back $|0\rangle$ by stimulated or spontaneous processes during the duration of the laser pulse, then it will likely absorb enough energy to make a transition back to state $|2\rangle$. This three level arrangement is actually considered to be two levels since $|2\rangle$ decays so rapidly.

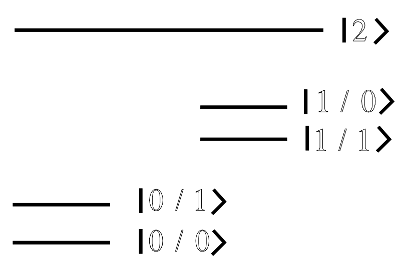

Figure 12.2.8: The energy levels for atom A given the state of atom B.

Adjacent atoms (say A,B) affect the energy levels of each other through an electric dipole interaction. Figure 12.2.8 shows how the state of atom B affects the states of atom A. The notation $|A/B\rangle$ refers to the state of A given the state of B. Notice the state of atom B shifts the energy of $|0\rangle$ and $|1\rangle$ with respect to $|2\rangle$. The frequency of the light required to induce a transition to level $|a/b\rangle$ to $|a'/b\rangle$ is denoted by $\omega_{a'a}^{B=b}$. The frequency of light controls the operation of the device and represents the program. For example, we can make a controlled inverter. Suppose B=1 then an electron in state A=0 will make a transition to A=1 when $\omega = \omega_{02}^{B=1}$. However, if B=0 then the same process cannot occur. The sequence of pulses determines the overall function of the computer.

The ion trap computer uses lasers to excite atoms in a well. NIST made the wells from RF waves rather than atomic barriers. These wells have parabolic shape and the well levels (restricted to 2) can encode a qbit. Additionally NIST encoded as second qbit in the energy levels of the valence electron. The scheme worked 90% of the time.

Interaction between neighboring atoms can produce a type of bus to carry the quantum information from one location to another. Other groups have considered a range of atoms and have discovered Yb would have a long enough lifetime to factor 385 bits.

The Cal-Tech QED-based (photonic) computer implements an XOR function. Figure 12.2.9 shows the gate. The target bit consists of linearly polarized photons, which can be decomposed into right and left circularly polarized components. The control bit is circularly polarized. On average, only a single control bit, target bit and cesium atom occupy the cavity at any time. The cavity resonant frequency matches the cesium transition energy and the energy of the two photons. The control and target bits interact with a cesium atom in a cavity. The phase of the shift of one component of the linearly polarized target bit depends on the atomic excitation and upon the polarization (right or left) of the control photon.

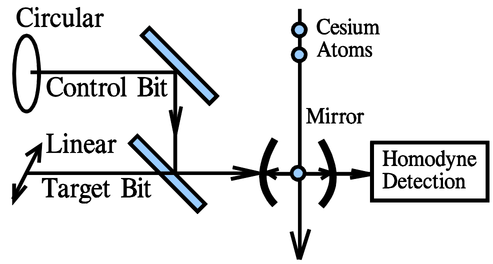

Figure 12.2.9: A block diagram of the QED-based computer.

The nuclear magnetic resonance NMR computer uses the spin of the nucleus. The large number of nuclei in a molecule along with the large number of molecules means that the answer occurs as an ensemble average. The state of the nucleus can be read-out by observing an the NMR spectrum. The shift of the resonance peak corresponds to a change of state in the spin.