Лекция 1. Введение.

1. Что такое квантовая информация?
2. Закон Мура, роль квантовых эффектов. Биты и их реализация. Регистры. Понятие машины Тьюринга. Классические вычисления. Логические операции. Сложение по модулю 2.
3. Требования, предъявляемые к квантовому компьютеру. Основные проблемы на пути к его созданию.

1. *Квантовая информация* - это новая область науки и технологии, сочетающая в себе разделы физики, математики, кибернетики и инженерии. Ее целью является выяснение роли фундаментальных законов физики, открытых в XX-ом веке в процессах получения, передачи и обработки информации. Теория классической информации не может адекватно ответить на вопрос, как информация может быть использована в реальном (физическом) мире - т.е. в квантовом мире. Некоторые выводы теории квантовой информации могут быть представлены как обобщение классической теории в тех случаях, когда информация передается и хранится с помощью квантовых состояний, а не в терминах классических битов.

Вычисление - это процесс, в ходе которого происходит определенное для каждой логической операции (ЛО) нелинейное взаимодействие потоков информации друг с другом и их преобразование. В зависимости от типа ЛО определенным образом изменяется состояние логического элемента (ЛЭ), а поступающая на его входы информация либо передается далее, либо как-то преобразуется. Управление или преобразование происходит под воздействием внешних сигналов. Это, например, переключение или инверсия ($0\Rightarrow1$, $1\Rightarrow0$), запись, сброс. Носитель информации на физическом уровне называется сигналом.

2. *Закон Мура*. - это эмпирический закон, согласно которому число транзисторов в кристалле одной интегральной схемы в течение первых 15 лет удваивалось каждый год, а затем и до сих пор такое удвоение происходит за 1.5 года. Если первые кремниевые микросхемы имели размеры элементов в плоскости кристалла порядка десятков микрон, то современные образцы

характеризуются размерами порядка 100нм, а контроль осуществляется с точностью порядка 10нм. Согласно закону Мура менее чем через 20 лет размеры интегральной схемы станут порядка атомных, а следовательно, законы их функционирования будут определяться законами микромира, т.е. квантовой механикой. Общеизвестно, что объекты микромира ведут себя совершенно необычно с точки зрения классического мира. Так, наблюдение за атомом возмущает его движение, в то же время в отсутствие наблюдения, атом как бы размыт по пространству и скоростям (отсутствие траектории - соотношение неопределенности Гайзенберга), как будто бы он находился бы в нескольких различных местах в одинаковые моменты времени.

Таким образом до сих пор квантовые эффекты, связанные с малостью размеров различных устройств воспринимались как преграда на пути к миниатюризации электронных устройств. Квантовая информатика должна выяснить как использовать фундаментальные квантовые свойства.

К настоящему времени, пожалуй, единственным приложением квантовой информатики является криптография. Здесь уже разработаны и реализованы алгоритмы, использующие свойства квантовых объектов (неклонируемость и невозможность измерения без возмущения). Основной выигрыш в квантовых криптографических протоколах - даже не абсолютная их секретность (в классической криптографии существуют безусловно секретные ключи), а то, что сам факт подслушивания становится известным для пользователей!

Итак, **Проблема 1** -уменьшение размеров интегральных схем, т.е. отдельных элементов. Нанотехнологии. Естественный предел здесь - характерный масштаб атома, когда вступают в силу законы микромира, т.е. квантовой механики.

Проблема 2- уменьшение доли рассеиваемой энергии. Логически обратимые операции - те, которые не сопровождаются рассеянием энергии (Ландауер, 1961г.). Универсальный цифровой компьютер типа вычислительной машины Тьюринга может быть построен на логически и термодинамически обратимых ЛЭ так, что энергия будет рассеиваться только за счет необратимых периферийных процессов (типа ввода информации в машину либо ее вывода). Типичные классические обратимые универсальные ЛЭ - это ЛЭ Тоффоли и Фредкина.

Для выполнения классических вычислений необходима физическая система, имеющая два устойчивых состояния. Триггеры - в радиоэлектронике. Представление в виде двоичной системы: 0;1 - биты.

Бит (binary digits = двоичные разряды)- это система, имеющая два устойчивых состояния. Между этими состояниями должен быть достаточно большой энергетический барьер, чтобы система не могла спонтанно переходить из одного состояния в другое. Например, в TTL (транзисторно-транзисторной)-логике уровню “0” (низкий уровень) соответствуют сигналы, принадлежащие диапазону $-0.5 < U_0 < 0.4$В, а уровню “1” - $2.4 < U_1 < 5.5$В. В булевой алгебре им соответствуют понятия “ЛОЖЬ” (“FALSE”) и “ИСТИНА” (“TRUE”). Термин “помехоустойчивость” используется для обозначения максимального уровня помехи, которая будучи прибавлена к логическому сигналу при самых неблагоприятных условиях, не будет приводить к ошибочной работе схемы. Для ТТЛ устройств помехоустойчивость составляет 0.4В

Классический регистр - это совокупность некоторого числа L битов. Он имеет 2^L различных состояний, В данный момент времени может существовать лишь одно из этих состояний.

Машина Тьюринга - это математическая модель идеализированного вычислительного устройства. Она имеет
1) ленту, разбитую на конечное число ячеек, в каждой ячейке ленты в определенный момент времени записан один из символов a0,a1,a2,...,aN. Совокупность этих символов называется входным алфавитом;
2) конечное управление, которое может находиться в каждый момент времени в каком-то одном состоянии q0, q1, q2,...,qM;
3) управляющую головку, которая может перемещаться вдоль ленты, считывать или записывать символы.

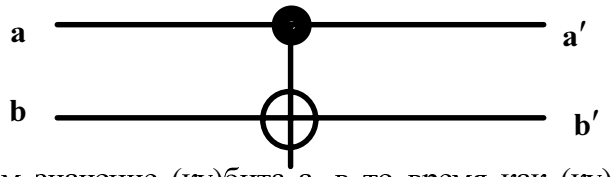
Машина действует в дискретные моменты времени. В зависимости от внутреннего состояния и от символа, считанного головкой, в следующий момент времени машина может перейти в другое состояние и записать в ячейке символ. Эти переходы из одной конфигурации в другую она выполняет согласно командам. Попав в представляющее состояние машина останавливается. Формальное определение базовой модели машины Тьюринга:
T= (K,S, Г, d, q_0, F);
K - конечное множество внутренних состояний;
S - входной алфавит;
Г - ленточный алфавит: S: Г-{$}, $-пробел;
d - команды, частичное отображение
d: KxГ->Kx (Г-{$})x{L,R}, где L,R -движение влево и вправо головки;
q_0- начальное состояние, с него машина начинает обработку, q0cK;
F - множество конечных состояний, в которых машина переходит в представляющее состояние.
Любая машина Тьюринга описывается таблицей, состоящей из строк вида (q_j c_k v_{jk} q_{jk}) , имеющих следующее значение: из состояния q_j, если под считывающей головкой находится символ c_k , принадлежащий Г, перейти в состояние q_{jk} и выполнить действие, предписываемое символом v_{jk} , принадлежащим CU{$, R, L, s}, где Г - рабочий алфавит машины Тьюринга; $ - пустая буква; R, L - сдвиг считывающей головки соответственно вправо или влево; s - символ останова

Классические вычисления.
Вычисление - это алгоритм, по которому некоторому значению на входе в систему (X) ставится в соответствие значение на ее выходе (Y).
X $\Rightarrow$ Y=F(X).
В общем случае классическое вычисление не является обратимым, т.е. не существует алгоритма, по которому
Y $\Rightarrow$ $X=F^{-1}(Y)$.

Основные ЛО это: И (AND), ИЛИ (OR), НЕ(NOT) и их комбинации, например, И-НЕ, ИЛИ-НЕ
ПРИМЕР 1: двухбитовая операция “Сложение по модулю 2” (операция исключающего НЕ, или CNOT или XOR).

a	b	a⊕b
0	0	0
1	0	1
0	1	1
1	1	0

Ее квантовый аналог



Сохраняем значение (ку)бита a, в то время как (ку)бит b меняется по закону XOR:

a	b	a′	b′
0	0	0	0
1	0	1	1
0	1	0	1
1	1	1	0

- бит **b** (мишень = target) меняет свое состояние тогда и только тогда, когда состояние контрольного (control) бита **a** соответствует 1; при этом, состояние контрольного бита не меняется.

ПРИМЕР 2: однобитовая операция “НЕ” (NOT).

a	F(a)
0	1
1	0

Эта операция обратимая, т.е. **F(F(a))=a.** В общем случае это не так.

Однако, классический компьютер (классическое вычисление) может быть сделан обратимым, если сохранять в памяти входной сигнал.

ПРИМЕР: Сложение по модулю 2, с сохранением **a**.

(a,b) ⇒ (a,a⊕b)

(a, a⊕b) ⇒ (a, a⊕b⊕a)

$F=F^{-1}$, в том смысле, что подействовав на входные данные два раза по одному и тому же алгоритму мы опять получаем входные данные:

F(F(X))=X, или F(Y)=X

a	b	a⊕b	a⊕b⊕a≡b
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	1

Логическая операция XOR (CNOT) иллюстрирует почему классические данные могут быть клонированы, а квантовые - нет. Заметим, что в общем случае под квантовыми данными мы будем понимать суперпозиции вида

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad (1)$$

где α и β - комплексные числа или амплитуды состояний, причем, $|\alpha|^2 + |\beta|^2 = 1$.

Согласно таблице истинности, если XOR применить к булевым данным, в которых второй бит находится в состоянии “0” (b), а первый -в состоянии “X” (a), то первый бит не изменяется, а второй становится его копией:

U_{XOR}(X, 0) = (X, X), где X = “0” или “1”.

В квантовом случае, в качестве данных, обозначенных символом “X”, нужно рассматривать суперпозицию (1):

$$U_{XOR}|X,0\rangle = |X,X\rangle.$$

Физически, данные можно закодировать, например, в поляризационном базисе |V> = 1, |H> = 0 (H,V)= (0,1):

$$U_{XOR}|0,0\rangle = |0,0\rangle, \text{ и } U_{XOR}|1,0\rangle = |1,1\rangle.$$

В первом случае действительно имеет место копирование состояния. Теорема о запрете клонирования утверждает, что невозможно копирование *произвольного* квантового состояния. В рассмотренном примере копирование произошло, поскольку операция производилась в собственном базисе (|0>, |1>), т.е. в частном случае квантового состояния.

Казалось бы, что операцию XOR можно использовать и для копирования суперпозиций двух булевых состояний, таких как $|45^0\rangle \rightarrow |V\rangle + |H\rangle$:

$$U_{XOR}|45^o, H\rangle = |45^o, 45^o\rangle.$$

Но это не так! Унитарность квантовой эволюции требует, чтобы суперпозиция входных состояний преобразовывалась в соответствующую суперпозицию выходных состояний:

$$U_{XOR}|45^o, H\rangle = \left(|45^o\rangle = 1/\sqrt{2}\left[|H\rangle + |V\rangle\right]\right) = 1/\sqrt{2}\left[|H,H\rangle + |V,V\rangle\right] \text{ или}$$

$$U_{XOR}|s,0\rangle = \left(|s\rangle = 1/\sqrt{2}\left[|0\rangle + |1\rangle\right]\right) = 1/\sqrt{2}\left[|0,0\rangle + |1,1\rangle\right].$$

Это т.н. перепутанное состояние (Φ^+), в котором каждый из двух выходных кубитов не имеет определенного значения (в данном случае - поляризации). Этот пример показывает, что логические операции, выполняемые над квантовыми объектами происходят по другим правилам, нежели в классических вычислительных процессах.

3. Квантовый компьютер - физическое устройство, выполняющее логические операции над квантовыми состояниями путем унитарных преобразований (т.е. сохраняющих энергию), не нарушающих квантовые суперпозиции в процессе вычислений. Схематично, работа квантового компьютера может быть представлена как последовательность трех операций:

1. “ЗАПИСЬ” (приготовление) начального состояния,
2. “ВЫЧИСЛЕНИЕ” (унитарные преобразования начальных состояний)
3. “ВЫВОД” результата (измерение, проецирование конечного состояния).

Также сюда следует отнести вспомогательную операцию “СБРОС”, приводящую регистр к основному состоянию.

ОБЩИЕ ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К КВАНТОВЫМ КОМПЬЮТЕРАМ.

1. Должна быть реализована система квантовых битов или кубитов:
   $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, где α и β - комплексные числа или амплитуды состояний, причем, $|\alpha|^2 + |\beta|^2 = 1$. Кубит - это когерентная суперпозиция двух различимых квантовых состояний. Например, поляризации фотона (H, V), два состояния спина электрона, внутренние электронные состояния индивидуального атома.
2. Должен существовать механизм, осуществляющий “перепутывание” кубитов:
   $|0\rangle|0\rangle \to \frac{1}{\sqrt{2}}\{|0\rangle|0\rangle + |1\rangle|1\rangle\}$, см. пример выше.

Перепутывание - (“запутывание”, “сцепленность”, “переплетение” - от немецкого verschrankung) - это квантовая разновидность корреляции, не имеющей классического аналога. Грубо говоря, *две подсистемы являются перепутанными, когда их совместное состояние более определено и менее стохастично, чем состояние любой из подсистем*.

3. Необходимо осуществлять контролируемым образом логические операции над кубитами (“условная динамика”).
   например, $|a\rangle_A|b\rangle_B \to |a\rangle_A|a \oplus b\rangle_B$. Устройства, служащие для этого называются логическим элементами (ЛЭ, gates).
4. Система должна быть масштабируемой, т.е. логические операции должны распространяться на *N* кубитов.
5. Физическая система, представляющая собой данные, должна быть квантово-механически стабильна.
6. Необходим механизм, осуществляющий запись, чтение и сброс данных.

Зачем нужен квантовый компьютер? Например, рассмотрим математическую проблему факторизации больших чисел - т.е. разложения произвольного числа на простые множители (Любое число можно разложить на простые множители. Док-во: Каждое целое число является либо простым, либо - нет. Если оно простое - то оно представимо в виде произведения единицы на само себя. Если нет - то оно выражается в виде произведения двух чисел. Рассмотрим каждое из них. Оно либо является простым, либо нет. И т.д.).
Эта задача непосредственно связана с криптографией, где секретные ключи формируются именно посредством такого алгоритма.
Математики твердо верят, хотя они и не доказали это, что для факторизации числа с *N* десятичными разрядами любому классическому компьютеру требуется число шагов, которое растет экспоненциально с *N*. Иначе говоря, добавление одного десятичного разряда к числу общем случае умножает время, необходимое для его факторизации, на постоянный множитель. Конкретно, время растет с ростом длины *N* факторизуемого числа как $\exp(N^{1/3})$. Так, задача вычисления произведения двух простых чисел 521 и 809 не вызывает проблем.

Однако, обратная задача - нахождение простых сомножителей числа 421489 потребует определенного времени.
Таким образом, при увеличении числа разрядов задача быстро становится неразрешимой. Наибольшее число, которое было разложено на простые множители в качестве математического соревнования, т.е. число, чьи простые множители были втайне выбраны математиками, чтобы составить задачу для других математиков, состояло из 129 разрядов. Если же число разрядов окажется порядка 1000, то никто не знает, как решить эту задачу. Квантовый алгоритм факторизации (П.Шора) позволит реализовать эту операцию за долю секунды. Невыполнимость факторизации лежит в основе наиболее надежных на сегодняшний день методов шифрования (в частности системы RSA - Rivest, Shamir, Adleman), которая используется для защиты электронных банковских счетов. Когда будет построена машина для квантовой факторизации все такие криптографические системы станут абсолютно бесполезны.

*Дополнительно - (если есть время). Как действует классический компьютер при факторизации числа на простые множители?
Алгоритм решения этой задачи сводится к нахождению периода вспомогательной функции. Такой функцией является остаток от деления степенной функции вида a^x на целое число N
$f_N(x) = a^x \ mod\ N$,
(где x = 0, 1, 2...), a - любое число, не имеющее общих делителей с рассматриваемым числом N.
Например, $N = 10$. Выберем $a = 7$.

x	0	1	2	3	4	5	6
a^x	7^0	7^1	7^2	7^3	7^4	7^5	7^6
a^x	1	7	49	343	2401	16807	117649
$a^x\ mod\ 10$	1	7	9	3	1	7	9

Видно, что период функции f равен 4: $r = 4$. Далее необходимо найти наибольшие общие делители чисел N (исходное число, которое нужно разложить) и $a^{r/2} \pm 1$. Для этого используют алгоритм Евклида:
Требуется найти наибольший общий делитель чисел P и Q.
$P = Q\,T + R_1$,
$Q = R_1T_2 + R_2$,
$R_1 = R_2T_3 + R_3$,
.....,
$R_{m-2} = R_{m-1}T_m + R_m$,
$R_{m-1} = R_mT_{m+1}$. Предшествующий остаток R_m и является наибольшим общим делителем.
В нашем примере:
$a^{r/2} \pm 1 = 48, 50$.
50 = 5x10, т.е. первый делитель - число 5,
48 = 4x10 + 8,
10 = 1x8 + 2,
8 = 4x2. Другой делитель - число 2
Итого, 10 = 5 x 2.

Если же одно из получившихся чисел не является простым, то для него указанный алгоритм повторяется.

Другой пример - Классическому компьютеру требуется время, пропорциональное N для поиска определенного элемента в базе данных, состоящей из N элементов. Это делается, например, методом перебора. Это пример т.н. проблема ***NP***-класса сложности, в смысле их выполнимости (*Nondeterministic polynomial-time complete*). Квантовый компьютер, работающей по алгоритму Гровера может выполнить эту операцию за время пропорциональное $\sqrt{N}$.
Заметим, что огромное увеличение производительности квантовых компьютеров по сравнению с классическими позволят рассчитать такие фундаментальные физические задачи как эволюция квантовой системы многих тел.

<u>ПРОБЛЕМЫ:</u>

1. Декогерентность (ее наличие ведет к необходимости использовать алгоритмы “коррекции ошибок”)
2. Как осуществить перепутывание контролируемым образом?
3. Как передать информацию от одной части вычислительного устройства к другой? Что является квантово-механическим аналогом проводов и шин данных в классических компьютерах
4. Проблема квантовых измерений. Как выполнять измерения без разрушения квантовых состояний? Необходимы алгоритмы “коррекции детектирования” и чтения данных.

Вычисление на (квантовом) компьютере называется эффективным, если число элементарных операций Q растет не быстрее, чем по полиномиальному закону по числу входных (ку)битов L Если убрать то, что стоит в скобках, получится классическое определение эффективности.

Пусть 1/R - некоторое характерное время, требуемое для выполнения элементарной операции. Пусть T_{dec} - время декогерентности системы (T_2). Тогда должно выполняться соотношение:
$Q/R<T_{dec}$.
Обозначим, $T_{dec} = L\gamma$, где γ - некоторое характерное время декогерентности кубита. Тогда число элементарных операций ограничено фактором:
$Q<R(L\gamma)$.