

## Потенциальное кодирование:

для кодирования 1 используется положительный потенциал, а для 0 отрицательный, при этом скорость передачи равна  $n$  бит в сек

В этом случае спектр сигнала состоит из бесконечного ряда гармоник,  $f_0, 3f_0, 5f_0, 7f_0$  и т.д. где  $f_0 = n/2$ .

При этом амплитуды гармоник убывают достаточно медленно с коэффициентами обратными коэффициентам стоящим у частот гармоник. Таким образом, передача сигнала в среде с узкой полосой пропускания труднореализуемо в следствии низкой помехоустойчивости. Потому например в телефонных линиях на каналах тональной частоты не применяется потенциальное кодирование (300-3400 Гц, человек. голос 10 — 10000 Гц).

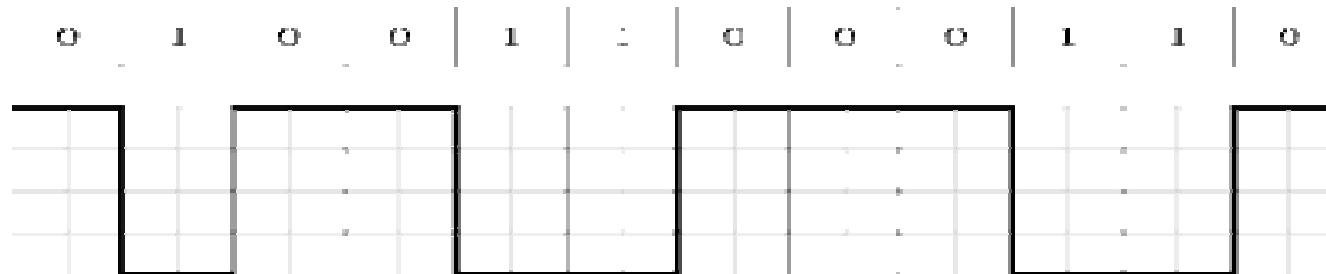
- При **амплитудной модуляции** появляется кроме несущей частоты две дополнительные гармоники  $f_c - f_1$ ,  $f_c + f_1$ . Единица кодируется максимумом амплитуды, 0 половиной этой амплитуды. При этом частота  $2f_1$  — определяет полосу пропускания для данного вида кодирования.
- При частотной и фазовой модуляции появляется более двух гармоник, но они симметричны относительно друг друга, потому подобрать полосу пропускания проще. В данном случае используется две различные частоты или две различные фазы. Существуют комбинированные методы кодирования например использования четырех уровней амплитуд и 8 разных фаз (с шагом 45 градусов). Для повышение надежности передачи в таком коде например в коде Треллиса разрешены только 3 из состояний, остальные используются для устранения искажений.

# Необходимость самосинхронизации

- В цифровом кодировании применяется **потенциальное и импульсное кодирование**, при этом необходима **синхронизация** источника сигнала и приемника, для того, чтобы приемник знал время когда нужно начать считывать полезный сигнал. В случае сетей реализация синхронизации сложнее чем в **близкорасположенных устройствах**, так как использование дополнительного канала передачи тактовых импульсов дорогая задача, при этом сами тактовые импульсы могут прийти позже или раньше сигнала в следствии сложности пути прохождения сигнала, потому применяется **самосинхронизация**. Например, может использоваться любой резкий перепад или фронт для синхронизации, синусоидальный сигнал синхронизирован, так как при изменении амплитуды несущей частоты может показать приход входного кода. При цифровом кодировании необходимо обеспечить наименьшую ширину спектра сигнала, распознавание ошибок и синхронизацию.

# Потенциальный код без возвращения к нулю (NRZ, non return to zero)

- В этом случае сигнал кодируется с помощью значения какого либо потенциала в качестве 1 и нуля с помощью нулевого потенциала, при этом в конце такта 1-ца не сбрасывается на ноль, данный код имеет тот недостаток, что при длинной последовательности 1 появляется низкочастотная составляющая увеличивая ширину спектра сигнала, при этом при ошибках в тактовых импульсах, при длинных последовательностях 0 и 1 можно прийти к рассинхронизации в 1 и более тактов. При чередовании 1 и 0 присутствует высокочастотная составляющая, потому сигнал может иметь широкий спектр.



# RZ (return to zero, с возвращением к нулю)

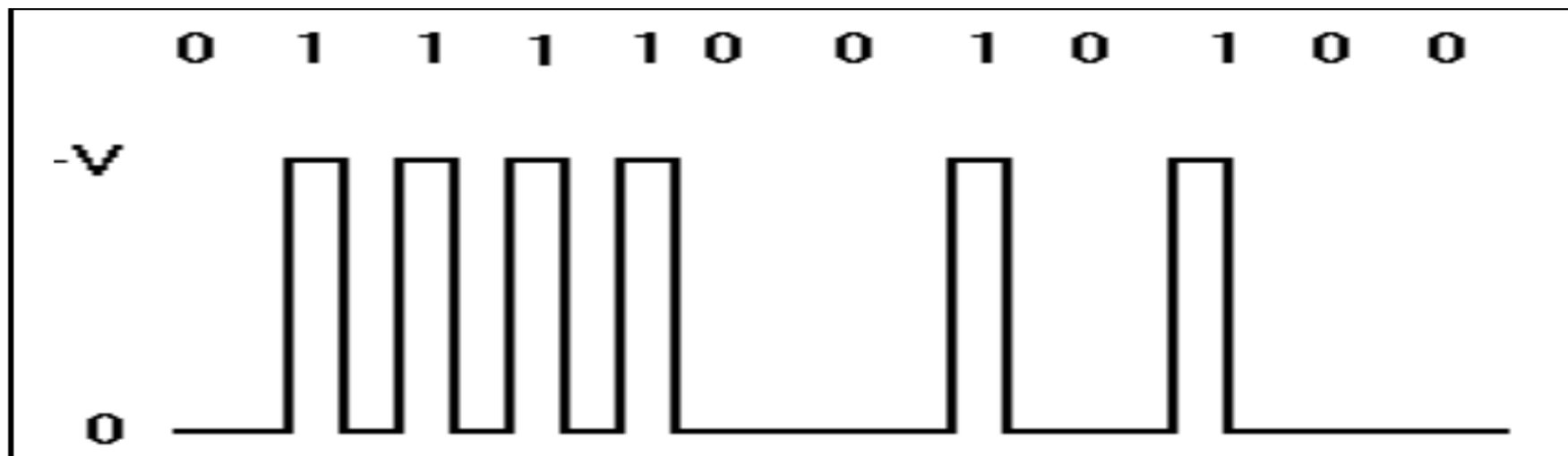
биты 0 представляются нулевым напряжением (0 В);

биты 1 представляются значением  $+V$  в первой половине и нулевым напряжением – во второй, т.е. единице соответствует импульс напряжения продолжительностью в половину длительности передачи одного бита данных.

Этот метод имеет два преимущества по сравнению с кодированием NRZ: средний уровень напряжения в линии составляет  $\frac{1}{4} V$  (вместо  $\frac{1}{2} V$ );

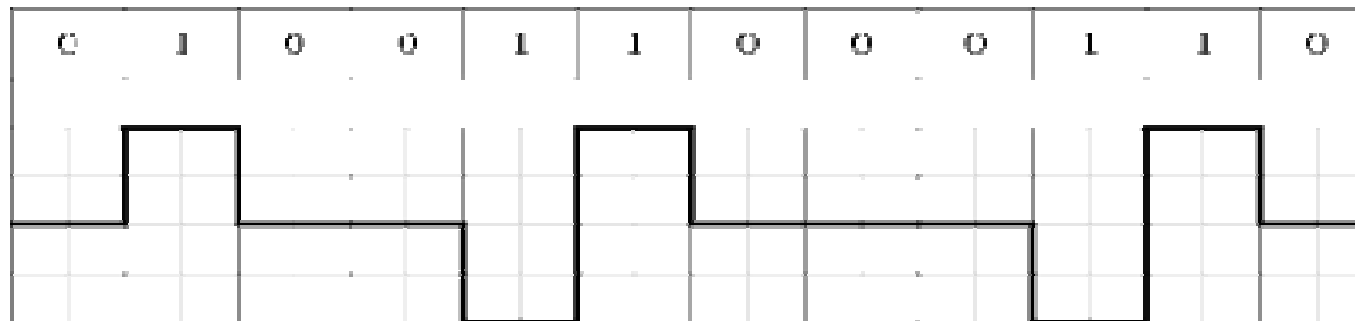
при передаче непрерывной последовательности 1 сигнал в линии не остается постоянным.

Однако при использовании кодирования RZ полоса сигнала может достигать значений, равных скорости передачи данных (при передаче последовательности 1).

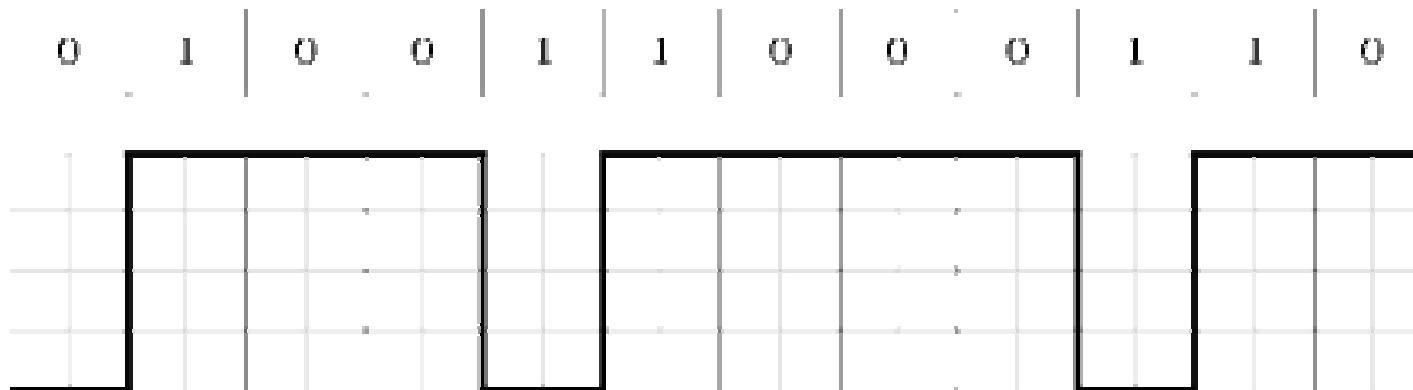


# Метод биполярного кодирования с альтернативной инверсией (АМІ)

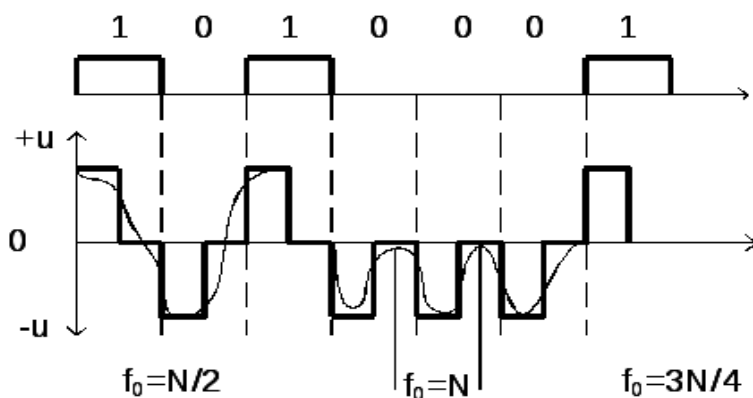
- Присутствует три уровня нулевой, положительный и отрицательный. 0 кодируется нулевым потенциалом, каждая единица последовательно сначала положительным потенциалом, следующая единица отрицательным и так далее. При этом уходит проблема рассинхронизации единиц, но при последовательности нулей остается также проблема, при этом частично остается проблема низкочастотной составляющей при длинных последовательностях нулей. Появляется возможность обнаружения ошибок при нарушении чередования полярностей в единицах.



- Потенциальный код с инверсией при единице: в этом случае **ноль** кодируется **неизменным потенциалом** на каждом такте, а при поступлении для кодирования **1** потенциал меняется на противоположный. Плохая синхронизация по нулям. NRZI (Non return to zero invertive)



- Биполярный импульсный код: В этом случае в такте передается импульс который длится пол такта, в этом случае, **0** кодируется импульсом одной полярности, а **1** другой. Код хорошо самосинхронизирован, но при этом может присутствовать низкочастотная составляющая при длинной последовательности 0 или 1.



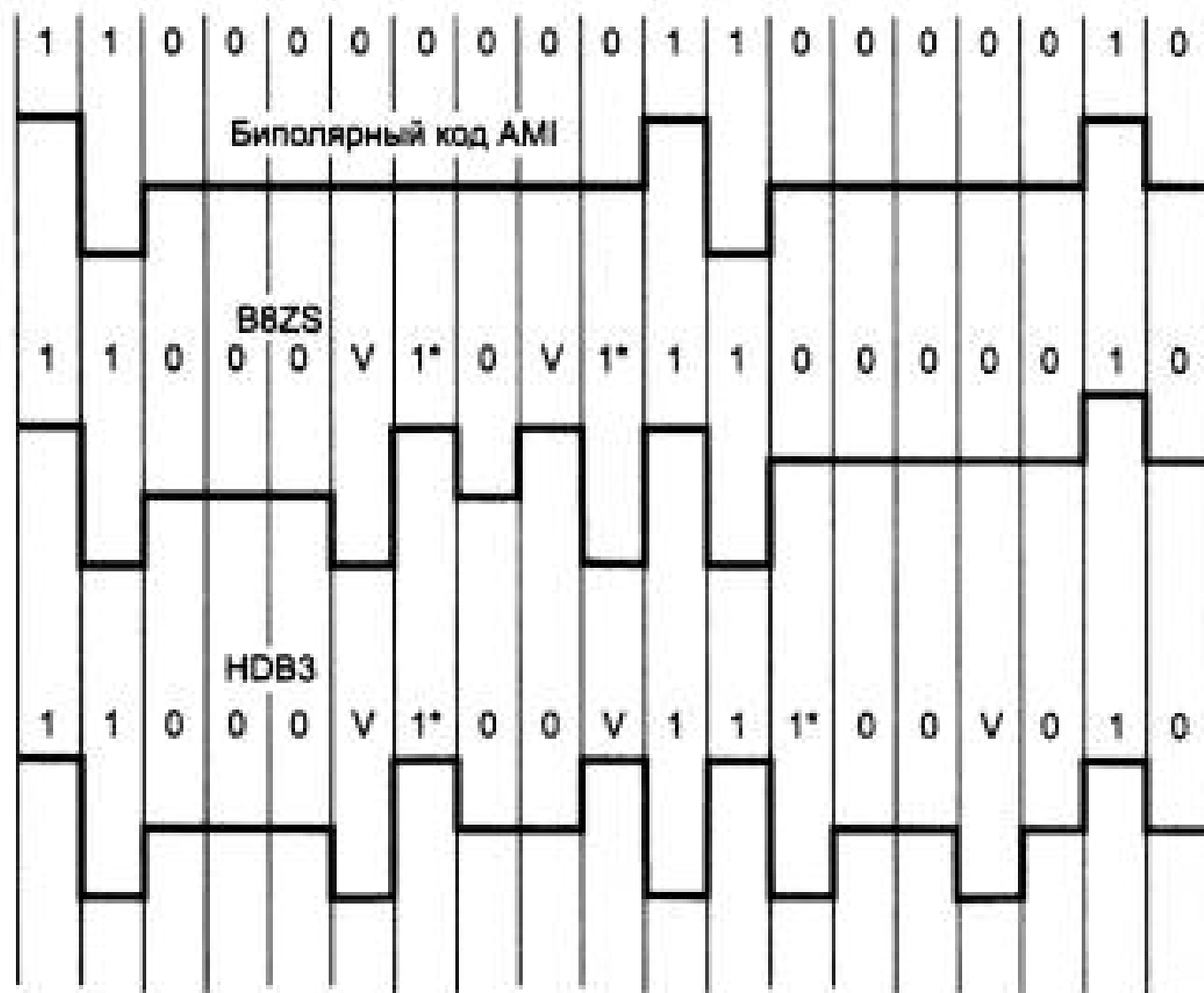
$f_0 = N$

$N$  – частота  
передачи бит в Гц



- Код **B8ZS** используется для корректировки биполярного кода путем замены 8 битных последовательностей нулей, после трех нулей, 5 оставшихся заменяются запрещенной полярностью при смене 1, при этом идут два подряд искажения, которые маловероятны и показывают, что на самом деле там идут нули. V — запрещенная смена полярности

- Код **HDB3** исправляет любые четыре подряд идущих нуля в исходной последовательности. Правила формирования кода HDB3 более сложные, чем кода B8ZS. Каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал V. Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах. Кроме того, для замены используются два образца четырехтактовых кодов. Если перед заменой исходный код содержал нечетное число единиц, то используется последовательность 000V, а если число единиц было четным - последовательность 1\*00V.



# Кодирование 4B3T

**4B3T** (4 Binary 3 Ternary, когда 4 двоичных символа передаются с помощью 3 троичных символов) Используется для передачи данных по физическому каналу связи такому как оптическое волокно, витая пара, коаксиальный кабель, инфракрасному излучению. Сигнал на выходе кодирующего устройства, согласно коду 4B3T, является трехуровневым, т.е. на выходе кодирующего устройства формируется сигнал с тремя потенциальными уровнями. Код формируется, например, согласно таблице кодирования MMS43. Каждые четыре битовые комбинации представляются трехуровневой (с тремя отличающимися потенциалами) комбинацией.

# Таблица декодирования 3Т4В

Trit	Bit	Trit	Bit	Trit	Bit
0 0 0	н/д	– 0 0	0101	+ – –	1010
+ 0 +	0000	– + +	0110	+ 0 –	1011
0 – 0	0000	– – +	0110	+ + +	1100
0 – +	0001	– 0 +	0111	– + –	1100
+ – 0	0010	+ 0 0	1000	0 + 0	1101
0 0 +	0011	0 – –	1000	– 0 –	1101
– – 0	0011	+ – +	1001	0 + –	1110
– + 0	0100	– – –	1001	+ + 0	1111
0 + +	0101	+ + –	1010	0 0 –	1111

## Логическое кодирование.

Необходимо для выявления и обнаружения ошибок, устранения проблем с длинными последовательностями из нулей.

- Введение избыточности, например в Fast ethernet используется код 4b, 5b. Четырехбитовая последовательность заменяется пятибитовой, в которой идет чередование 1 и нулей, код гарантирует, что последовательность с более чем тремя нулями не будет встречаться.

## Кодовые Символы 4B/5F

Коды Данных

4B Код	5B Символ
0000	1'110
0001	0'001
0010	10100
0011	10101
0100	0'010
0101	0'011
0110	0'110
0111	0'111
1000	10010
1001	10011
1010	10110
1011	10111
1100	1'010
1101	1'011
1110	1'100
1111	1'101

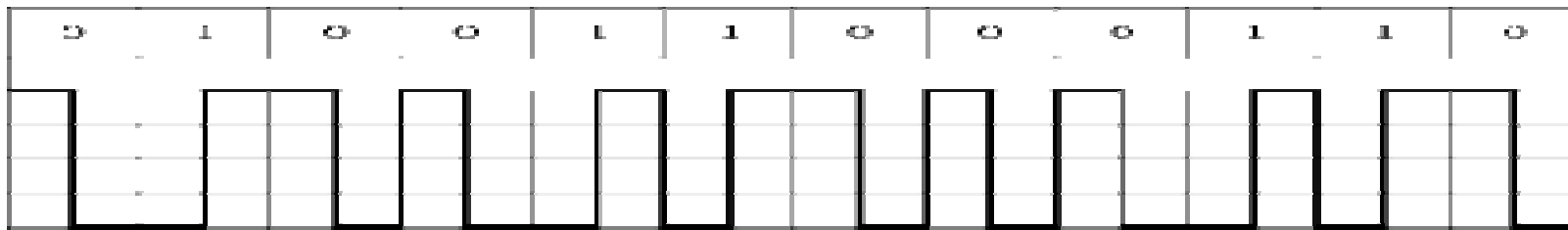
Управляющие и Недопустимые Коды

4E Код	5B Символ
Ожидание	11111
Начало потока	11000
Начало потока	10001
Конец потока	01101
Конец потока	00111
Ошибка передачи	00100
Недопустимый	00000
Недопустимый	00001
Недопустимый	00010
Недопустимый	00011
Недопустимый	00100
Недопустимый	00101
Недопустимый	00110
Недопустимый	01000
Недопустимый	10000
Недопустимый	11001

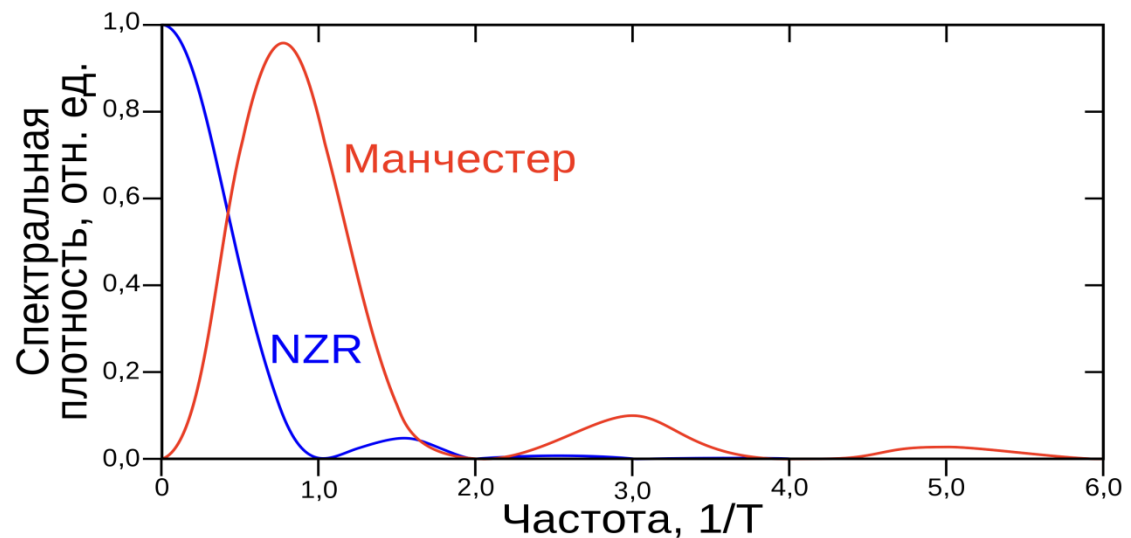
- **Скрэмблирование**, основано на смешивании кодируемого бита с предыдущими значениями кода полученного **скрэмблером**.  
Например
- $V(i) = A(i) + V(i-3) + V(i-5)$  — сложение по модулю два. Дискрэмблирование
- $A(i) = V(i) + V(i-3) + V(i-5)$
- (может использоваться большее число предыдущих бит). На начальном этапе, начальные кодирующие скрэмблером биты равны исходным. Допустим  $V(1) = A(1)$   $V(4) = A(4) + V(1)$ . Для кода 110110000001 получается код 110001101111, то есть ушла длинная последовательность нулей.



- Манчестерский код. Когда-то был распространён в локальных сетях (технологии Ethernet (10Base-T, 10мбит/с), Talking Ring). 0 и 1 кодируются перепадом потенциала в середине такта, от положительного к отрицательному и наоборот. Полоса пропускания для него уже чем у биполярного импульсного. Самосинхронизирован, но повышенные требования к джиттеру – стабильности фронтов.

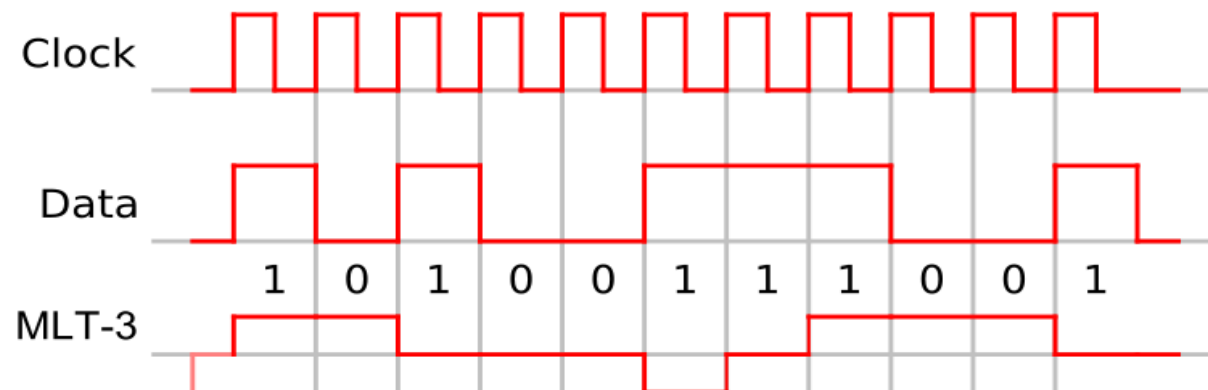


Сравнение спектров Манчестерского кода и потенциального без возвращения к нулю. Манчестерский имеет максимум на частоте передачи тактовой последовательности.

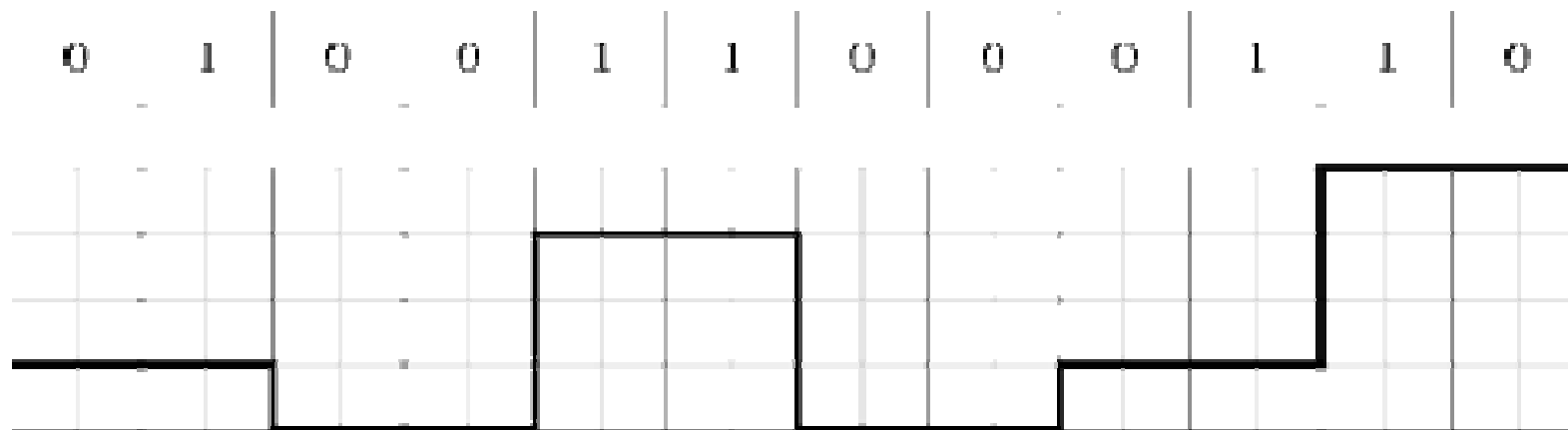


# Код MLT-3 (Multi Level Transmission-3)

Применялся в сетях FDDI и применяется в сетях Fast Ethernet 100 мбит/с (100Base-TX). Похож на NRZI, но для 1 использует три уровня последовательной смены потенциала -1, 0, 1. Нулевой бит при этом не меняет уровень. То есть каждая встреченная 1 кодируется сначала ,например как -1, потом как 0, потом как 1, потом как 0, потом как -1 и т.д. Максимальная частота основной гармоники 25 МГц, тогда как у манчестерского кода 100 МГц и у NRZ 50 МГц для скоростей 100 Мбит/с, и соответственно в отличие от MLT-3 для кабелей 100Base они не подходят.

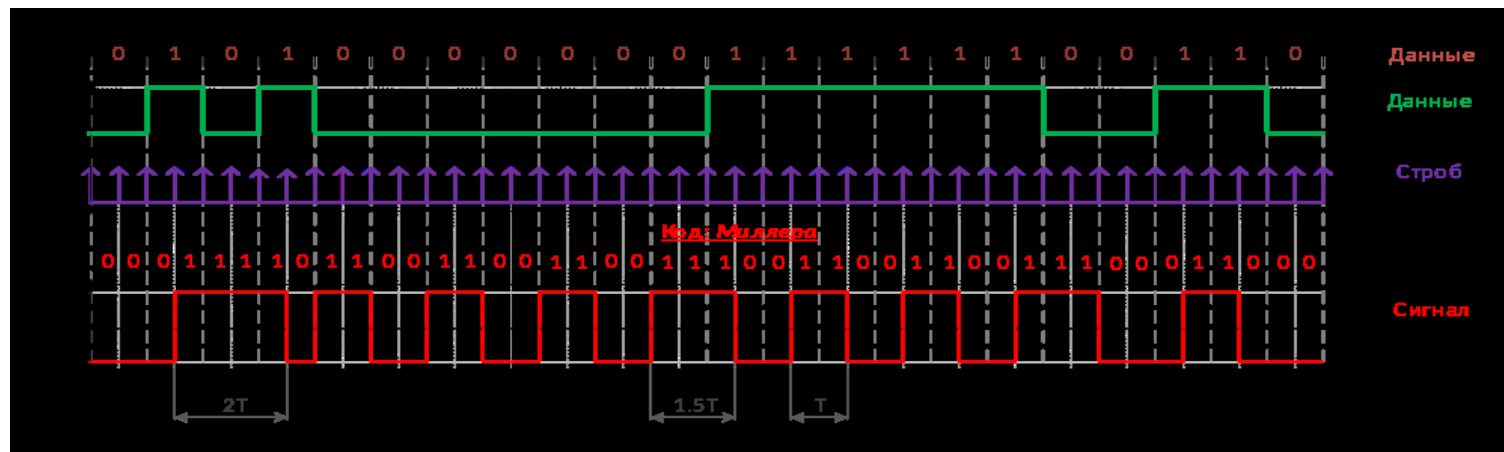
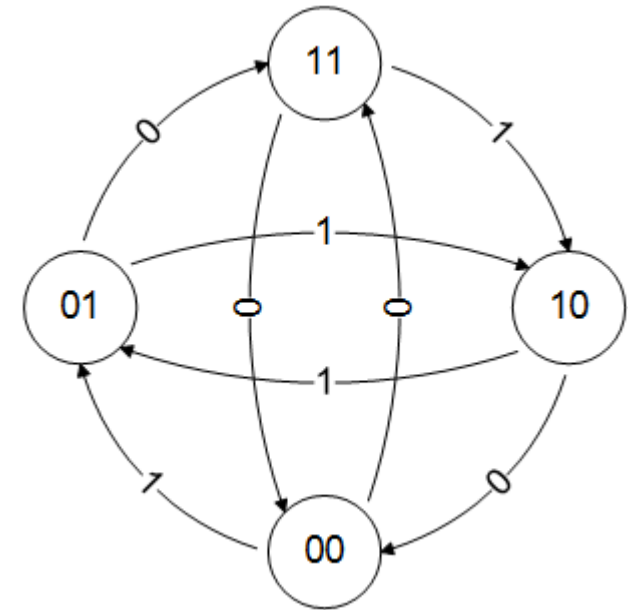


- Потенциальный код **2B1Q**, здесь используется четыре уровня потенциала **-2.5, -0.833, 0.833, 2.5 В**. При этом каждое состояние кодирует 2 бита.
- Достоинство метода 2B1Q
- **Спектр сигнала в два раза уже** чем у кодов NRZ и AMI. С помощью 2B1Q-кода можно по одной и той же линии передавать данные **в два раза быстрее**.
- Недостаток метода 2B1Q
- Реализация этого метода требует более мощного передатчика и более сложного приемника, который должен различать четыре уровня.



# Код Миллера

- один информационный символ кодируется двумя дибитами с учетом предыдущего состояния. Граф переходов представлен на рисунке. В ребрах поступающий информационных бит, в вершинах предыдущие полученные дибиты. Хорошо синхронизован. Спектр сформированный такой последовательностью будет иметь три различных полосы, соответствующие периоду  $T$ ,  $1.5T$  и  $2T$ .



# РАМ-5

Коды использующиеся в технологиях 1Гбит/сек 1000Base-T. В данном случае используется пятиуровневая схема кодирования похожая на 2B1Q. Но с уровнями (-2, -1, 0, 1, 2). Нулевой уровень используется для коррекции ошибок, четыре других уровня используются для передачи дибитов (двойных битов). При этом дибиты формируются по схеме треллис кодирования из одного информационного бита, и декодируются с использованием алгоритма Витерби.

# Треллис кодирование и витерби декодирование

Реализуется с использованием нескольких ячеек памяти. Схема кодирования с двумя состояниями получается по следующим формулам. Получение двух дибит. Начальное состояние ячеек нулевое, то есть  $A_0=0$ ,  $A_1=0$ ,  $A_2$  можно считать информационным битом.

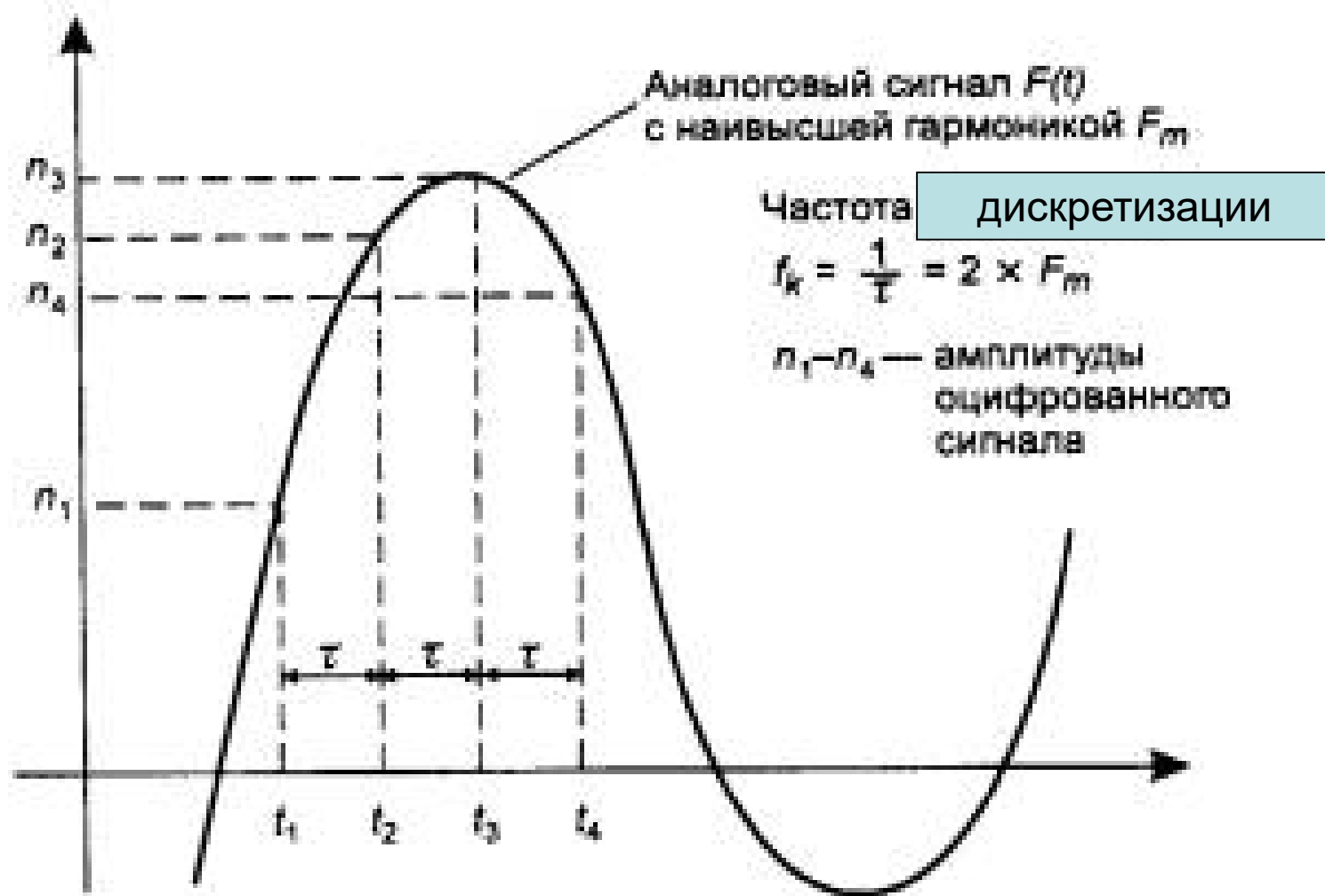
$$V_{i*2}=A_i+A_{i-1}+A_{i-2}$$

$$V_{i*2+1}=A_i+A_{i-2}. \text{ Сложение по модулю два.}$$

Алгоритм Витерби основан на анализе дерева смены состояний и выбора наиболее вероятного состояния с минимальным расстоянием Хэмминга. Фактически используется метод динамического программирования.

Например, мы передали код 0, на приемнике, допустим получили 00, в данном случае наиболее вероятно 0, при начальном состоянии 00 с расстоянием 0, если мы передали бы 1, то получили бы 11, но мы получили 00, оставим возможность данной ветки с расстоянием хэмминга 2. Далее рассматриваем возможные смены состояний, накапливая расстояние хэмминга и затем выбираем ветку с наименьшим расстоянием хэмминга.

- **Дискретная модуляция аналоговых сигналов.**
- При дискретной модуляции применяется квантование и дискретизация, когда значения сигнала разбиваются на интервалы — квантование и по аргументу сигнала (времени или пространству) сигнал дискретизируется. Например дискретизации 8000 Гц вполне достаточно для передачи звука, при этом если квантование подразумевает 7 бит на уровни, то мы получим 56 Кбит в секунду требуется для передачи цифрового звукового сигнала. Цифровой сигнал обладает при этом тем преимуществом, что обладает большей помехоустойчивостью, дает возможность обнаружить более просто помехи и искажения, в отличии от того же аналогового. Теорема Котельникова говорит о том, что для точного восстановления сигнала при кодировании достаточно дискретизировать сигнал с частотой в два раза большей чем максимально возможная частота присутствующая в сигнале. Примеры сетей передающих цифровой звук и изображения — isdn, atm.





- **Асинхронная и синхронная передача.**
- Физический канал связи предусматривает побитовую синхронизацию при передаче данных. Канальный оперирует с кадрами и обеспечивает кадровую синхронизацию. Приемник должен распознать первый байт кадра, распознать границы полей кадра, а также окончание кадра. Иногда вводится побайтовая синхронизация, в этом случае каждый байт снабжается дополнительным старт и стоп сигналами, обозначающими начало и конец байта, старт сигнал обычно длится 1 такт, стоп -1, 2 такта. При этом передаваемые байты могут быть разделены временными промежутками, такая передача называется асинхронной. При синхронной передаче байт, весь кадр последовательно байт передается побитово в каждом такте, при этом кадр снабжается старт байтом и стоп байтом.

# Методы и протоколы передачи данных канального уровня.

- Канальный уровень обеспечивает прием пакетов от вышестоящего уровня, а также доставку по адресу указанного в данном пакете. Протоколы данного уровня оформляют пакеты в кадры собственного формата, помещая адрес назначения в одно из полей кадра и добавляю контрольную сумму. Канальный протокол имеет обычно локальный смысл в пределах локальной сети с простой топологией связи. Также обеспечивает соединение типа точка-точка в глобальных сетях при передаче кадров непосредственному соседу.

- Асинхронные протоколы в самом начале оперировали не кадрами, а отдельными символами или байтами со стоп и старт битами (старт стоп сигналами) или символами. Например в кодировке ascii. Некоторые символы могли иметь управляющий характер. Далее протоколы усложнялись и уже передавались целые кадры и блоки данных. Например, в протоколе xmodem принимающая сторона постоянно отправляет символ ascii nak, отправляющая сторона получая данный символ отправляет блок из 128 байт из данных, заголовка и концевика с контрольной суммой. Принимающая сторона проверяет данные и контрольную сумму, если все правильно то, принимающая сторона отправляет символ ack, в противном случае nak, и отправитель заново делает передачу. В конце передачи следует символ eoh. То в данном протоколе применяется как асинхронная передача с управляющими символами и покадровая присущая синхронным протоколам.

- Синхронные протоколы не предполагают передачу отдельных байт. Все обмены данными осуществляются кадрами. Передача в данном случае осуществляется одним потоком бит в кадре, при этом скорость передачи выше. Каждый кадр в данном случае содержит заголовок, поле данных и концевик. При этом, задача приемника распознать старт биты и стоп биты. Поле данных может иметь как фиксированную (например протокол АТМ, с длиной 53 байта) так и переменную длину, протоколы ethernet, требующий размера поля данных не менее 43 байт, и протокол fddi допускающий поле данных нулевой длины. В АТМ протоколе таким образом не нужно стоп синхробит.

- Синхронные символьно (байт) - ориентированные протоколы:
- Включают специальные символы означающие начало блока данных и конец, например, в коде `ascii` `syn` — старт байт синхронизации (0010110), после синхронизации и возможности отделения символов друг от друга передается символ `stx` (00000010) и `etx` — стопбайт синхронизации (00000011). Естественно запрещается применять в данных символы синхронизации.

- Синхронные бит-ориентированные протоколы имеют несколько схем: Первая основана на том, что приемник постоянно передает 1 сигнал, затем идет передача специального стартового флага байта 01111110, в конце сообщения также передается флаг байт 01111110. В поле данных запрещается передавать флаг байт, достигается это путем операции стаффинга. После каждых пяти единичных бит вставляется нулевой бит, что обеспечивает отсутствие байта флага в поле данных. Аналогично приемник встречая 5 подряд единичных бит удаляет лишний 0 в данных. Вторая схема основана на том что после флага добавляется информация о длине кадра, тогда завершающий флаг может отсутствовать. В этом случае отправляющая станция синхронизирует принимающие начальной отправкой сигнала 01010101 с помощью которого принимающая входит в синхронизацию, затем анализируя поток на побитовой основе находит стартовый флаг 10101011. Третья схема зависит больше от физического кодирования, когда допустим коде вводятся запрещенные сигналы в качестве стартостопных сигналов. Например, в манчестерском коде — постоянный высокий уровень полярности, или постоянный низкий в течении такта. Старт байт кодируется  $j k 1 j k 1 0 0$ . Где  $j$  — неизменный низкий,  $k$  — неизменный высокий.

## Протоколы с гибким форматом кадра.

- В данном случае поле данных кадра состоит из множества полей для каждого из которых задается их размер, также может задавать тип данных. Например, такими протоколами являются прикладной протокол управления сетями snmp и протокол канального уровня rrr — для реализации соединений точка-точка.

- Существует передача с установлением и без установления логического соединения.
- **Дейтаграммные протоколы**— без установления соединения. Дейтаграммные протоколы не требуют дополнительной служебной информации, а сразу в любой момент времени могут отправить кадр или сообщение, при этом нет гарантии доставки этого сообщения, предполагается, что приемник всегда принимает данные. Протоколы с установлением соединения предварительно отправляют служебный кадр с запросом на соединение, получают ответный кадр с подтверждением, отсылают данные, и получают квитанции кадров подтверждения получения данных, затем отсылают запрос разрыва соединения.



- **Обнаружение и коррекция ошибок.**
- Канальный уровень должен обеспечивать контроль и исправления ошибок в кадрах, большинство протоколов позволяет просто обнаружить ошибки, так как считается что отвечать за правильность отправки пакетов должны отвечать протоколы более высоких уровней, так работают сети ethernet, talking ring, fddi.
- Протоколы канального уровня с устранением искажений — llc2, lap-b. Обычно протоколы канального уровня с исправлением искажений работают в ненадежных сетях или сетях со малой помехоустойчивостью. При этом они позволяют ускорить работу такой сети, более чем если бы это решалось на высоком уровне. В сетях типа ethernet не имеет смысла использовать протоколы канального уровня для исправления искажений, так как это может замедлить работу сети и ведет к удорожанию оборудования.

- **Методы обнаружения ошибок** — основаны на введении избыточности в код, вычислении контрольной суммы или однозначной функции от посылаемого сообщения.
- **Контроль по паритету** — суммирование по модулю два всех бит, и добавление этого бита в код. Обнаруживает одно искажение.
- **Вертикальный и горизонтальный контроль по паритету** — биты байт выстраиваются в матрицу 8 на 8 и при это для столбца и строки рассчитывается сумма бит для каждой строки и столбца, позволяет обнаруживать двойные ошибки.
- **Циклический избыточный контроль** является в настоящее время наиболее популярным. Последовательность байт воспринимается как одно число, контрольная сумма получается путем получения остатка от деления на число большой разрядности (17 разрядов допустим). Приемник получая данные делит опять на тоже число и получает остаток, прибавляет полученное к данным и делит снова, если остаток равен 0, то код правильный.

## Вертикальный и горизонтальный контроль по паритету.

Число 10111101 содержит 6 '1' битов. Бит чётности будет 0, получаем кодовое слово 101111010.

Число 01110011 содержит 5 '1' битов. Бит чётности будет 1, получаем кодовое слово 011100111.

Число 00000000 содержит 0 '1' битов. Бит чётности будет 0, получаем кодовое слово 000000000.

Биты паритета байтов (нечётность)	P	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>
	0	0	0	0	0	0	1	0
	1	0	1	0	1	0	0	0
	0	1	0	0	0	1	1	0
	0	0	1	0	0	0	0	0
	1	0	1	0	1	1	0	1
	0	1	0	0	0	0	0	0
	1	1	1	0	0	0	1	1
	1	0	0	0	0	0	1	1
	1	1	0	0	0	0	0	1
Биты паритета столбцов (чётность)	1	1	0	0	0	0	0	1
	1	1	0	0	0	0	0	1

# Сверточное и блочное кодирование

- Блочное кодирование – добавление к информационной последовательности (блоку) проверочных бит. Каждый блок проверяется своей последовательностью. (Например, код Хэмминга в матричном представлении)
- Сверточное – вставка между последовательностями информационных бит проверочных, при этом проверка потоковая, нет деления на блоки. (Сверточный код Финка).
- Витерби показал, что наиболее оптимальными являются сверточные коды.

# Рекуррентный код Финка

- $A_1, B_1, A_2, B_2, A_3, B_3 \dots A_k, B_k, A_{k+1}, B_{k+1} \dots$
- $B_k = A_{i-s} \odot A_{i+s+1}$  расчет информационного символа  $B$
- С учетом данного способа, ошибка исправляется если в проверочных битах  $k_1 = i-s-1$ , и  $k_2 = i+s$  ошибка, и  $k_2 - k_1 = 2s+1$ , например  $s=0$ ,  $B_1$  и  $B_2$  неверные значит  $A_2$  инвертировать.
- Сам проверочный символ  $k=i$  неверен если ошибка при  $A_{i-s}, A_{i+s+1}$

$S = 0$										
Маркировка символов, текущее значение $i$	$a_1b_1$ 1	$a_2b_2$ 2	$a_3b_3$ 3	$a_4b_4$ 4	$a_5b_5$ 5	$a_6b_6$ 6	$a_7b_7$ 7	$a_8b_8$ 8	$a_9b_9$ 9	$a_{10}b_{10}$ 10
Информационные символы, суммирование по mod2, проверочные символы	0	0	0	1	1	0	1	0	1	1
Выходная последовательность	0	0	0	0	1	1	0	1	1	0
Декодирование при ошибке в одном информационном символе										
Принятая последовательность, суммирование по mod 2, результат суммирования	0	0	0	1	1	0	1	0	1	1
Декодированная последовательность	0	0	0	1	1	0	1	0	1	1
Декодирование при ошибке в одном проверочном символе										
Принятая последовательность, суммирование по mod 2, результат суммирования	0	0	0	1	1	0	1	0	1	1
Декодированная последовательность	0	0	0	1	1	0	1	0	1	1
Декодирование при ошибке в двух проверочных символах										
Принятая последовательность, суммирование по mod 2, результат суммирования	0	0	0	1	0	0	1	1	1	1
Декодированная последовательность	0	0	0	1	1	0	1	0	1	1

$\longrightarrow \quad l_c = 3 \quad \longleftarrow$

Часто сверточные коды описываются многочленами.

Информационная последовательность:

$$A(x)=a_0+a_1X+a_2X^2+\dots \text{ а } \{0,1..\}$$

Формируемые передаваемые

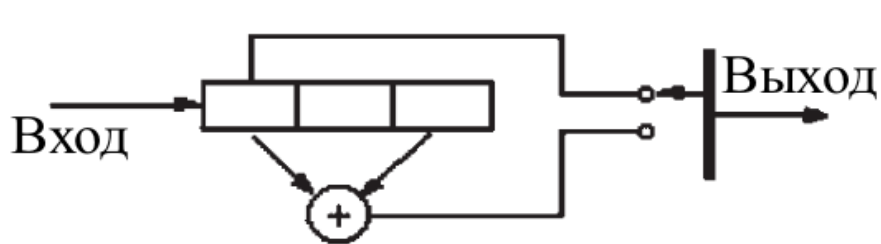
$$B_j(X)=b_0+b_1X+b_2X^2+\dots$$

$$B_j(X)=G_j(X)A(X)$$

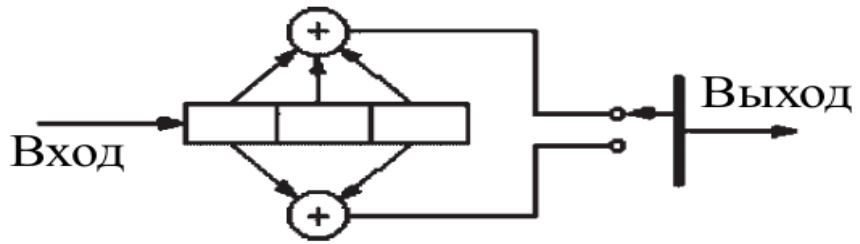
$G$  – порождающие полиномы

Систематические коды не меняют информационную последовательность.

Несистематические меняют. Примеры.



$$G_1(X)=1; G_2(X)=1+X^2$$



$$G_1(X)=1+X+X^2$$
$$G_2(X)=1+X^2;$$

Для кода финка с  $S=0$

$$G_1(X)=1$$

$$G_2(X)=1+X;$$

# Код Хэмминга

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	
1	X		X		X		X		X		X		X		X		X		X		X	1
0		X	X			X	X			X	X			X	X			X	X			2
1				X	X	X	X					X	X	X	X					X	X	4
0								X	X	X	X	X	X	X	X							8
0																X	X	X	X	X	X	16

Каждую последовательность суммируем по модулю 2 (операция xor), получая код:

$$0+0+1+0+0+0+0+1+1+1+1=1$$

$$0+0+0+0+1+0+0+1+1+1=0$$

$$0+1+0+0+0+0+0+1+0+1=1$$

$$0+0+1+0+0+0+0+1=0$$

$$0+1+1+1+0+1=0$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	



1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1	0	0	1	1	0	0	0	0	1	<del>1</del>	0	0	0	1	0	1	1	1	0	1	
X		X		X		X		X		X		X		X		X		X		X	1
	X	X			X	X			X	X			X	X			X	X			2
			X	X	X	X					X	X	X	X					X	X	4
							X	X	X	X	X	X	X	X							8
															X	X	X	X	X	X	16

1+0+1+0+0+1+0+1+1+1+1
=
1 1

0+0+0+0+1+1+0+1+1+1
=
1 2

1+1+0+0+0+0+0+1+0+1
=
0 4

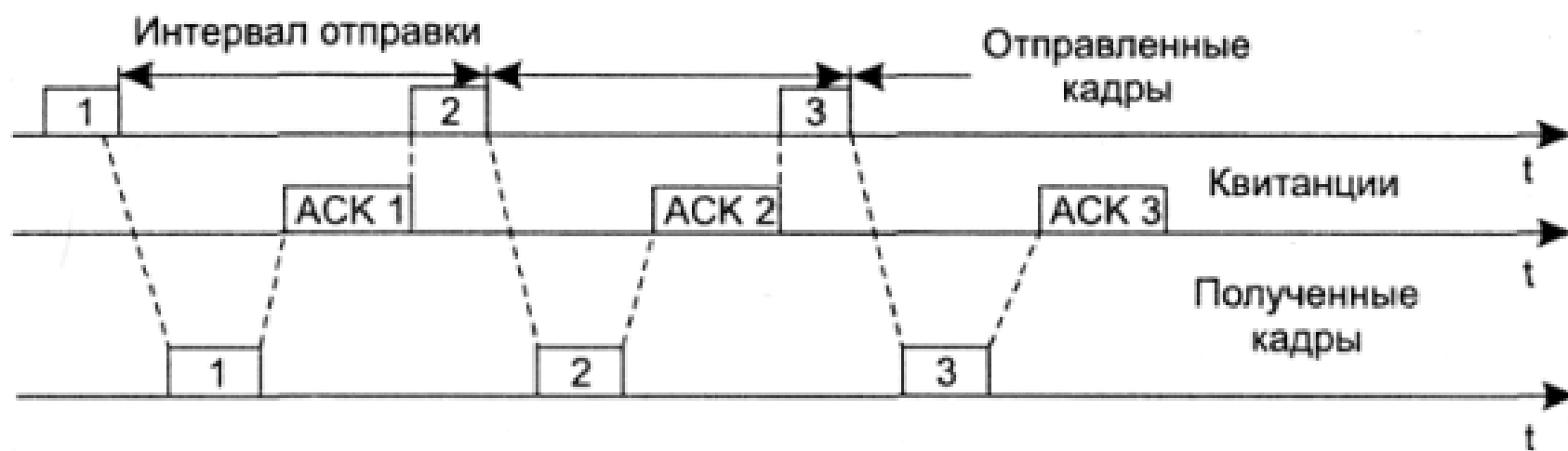
0+0+1+1+0+0+0+1
=
1 8

0+1+1+1+0+1
=
0 16

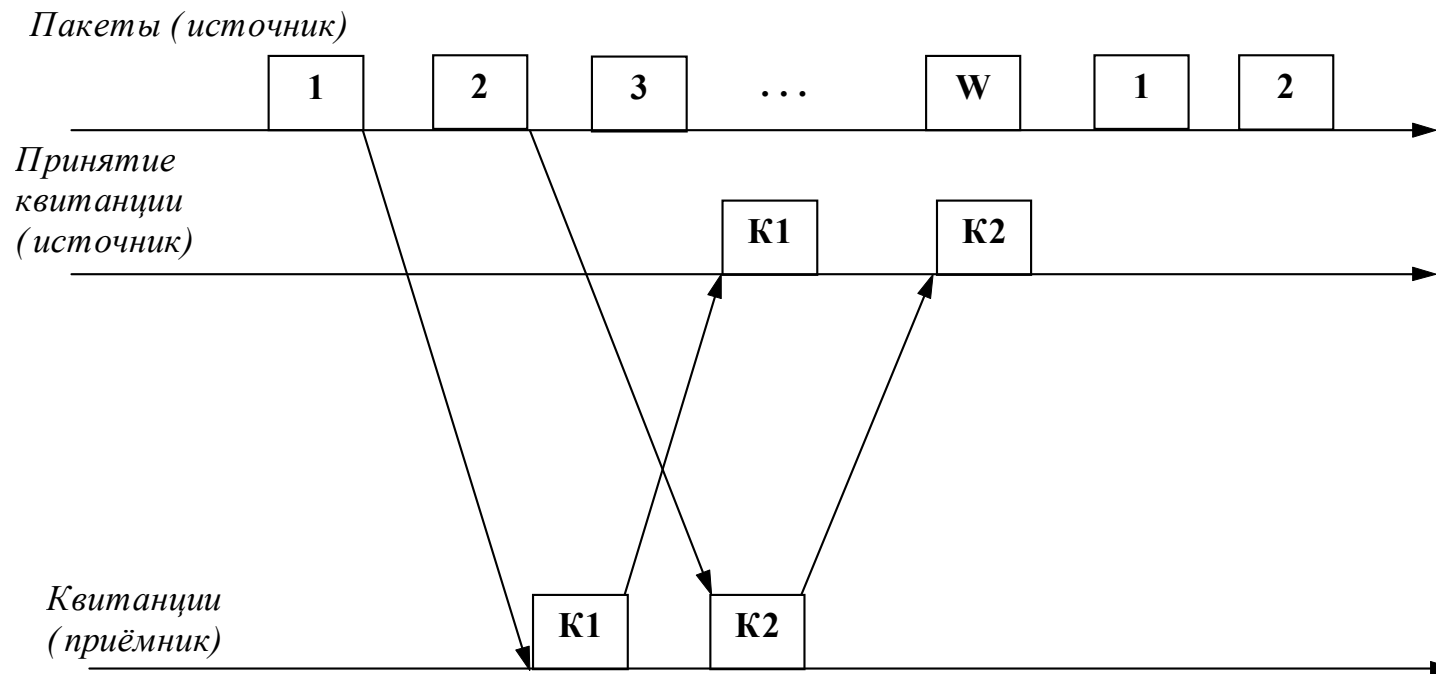
11

- Циклический избыточный контроль (Cyclic Redundancy Check, CRC)
- Этот метод является в настоящее время наиболее популярным методом контроля в вычислительных сетях (и не только в сетях, например, этот метод широко применяется при записи данных на диски и дискеты).
- Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель  $R$ .
- Обычно в качестве делителя выбирается семнадцати- или тридцати трехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт)- CRC16, или 32 разряда (4 байт)- CRC32.
- При получении кадра данных снова вычисляется остаток от деления на тот же делитель  $R$ , но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на  $R$  равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.
- Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету.
- Метод CRC обнаруживает все одиночные и двойные ошибки, а также ошибки в нечетном числе бит. Метод обладает также невысокой степенью избыточности. Например, для кадра размером в 1024 байт контрольная информация длиной в 4 байт составляет только 0,4 %.

- **Методы борьбы с искажениями** основаны не переотправке поврежденных кадров отправителем. Отправитель ждет от получателя квитанции о том, что данные правильные, каждый кадр имеет нумерацию, если квитанция по кадру не подтверждает правильности приема, отправитель снова отправляет нужный кадр. При этом если положительная квитанция не получена в течении какого то времени, кадр считается утерянным. Существует возможность выдачи групповой квитанции, на несколько кадров. Метод с простоями — отправка следующего кадра после получения подтверждающей квитанции. Метод скользящего окна — интенсивная отправка кадров без получения подтверждения какое то время, при этом кадры хранятся в буфере отправителя или узла, пока не получено подтверждение.



## 2. Метод "скользящего окна"



**W** - размер окна - количество кадров, которые разрешается передавать без получения квитанции

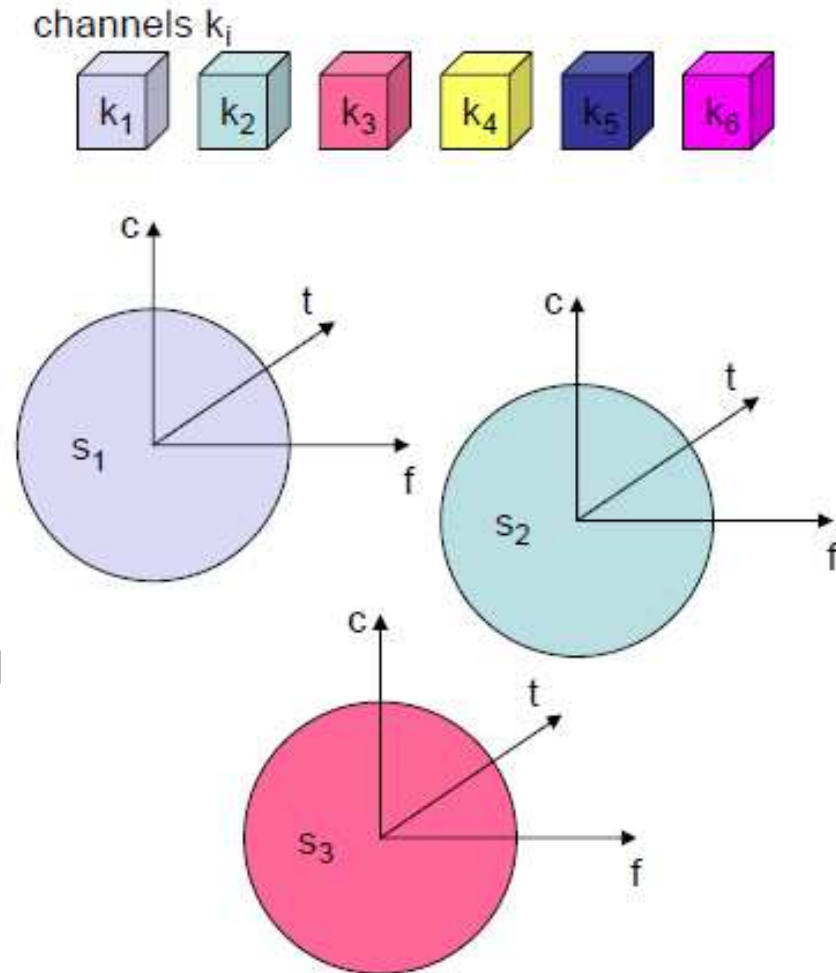
# Мультиплексирование

Мультиплексирование  
каналов ( $k$ )  
в четырех измерениях :

- пространство ( $s$ )
- время ( $t$ )
- частота ( $f$ )
- код ( $c$ )

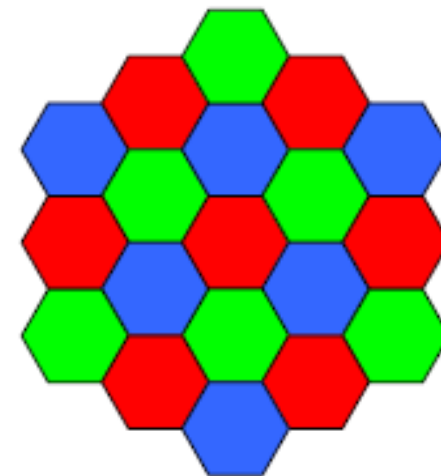
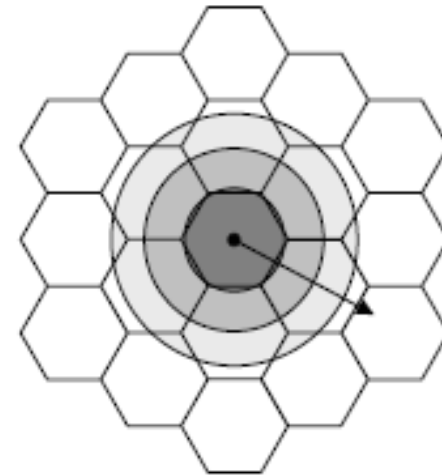
Цель: множественный доступ  
к разделяемой среде.

Пример: радио передача

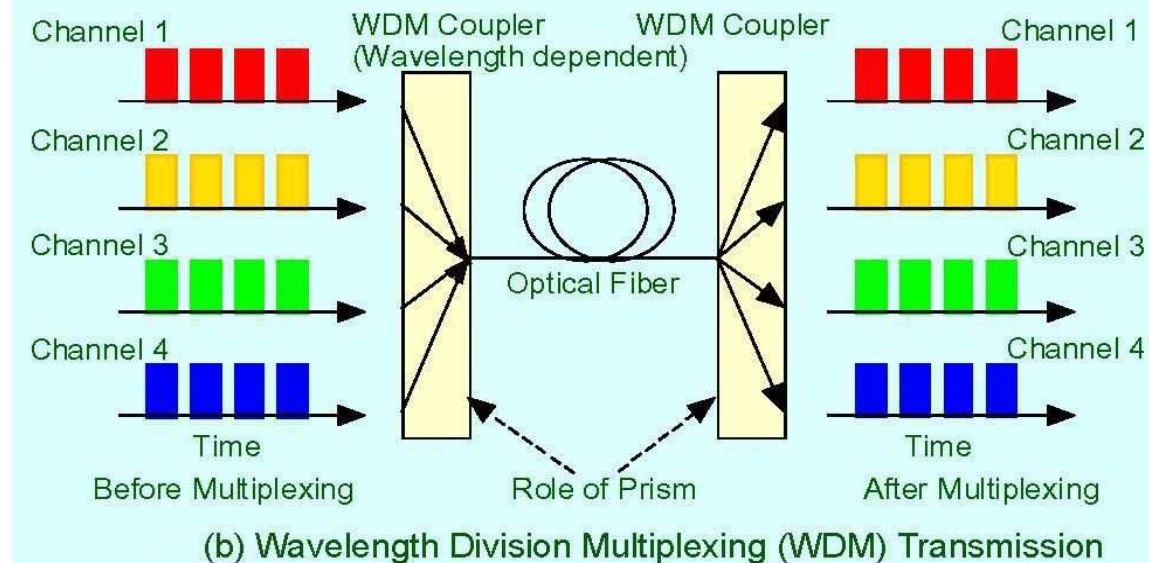
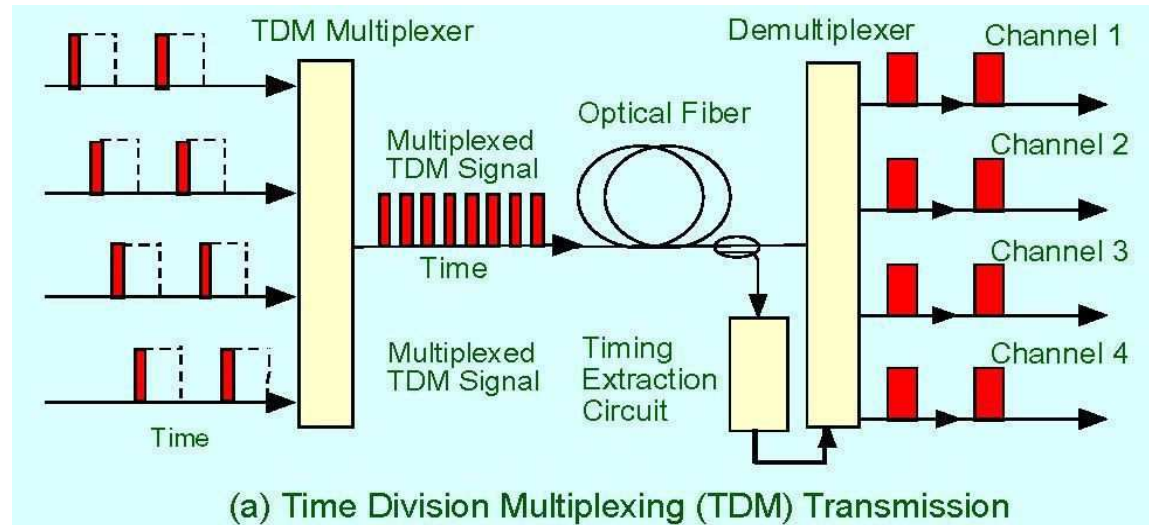


# Пример пространственного мультиплексирования – сотовая сеть

- Упрощенная шестигранная модель
- Области распространения сигнала:  
частота используется многократно при определенном расстоянии между базовыми станциями
- Проблема окрашивания графа
- Интерференция из соседних ячеек (другого цвета) может управляться фильтрами передачи и приема



# Мультиплексирование TDM, FDM, CDMA, WDM





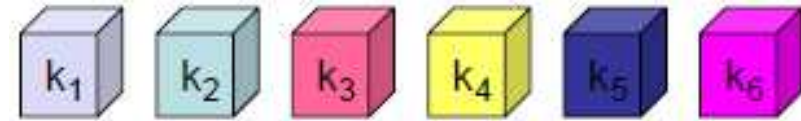
# **FDMA - множественный доступ с частотным разделением**

Стандарт FDMA широко используется как в традиционных аналоговых системах сотовой связи, так и в современных цифровых системах (как правило в сочетании с другими методами).

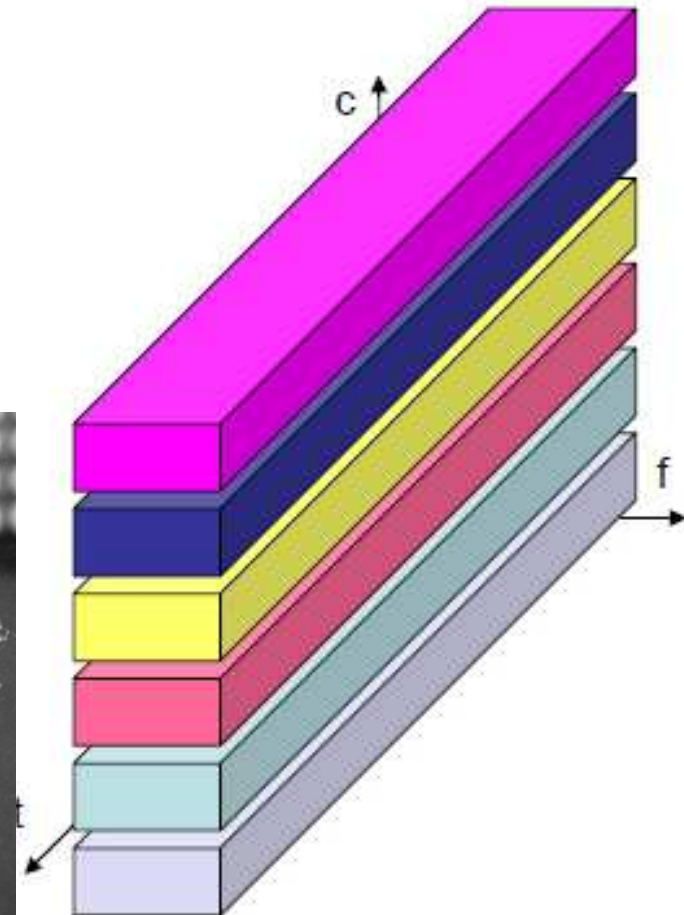
Из всего доступного диапазона каждому абоненту выделяется своя полоса частот, которую он может использовать все 100% времени. Таким образом не временной фактор, а только лишь различия в частоте используются для разделения (дифференциации) абонентов. Подобный подход имеет заметное преимущество: вся информация передается в "реальном времени", и абонент получает возможность использовать всю полосу пропускания, выделенного ему сегмента. Ширина полосы сегмента может варьироваться в зависимости от используемой системы связи.

# Code Division Multiplex (CDM)

- Каждый канал имеет уникальный код
  - Все каналы используют один спектр в одно время + эффективное использование диапазона частот
  - + нет согласования или синхронизации
  - + трудно подключиться
  - + почти невозможны коллизии
  - низкая скорость передачи
  - более сложное восстановление сигнала
- Пример: UMTS
- Расширенный спектр
  - U. S. Patent 2 292 387, K Hedy. Markey (a.k.a. Lamarr Или Kiesler) и George Antheil (1942)



9.11.1914 – 19.01.2000





# Идея которую предложила Ламар

- Если дистанционно сообщать координаты цели управляемой торпеды по одной частоте, то враг может легко перехватить сигнал, **заглушить его узкополосным сигналом** или перенаправить торпеду на другую цель, а если **использовать на передатчике случайный код, который будет менять канал передачи**, то можно синхронизировать такие же частотные переходы и на приёмнике. Такая смена каналов связи гарантирует безопасную передачу информации. До того времени псевдослучайные коды использовались для шифровки информации, передаваемой по неменяющимся открытым каналам связи. Здесь же был сделан шаг вперёд: секретный ключ стал использоваться для быстрого изменения каналов передачи информации.

# Получение патента в 1942

- В августе 1942 года Ламарр и Антейл получили патент на изобретение «Системы секретной связи» (англ. Secret Communication System), зарегистрированный под номером US2292387A. Патент, сроком действия до 1955 года, описывает секретные системы связи, включающие передачу ложных каналов на разных частотах. Однако американский флот тогда отверг этот проект из-за сложности реализации; ограниченно использовать его начали лишь в 1962 году, и, таким образом, отчислений за него изобретатели не получили. Но спустя полвека этот патент лёг в основу связи с расширенным спектром, которая стала использоваться повсеместно, от мобильных телефонов до Wi-Fi.
- 
- В 1997 году Хеди Ламарр была официально награждена за своё изобретение, но актриса не присутствовала на церемонии и лишь передала аудиозапись своего приветственного слова

Расширение спектра скачкообразной перестройкой частоты. Медленное расширение спектра. Частота переключения на канал канал другой частоты меньше частоты передачи информационного сигнала. В данном случае в два раза.



Быстрое расширение спектра. Частота переключения канала больше частоты передачи информационного символа. В данном случае в четыре раза. При передаче символа канал успевает смениться четыре раза. Менее подвержен узкополосной помехе, так как содержание символа в разных частотных каналах, менее подвержен помехе межсимвольной интерференции, так как переотраженный задержанный сигнал приходит на предыдущей частоте не мешающей текущему прямому сигналу на новой частоте.



## **CDMA - множественный доступ с кодовым разделением**

Каналы трафика при таком способе разделения среды создаются присвоением каждому пользователю отдельного кода, который распространяется по всей ширине полосы. В данном случае не существует временного разделения, и все абоненты постоянно используют всю ширину канала. Нужно заметить, что полоса частот, выделяемая для организации одного канала, очень широка. Вещание абонентов накладываются друг на друга, но поскольку их коды отличаются, они могут быть легко дифференцированы.



# Пример

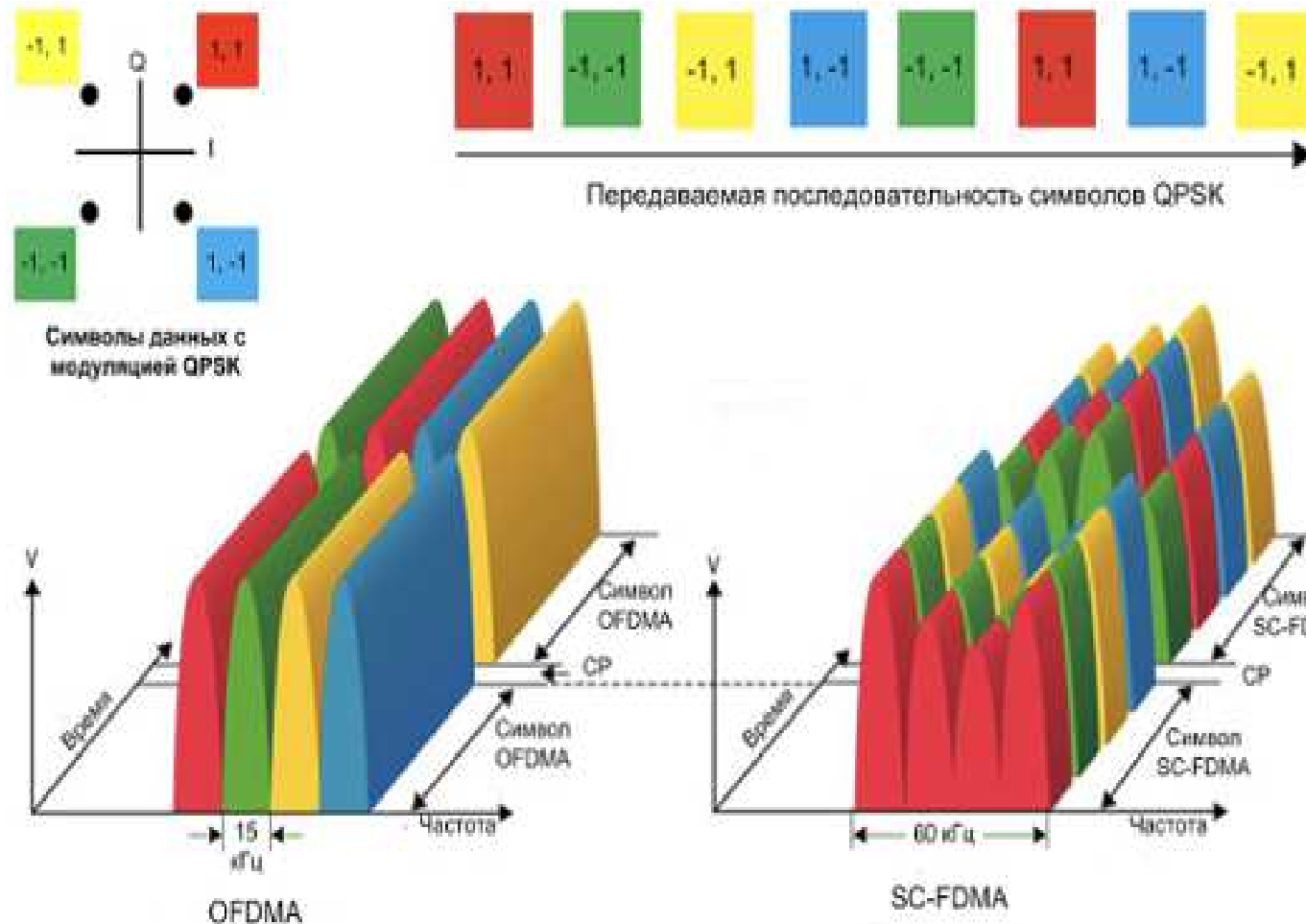
- Пусть в сети работает четыре узла: А, В, С и D. Каждый узел использует следующие значения расширяющей последовательности. Кодируем положительным и отрицательным значением 0 :-1, 1: +1.
- A: 0 1 0 1 0 1 0 1
- B: 1 0 1 0 0 1 0 1
- C: 1 0 0 1 1 0 0 1
- D: 1 1 1 1 1 1 1 1

A: -1 +1 -1 +1 -1 +1 -1 +1  
B: +1 -1 +1 -1 -1 +1 -1 +1  
C: +1 -1 -1 +1 +1 -1 -1 +1  
D: +1 +1 +1 +1 +1 +1 +1 +1

- Пусть теперь каждый из 4-х узлов независимо от других передает в сеть один бит исходной
- информации: узел A  $\rightarrow$  1, узел B  $\rightarrow$  0, узел C  $\rightarrow$  0, узел D  $\rightarrow$  1. Фактически каждая последовательность умножается на свой бит, только вместо 0 мы берем -1.

## Матрично передача выглядит так:

- $[1 \ -1 \ -1 \ 1] * \{-1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1\}$
- $\{1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1\}$
- $\{1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1\}$
- $\{1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1\}$
- $= \{-2 \ 4 \ 0 \ 2 \ 0 \ 2 \ 2 \ 0\}$
- $\text{Info}_{abcd} * M_{abcd} = \text{Signal}$
- $\text{Info}_{abcd} * M_{abcd} * M_{abcd}^T = \text{Signal} * M_{abcd}^t$
- $\text{Info}_{abcd} = \text{Signal} * M_{abcd}^t (M_{abcd} * M_{abcd}^t)^{-1}$
- При ортогональности кодов A, B, C, D мы получим единичную матрицу умноженную на величину обратную длине вектора, которая для нашего 8 битного расширяющего слова равна 8.
- $\text{Info}_{abcd} = \text{Signal} * M_{abcd}^t * 1/8$ . При этом для получения каждого инфо бита мы должны умножить на соответствующую кодовую последовательность скалярно.
- $S \times A = (-2 + 4 \ 0 + 2 \ 0 + 2 + 2 \ 0) \times (-1 + 1 \ -1 + 1 - 1 + 1 \ -1 + 1) * 1/8 = 1$ .
- $S \times B = (-2 + 4 \ 0 + 2 \ 0 + 2 + 2 \ 0) \times (+1 \ -1 + 1 \ -1 \ -1 + 1 \ -1 + 1) * 1/8 = -1$ .



# OFDMA и SC- FDMA

Хотя OFDMA имеет хорошие показатели по борьбе с межсимвольной интерференцией, но имеет проблемы с PAPR (Peak to Average Ratio), высокое отношение пиковой мощности к средней, что требует особых блоков питания с низким КПД на базовых станциях и повышенному энергопотреблению, при использовании на мобильных устройствах это приводит к быстрой разрядке мобильных БП, потому в LTE на прием мобильных устройств используется OFDMA, на передачу SC-FDMA( с худшими показателями на межсимв. интерф())