

# ОПТИЧНИ СЪОБЩИТЕЛНИ ЛИНИИ

---

## 1. Оптични влакна

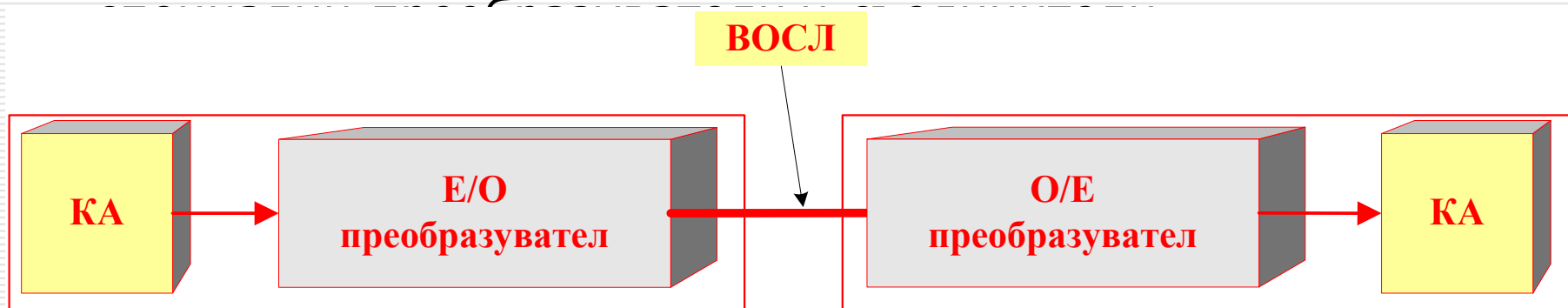
- 1.1. Структурна схема на оптичното пренасяне
- 1.2. Физически принципи на оптичното предаване
- 1.3. Структура на оптичните влакна
- 1.4. Видове оптични влакна

## 2. Оптични кабели

- 2.1. Конструкция на оптичния кабел
- 2.2. Изисквания към оптичните кабели
- 2.3. Видове оптични кабели
- 2.4. Предимства на оптичните кабели
- 2.5. Параметри на оптичните линии
- 2.6. Оптични излъчватели, приемници и съединители

# Структурна схема на оптичното пренасяне

- ❑ **Влакнесто-оптични съобщителни линии** - направляваща среда за светлинни сигнали с честота в оптичния диапазон
- ❑ Състав - **оптични влакна** (изключително тънки нишки от съвършено чисто стъкло)
- ❑ *Необходим преход* от електрически към оптически сигнали при предаване на информация - реализира се със



# Структурна схема на оптически преносна система

---

- ❑ КА (*комуникационна апаратура*) - формира сигналите като последователни електрически импулси
- ❑ Е/О (*електрооптичен преобразувател*): полупроводникови квантови генератори или светодиоди - преобразуват електрическата енергия в светлинна
- ❑ ВОСЛ (*vlakнесто-оптична съединителна линия*) - изпраща светлинни импулси до оптичния приемник
- ❑ О/Е (*отоелектрически преобразувател*) - превръща светлинните сигнали в електрически импулси
- ❑ *Оптични приемници* - фотодиоди, фоторезистори, фототранзистори и др.

## Физически принципи на оптичния пренос

---

- ❑ *Видима светлина* - дължина на вълните:  $380\text{nm}$  (виолетово)  $780\text{ nm}$  (червено).
- ❑ ВОСЛ - използва сигнали в близкия инфрачервен обхват с дължина на вълните  $\lambda=0,85\div 1,55\mu\text{m}$ .
- ❑ Излъчване на оптични вълна - при  $\lambda=1\ \mu\text{m}$  честотата е  $300\text{ THz} = 300000\text{ GHz}$
- ❑ Намалване светлинните загуби - *дебелината* на оптичното влакно трябва да има порядъка на *дължината на вълната*  $\lambda$ .
- ❑ *Диаметър на оптичните влакна* -  $8\div 50\mu\text{m}$  – изготвят се по вакуумни технологии → относително висока цена; голяма чистота и гъвкавост на стъклени нишки; много малко затихване

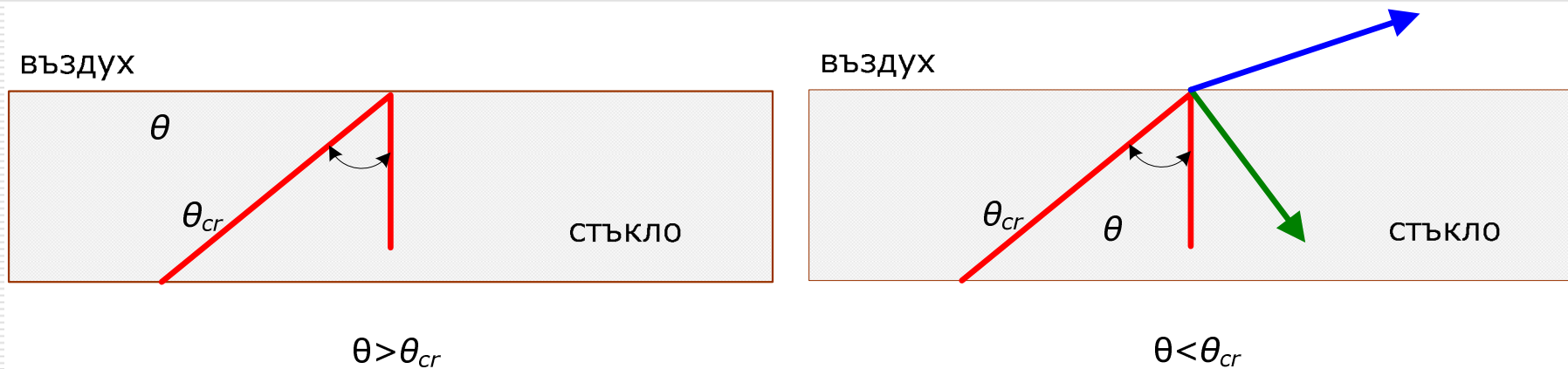
## Физически принципи на оптичния пренос

---

- ❑ *Еднопосочното предаване на сигнали* - за дуплексна връзка са необходими два отделни канала с противоположна посока
- ❑ *Диелектричен вълновод* от оптични влакна за предаване оптичните сигнали
- ❑ *Свойства на материалите* за оптични влакна - трябва да имат *голяма* пропускателна способност и *минимално* затихване при разпространението на светлината → използва се *силикатно стъкло* или *свръх чисто синтетично кварцово стъкло*
- ❑ *Усъвършенстване на технологията за изработване на влакна*  
→ увеличаване на *чистотата* на стъклото
  - намалява километричното затихване от  $20 \text{ dB/km}$  (1970г.) до  $0,5 \text{ dB/km}$  (днес)
  - примес от метали - ↓ само с  $10 \text{ mg/t}$  → ↑ затихването с  $1 \text{ dB/km}$

# Физически принципи на оптичния пренос

- *Предаване на границата между две среди – въздух и стъкло*



- **Пречупване на светлинния лъч**

- ❖ различни *скорости* на разпространение на лъча, което зависи от различните дължини на вълната
- ❖ **коефициент на пречупване  $\eta$**  – отношение на тези скорости (стъкло/въздуха  $\eta \approx 1,5$ )

# Материали за изготвяне на оптични влакна

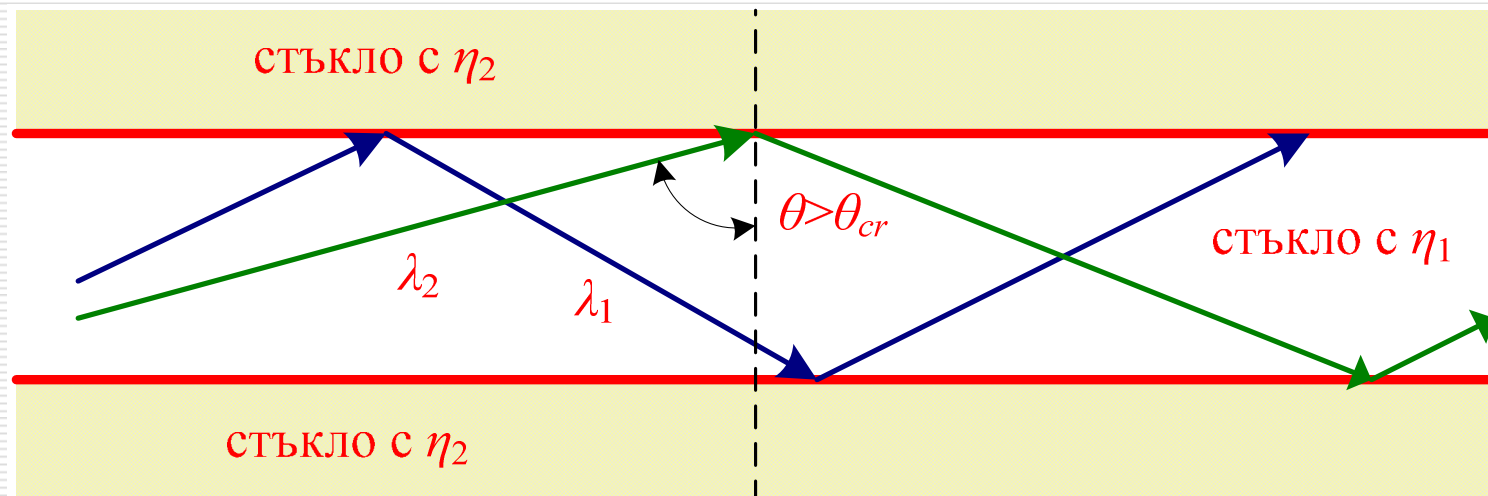
---

## □ **Отражение на светлинния лъч**

- ❖  $\theta$  - **ъгъл на падане** на границата между двете среди
- ❖ *поглъщане на светлината* за малки ъгли на падане - дължи се на наличието на примеси на вода (хидроксилни йони) и на метали (желязо, мед, хром, манган, никел, кобалт и др.)
- ❖ *пълно вътрешно отражение* - съществува критичен ъгъл на падане  $\theta_{cr}$  (стъкло/въздуха  $\theta_{cr}=42^\circ$ )

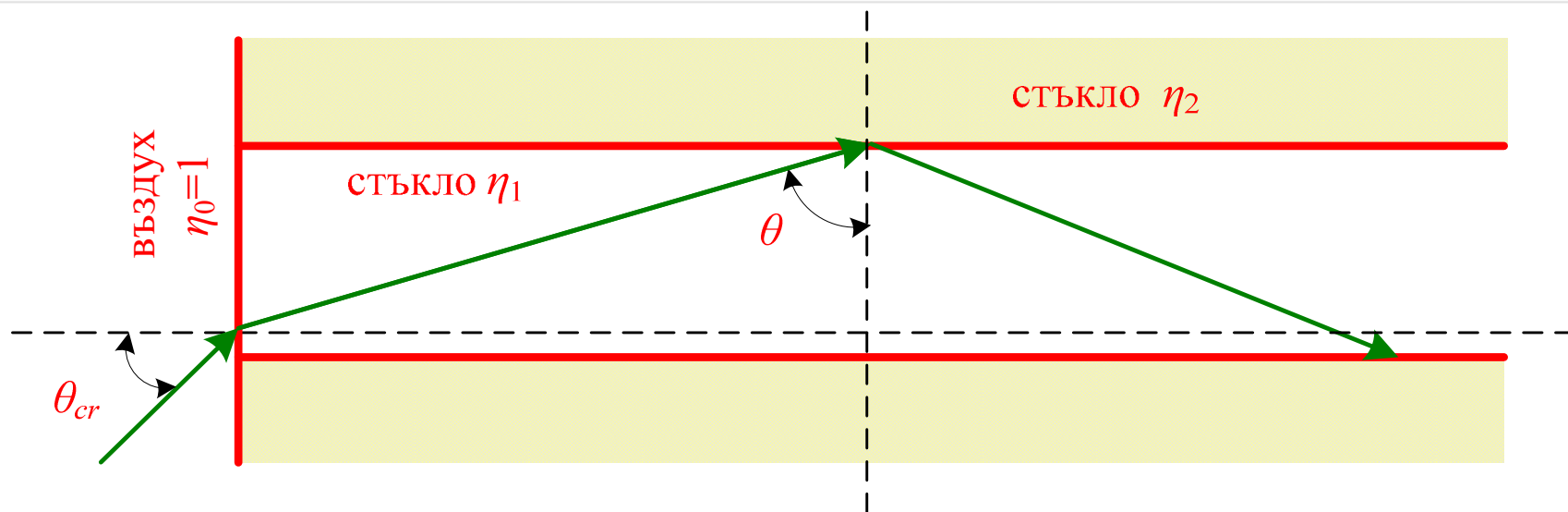
## Структура на оптичните влакна

- Двуслойна структура - сърцевина (core) и обвивка (cladding) от стъкло с различен състав на кварца, която прави коефициента на пречупване на светлината по-голям, когато разликата им е над 10%.
- При ъгъл на падане  $\theta > \theta_{cr}$  светлината се разпространява по дължината на сърцевината чрез *последователни пълни вътрешни отражения*



## Структура на оптичните влакна

- Граница между двете среди се държи като огледало, което направлява светлинния лъч към центъра на сърцевината, като не му позволява да излезе извън обвивката
- Всяка светлинна вълна с ъгъл на падане  $\theta_i > \theta_{cr}$  се разпространява в сърцевината, докато лъчите с  $\theta_i < \theta_{cr}$  се пречупват в обвивката и се губят
- $\theta_{cr}$  ъгъл на приемане



## Структура на оптичните влакна

---

- $\theta_{cr}$  ъгъл на приемане

$$\frac{\sin \theta_{cr}}{\sin(90 - \theta)} = n_1$$

$$\sin \theta_{cr} = n_1 \cos \theta = n_1 \sqrt{1 - \sin^2 \theta}$$

- NA - числена апертура (*numerical aperture*- важен параметър на оптичното влакно, зависи коефициентите на пречупване

$$NA = \sin \theta_{cr} = \sqrt{n_1^2 - n_2^2}$$

- Представя възможността на влакното за приемане на светлина - количеството светлинна  $\sim$  с квадрата на NA

## Видове оптични влакна

---

- Според разпространяваната светлина:
  - ❖ с една вълна (мода)- едномодово влакно (*monomode fiber*)
  - ❖ едновременни вълни с различни дължини -многомодов режим (*multimode fiber*).
- **Едномодов режим**
  - ❖ дебелината на сърцевината  $\sim$  дължината на вълната
  - ❖ оптична плътност- на сърцевината  $>$  от на обвивката  $\rightarrow$  малък диаметър ( $6\div 8\mu m$ ), което създава технологични трудности при изготвяне на влакната.
  - ❖ оптични излъчватели в едномодовите световоди - лазерни устройства, което оскъпява едномодовите ВОСЛ
  - ❖ широко разпространение в обществените телефонни мрежи - широка честотна лента и малко затихване

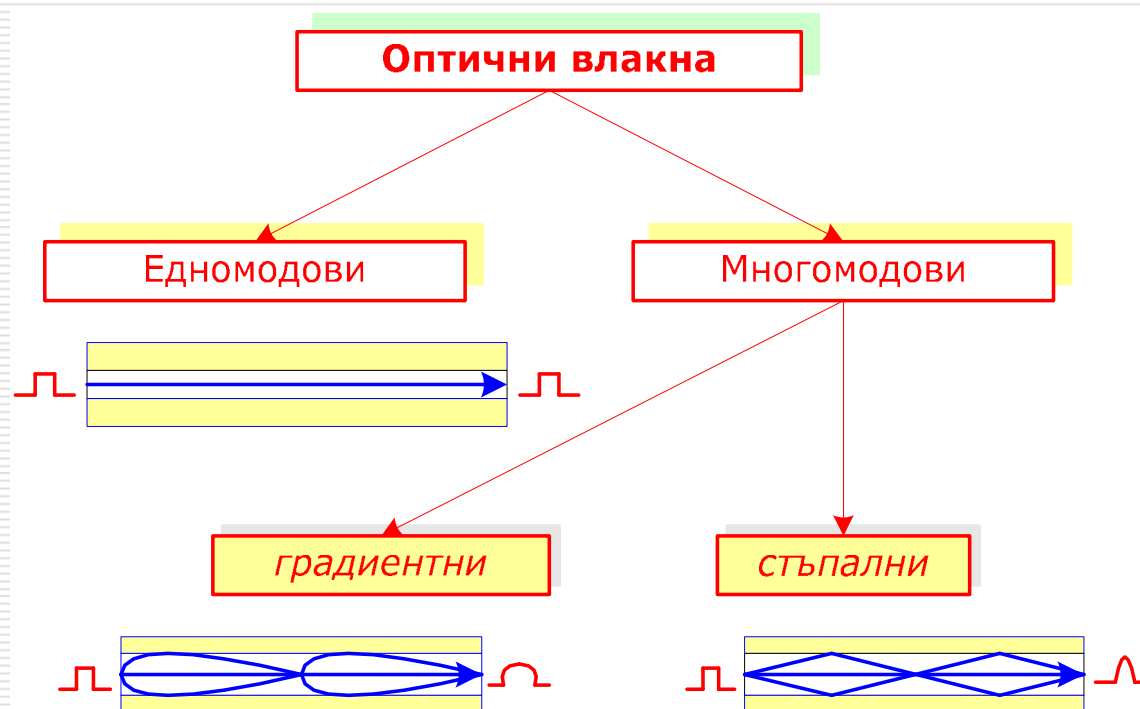
## Видове оптични влакна

---

- **Многомодов режим** – разпространение на вълни с пълно вътрешно отражение по различни лъчи
  - ❖ многомодови оптични влакна - захранват се от светодиоди и имат с по-голяма дебелина на сърцевината ( $50\mu m$ ) → по-евтини, по-здрави, но с по-ограничени параметри
  - ❖ диаметър на ядрото може да бъде  $> 50\mu m$ , обвивка (125-150)  $\mu m$
  - ❖ използване се за пренос на къси разстояния с пропускателна способност около 20 MHz (локални компютърни мрежи)
- Видове многомодови влакна - **градиентни** (*graded-index fiber*) и **стъпални**

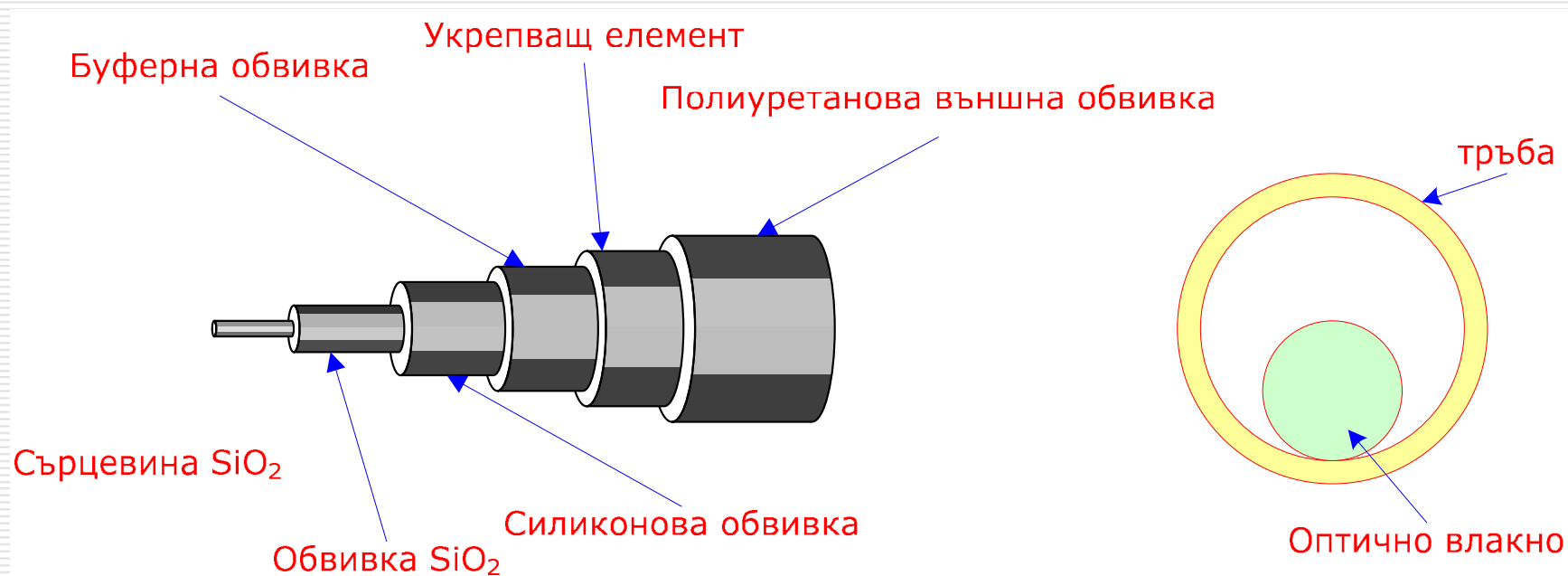
## Видове оптични влакна

- ❖ *градиентни* - плавна промяна на коефициента на пречупване на светлината от от центъра към периферията; широка лента на пропускане и малко затихване; скъпи със сложната технология на изготвяне
- ❖ *стъпални* - постоянен коефициент на пречупване за сърцевината, които се променя в обвивката стъпално



## Конструкция на оптичния кабел

- ❑ Оптичният лъч се разпространява надлъжно на сърцевината, чрез частични пълни вътрешни отражения
- ❑ Стъклената обвивка има друг химически състав и повишава здравината на конструкцията



## Конструкция на оптичния кабел

---

- ❑ *Силиконова обвивка* - защитно укрепващо покритие с дебелина  $125\mu m$ ; предпазва влакното от влага и механични повреди
- ❑ *Полимерна буферна обвивка* от подходящи материали (полиамид, флуоропласт, полипропиоен, полиетилен с голяма плътност) - дебелина  $(200-250)\mu m$
- ❑ Оптичното влакно се разполага по различен начин в кабела
  - ❖ кабели с влакно, поставено свободно в тръба от флуоропласт с по-голям диаметър
  - ❖ кабели с влакното, поставено плътно в двуслойна полимерна тръба с твърда вътрешност и мека външност, например от пенопласт
  - ❖ строителна дължина на влакната е от  $1250 m$  и  $2500 m$

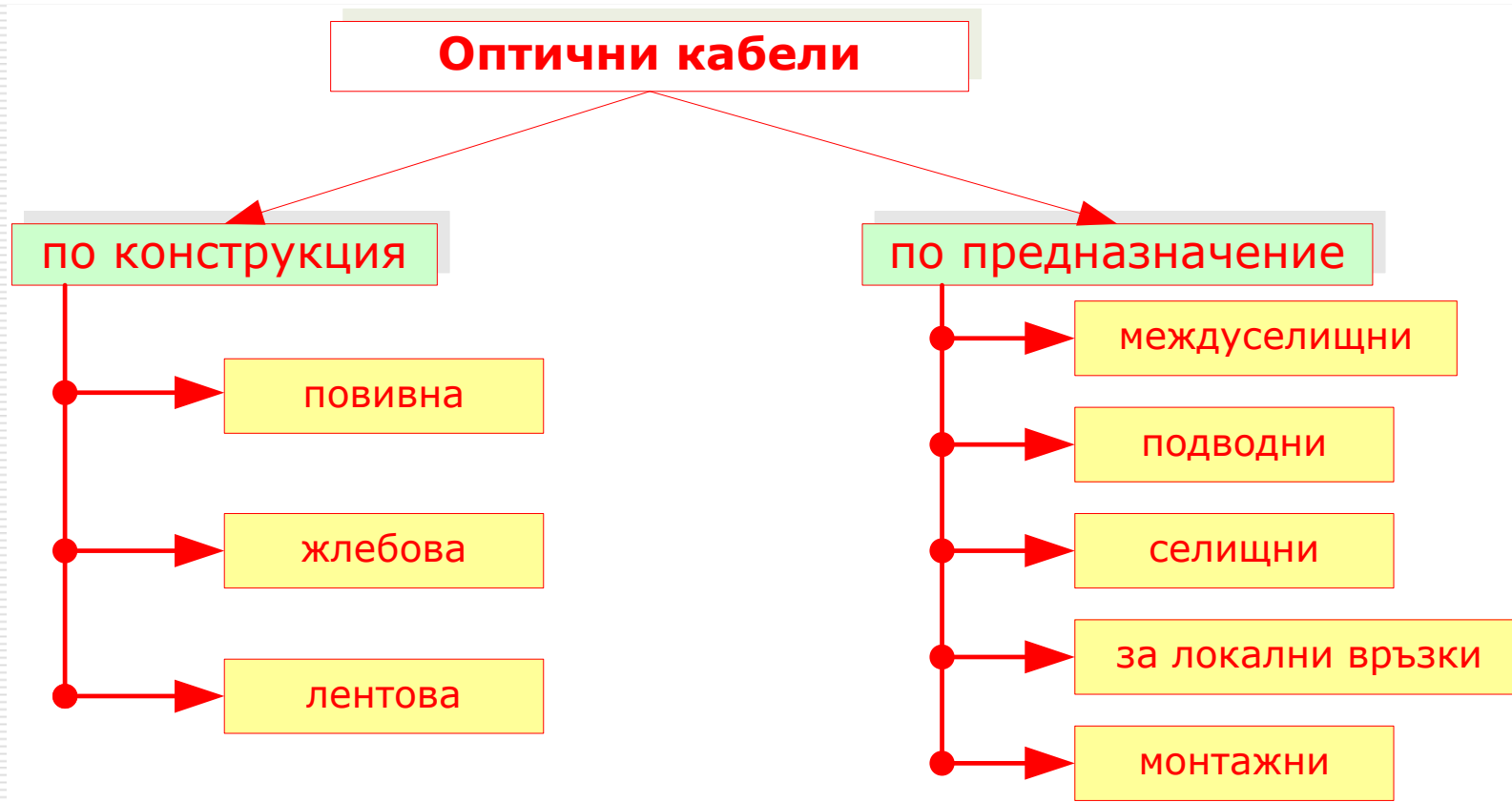
## Изисквания към оптичните кабели

---

- ❑ *Оптичен кабел* - сноп от оптични влакна, свързани в определена система и обвити със защитни обвивки, които се полага в кабелни шахти
- ❑ Изисквания към оптичните кабели
  - ✓ добри преносни параметри
  - ✓ висока якост на опън и натиск
  - ✓ гъвкавост, която не нарушава оптичните свойства на снопа от влакна
  - ✓ влагонепроницаемост, химична и термоустойчивост
  - ✓ простота в монтажа и полагането на кабелите

# Видове оптични кабели

- Според предназначението и според конструкция



## Видове оптични кабели

---

- ❑ **Междуселищни** - за междуселищни комуникации с цифрови мултиплексни системи
  - ❖ едномодови или многомодови градиентни оптични влакна с дебелина  $50\mu m$  и обвивка  $125\mu m$
  - ❖ в кабела се поместват 4, 8, 10 или 12 оптични влакна, по които се пропускат лъчи с дължина вълната  $(1,3-1,6)\mu m$
  - ❖ оптоизлъчвател - полупроводников лазер, а за приемник-лавинен фотодиод или пинфотодиод
  - ❖ затихване  $(0,5-3) dB/km$
  - ❖ дължина на регенераторните участъци  $(15\div 36) km$
- ❑ **Подводни** - за преодоляване на големи водни басейни; специално укрепени за по-голяма здравина; отговарят на строги изисквания по оптични параметри, физикохимически свойства и дълготрайност

## Видове оптични кабели

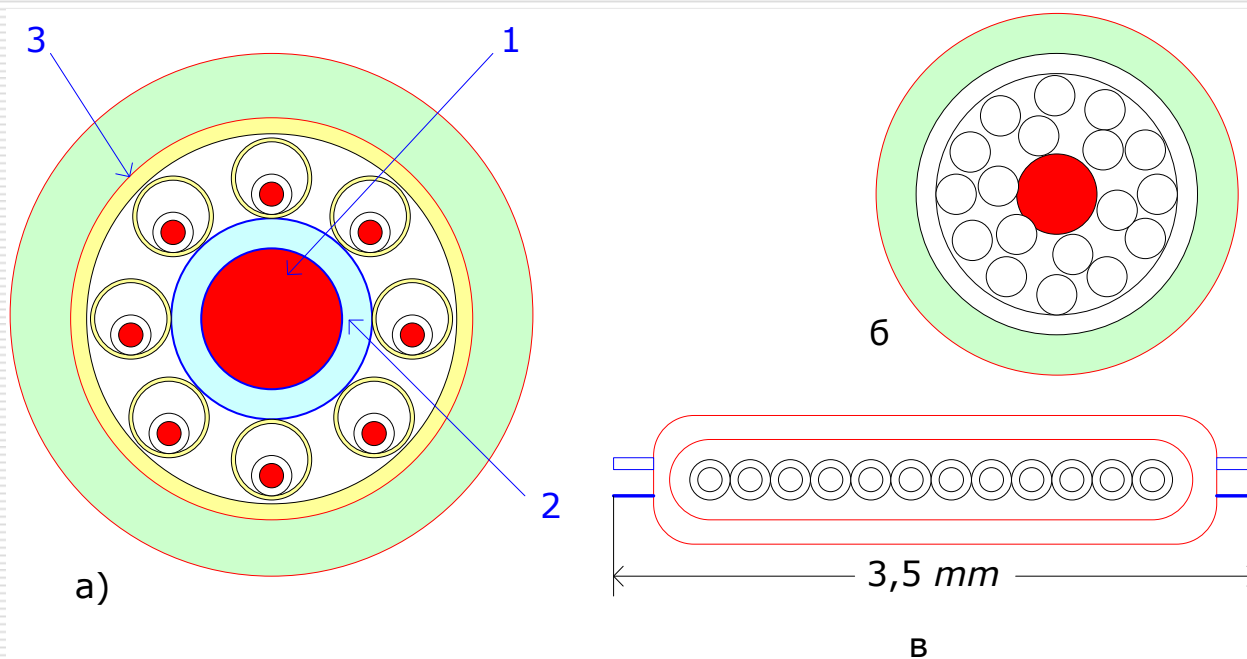
---

- ❑ **Селищни** - съединителни линии между комутационните центрове в дадено селище; голям брой оптични влакна за къси разстояния (2-15) *km*; дължина на вълните (0,85-1,3)  $\mu m$  ; затихване (0,8-3,5) *dB/km*
- ❑ **За локални връзки** - LAN мрежи с 2, 4 или 6 стъпални многомодови силикатни световоди с размери 50/125 или 200/400  $\mu m$ ; дължина - няколко километра; затихване (5-10) *dB/km*
- ❑ **Монтажни** - за връзка на апаратурата елементите в комуникационните мрежи; плоско оформени (2-12) оптични влакна с размери 50/125  $\mu m$  или 200/400  $\mu m$  и дължина на вълната 0,85  $\mu m$

# Видове оптични кабели

## □ Кабели с **повивна конструкция**

- ❖ фиг. (а) с външен диаметър  $14,4 \text{ mm}$  - прилага се в селищни кабели; кабелен сноп от 8 тръби с влакна; централен елемент от многожилно въже армиращо 1, покрито с поливинилхлоридна обвивка 2; снопът е обвит с поясна изолация от  $2 \text{ mm}$  диелектрична лента 3;



## Видове оптични кабели

---

- ❖ фиг. (б) с външен диаметър 8 *mm* – повивна конструкция с 18 оптични влакна; центриращо медно жило и оловна влагозащитна обвивка; голям брой на влакната - 42, 126 и по-големи
- Кабели с **лентова конструкция** - фиг. (в)
  - ❖ изграждат се от двуслойни полиетиленови ленти, в които се пресоват 12 влакна
  - ❖ използват се по-рядко

## Предимства на оптичните кабели

---

- ❑ Голям капацитет - пренасят голям обем от информация с голяма скорост
- ❑ Малко затихване - генератори на разстояние (30-100) *km*
- ❑ Нечувствителност към външни електромагнитни влияния - няма смущение/преслушване → могат да се окачват на високоволтовите линии на електропроводите
- ❑ Няма къси съединения в оптичната линия, т.е няма чести повреди
- ❑ Компактност на оптичните кабели - малки размери и тегло; не са метални
- ❑ Относително ниска стойност
- ❑ Дълъг жизнен цикъл - над 30г.

## Параметри на оптичните линии

□ **Затихване** (*attenuation*) -  $\alpha = \alpha_c + \alpha_k$

➤ собствени загуби  $\alpha_c$  на оптична мощност - определят се от чистотата на стъклото

❖  $\alpha_c = \alpha_n + \alpha_p$  - състоят от загуби на поглъщане  $\alpha_n$  и загуби на разсейване  $\alpha_p$

❖  $\alpha_n$  - дължат се на топлинни изменения в коефициента на пречупване на влакното

❖  $\alpha_p$  - дължат се на примесите в материала на световода; зависят обратно пропорционално на дължината на вълната

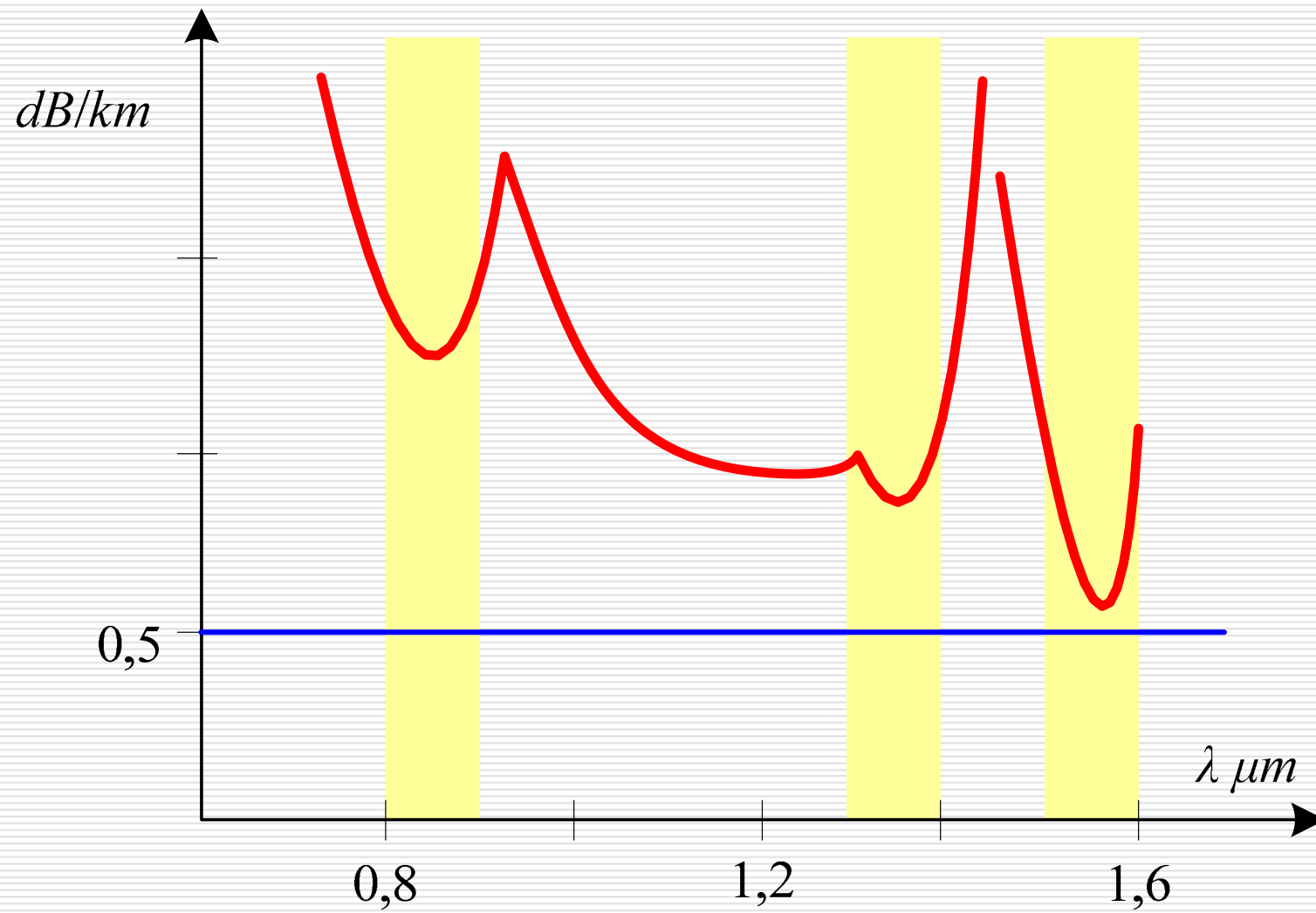
$$\alpha_p = \frac{K_p}{\lambda - 4}$$

## Параметри на оптичните линии

---

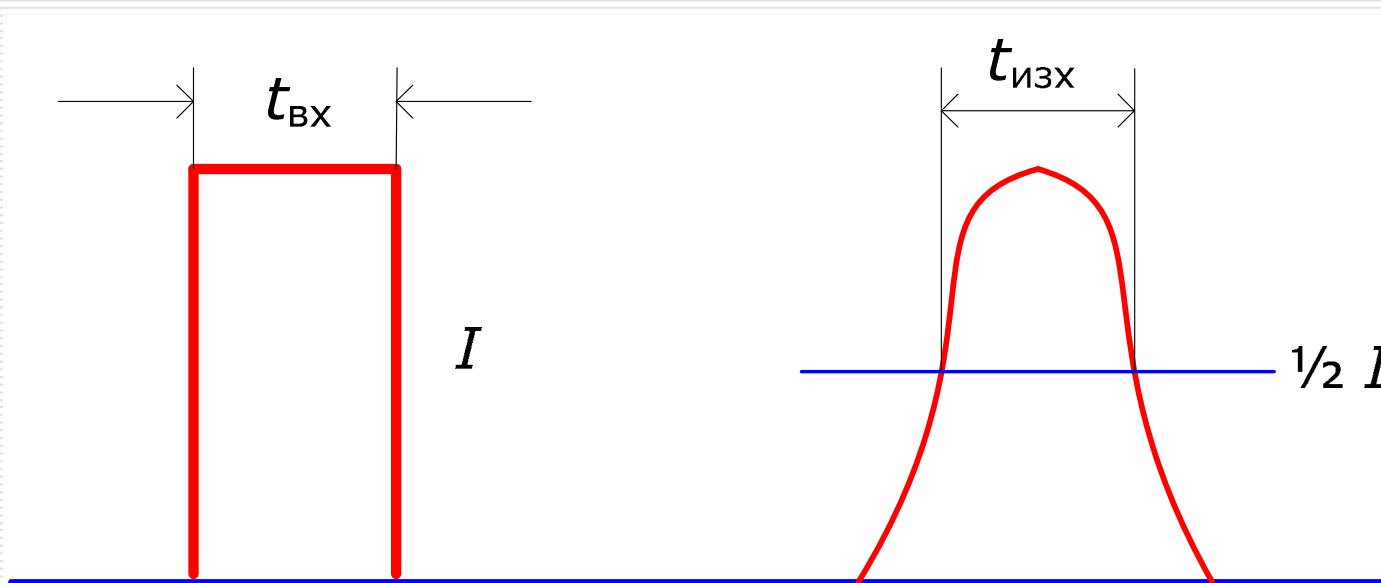
- допълнителни кабелни загуби  $\alpha_k$  -  $\sim 20\%$  от общото затихване; дължат се на усуквания, огъвания, деформации и други механични въздействия при полагане и експлоатация на кабелите → нараства загубата на енергия, вследствие на поглъщане и излъчване в местата на огъване и деформации
- три прозореца на минимално затихване светлинните вълни с дължина  $(0,8 - 1,6) \mu m$  има - около  $0,85$ ;  $1,3$  и  $1,55 \mu m$ ; първото поколение свето- и лазерни диоди излъчват в обхвата  $(0,815-0,950) \mu m$ ; най-малко затихване при третия минимум  $-0,5 \text{ dB/km}$

# Параметри на оптичните линии



## Параметри на оптичните линии

- ❑ **Дисперсия** (*dispersion*) - представлява увеличение на продължителността на предаваните оптични импулси при преминаване през оптично влакно
  - различно удължение за всяка дължина на вълната, тъй като правоъгълни импулси се състоят се от различни хармонични честоти, които се деформират



## Параметри на оптичните линии

- дефинира се като квадратичната разлика между дължината на импулсите на изхода и входа на влакното

$$\tau = \sqrt{t_{\text{изх}}^2 - t_{\text{вх}}^2}$$

- нормира се спрямо дължина от 1 *km* - измерва в *ps/km*
- определя горната граница на скоростта на предаване → предаваните оптични импулси започват да се припокриват и приемника не ги различава
- Физически смисъл на връзката между дисперсията  $\tau$  и честотната лента  $\Delta F$  на пропускане на влакно с дължина 1 *km* - задава ограниченията на честотната лента

$$\Delta F = \frac{0,44}{\tau} \text{ MHz / km}$$

## Параметри на оптичните линии

---

- Основни източници на дисперсия:
  - ❖ Свойствата на материала на оптичното влакно
  - ❖ Направляващите свойства на световодната структура
  - ❖ Различните скорости на разпространение на светлинните вълни с различни дължини

□ *Затихването и дисперсията* налагат по дължината на линията да се разполагат **регенератори** за възстановяване нивото и формата на сигнала - през десетки километра

□ Въпреки *затихването и дисперсията* оптичните кабели имат големи спектрални възможности - дължина на светлинните вълни с честота  $f=1 \cdot 10^{14} \text{ Hz}$ , т.е. ефективната ширина на лентата  $F_k$  е много голяма (3000-6000) GHz

## Параметри на оптичните линии

---

- Приема се, че потенциалната ширина на лентата е до 20000 *GHz* - по едно оптично влакно могат да се организират до 5 милиарда телефонни или няколко милиона телевизионни канала. Тези възможности реално не могат да се използват, поради следните причини:
  - липса на обективна необходимост
  - съществуват практически трудности за реализиране на *излъчватели и приемници* с такова бързодействие в честотната област на *оптоелектричните* и *електрооптичните* преобразуватели

## Оптични излъчватели, приемници и съединители

---

- ❑ **Светодиодни излъчватели** (LED) - при теснолентови системи за предаване на къси разстояния; преходен процес (3-5)  $ns$  → може да се предава със скорост до 300  $Mbit/s$
- ❑ **Лазерни излъчватели** - светодиоди, обединени с оптичен резонатор
  - ❖ управляващия ток на резонатора надвишава даден праг и оптичният резонанс предизвиква принудителна фотонна емисия, която увеличава многократно и мощността на изходния сигнал
  - ❖ кохерентна излъчвана светлина (една и съща дължина), която се разпространява под малък ъгъл
  - ❖ използват се за едномодови влакна - честоти от порядъка на  $GHz$

# Оптични излъчватели, приемници и съединители

---

- **Оптични приемници** - за обратното преобразуване на светлината в електрически сигнали
  - ❖ използват се полупроводникови фотодиоди с обратно включен *PN*-преход
  - ❖ два типа фотодиоди: PIN-фотодиоди и лавинни APD фотодиоди
- **Оптични съединители** - осъществяват връзка между еднакви по размери и коефициенти на пречупване оптични влакна
  - ❖ за да няма затихване в точката на свързване челните повърхности трябва да съвпадат точно и да са съосни
  - ❖ загуби поради отстояние между свързаните оптични влакна

## Оптични излъчватели, приемници и съединители

---

- ❖ **неразглобяеми съединения** - *заваряване* или *механично свързване* на влакната чрез залепване с прозрачна епоксидна смола
- ❖ **разглобяеми съединения** - налага се в края регенерационните участъци; връзка по два начина: чрез **лещи** и **челно свързване**

# КОДИРАНЕ И МОДУЛИРАНЕ НА СИГНАЛИТЕ

---

## 1. Линейно кодиране

1.1. Кодиране на цифрова информация в цифров сигнал

1.2. Неполярно кодиране

1.3. Еднополярно кодиране

1.4. Двуполярно кодиране

## 2. Дискретно-аналогова модулиране

## 3. Аналогово-дискретно и аналогово-аналогово преобразуване

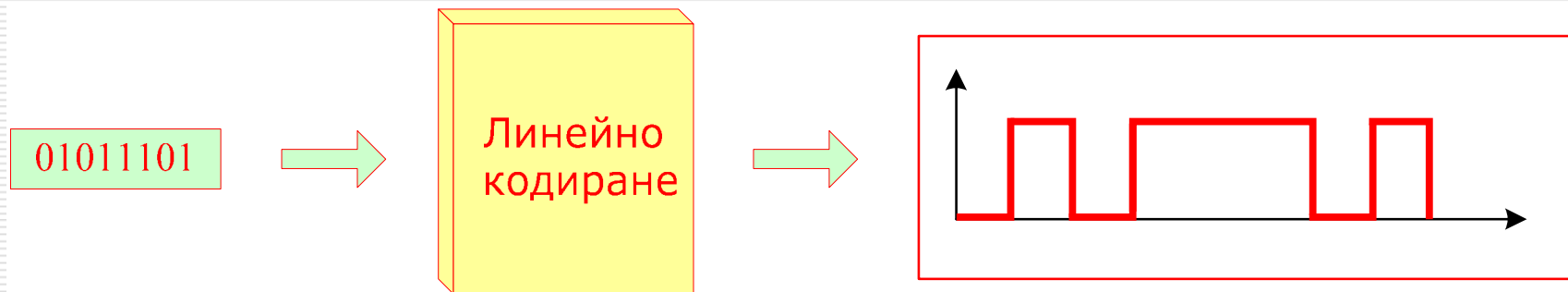
# Кодиране на цифрова информация в цифров сигнал

---

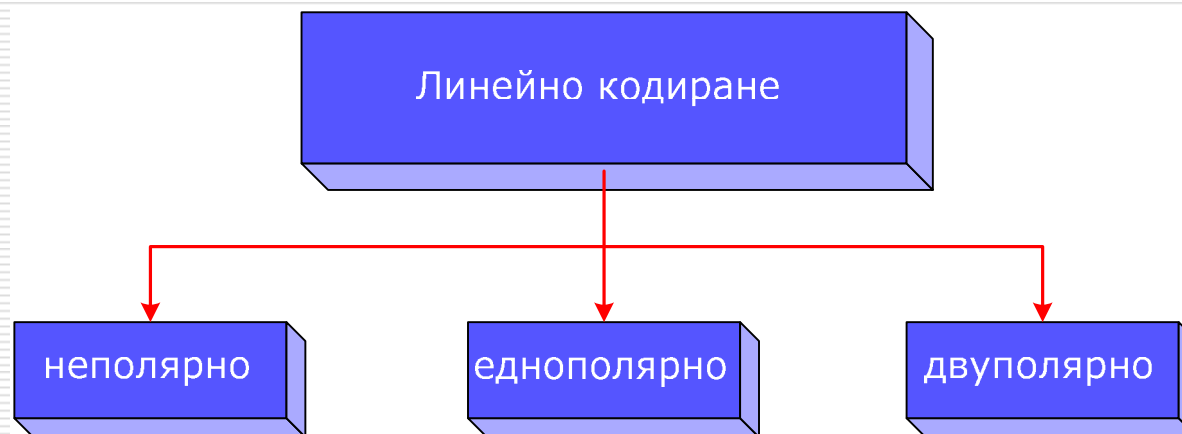
- Кодиране – трансформиране на оригиналната информация във формат, удобен за използване от комуникационната система
  - простият сигнал не носи информация, поради което се *манипулира* така, че лесно да се идентифицира от предавателя и приемника
  - *Пример:* компютър - информацията като **0** и **1** в ASCII код → за предаване е необходимо преобразуване в цифрови/аналогови сигнали
- Код - съвкупност от правила за представяне на данни чрез символи, т.е. условно информационно съответствие между елементите на две множества

# Кодирание на цифрова информация в цифров сигнал

- ❑ Двоичните **1** и **0** генерирани от РС се превръщат в поредица от напреженови импулси

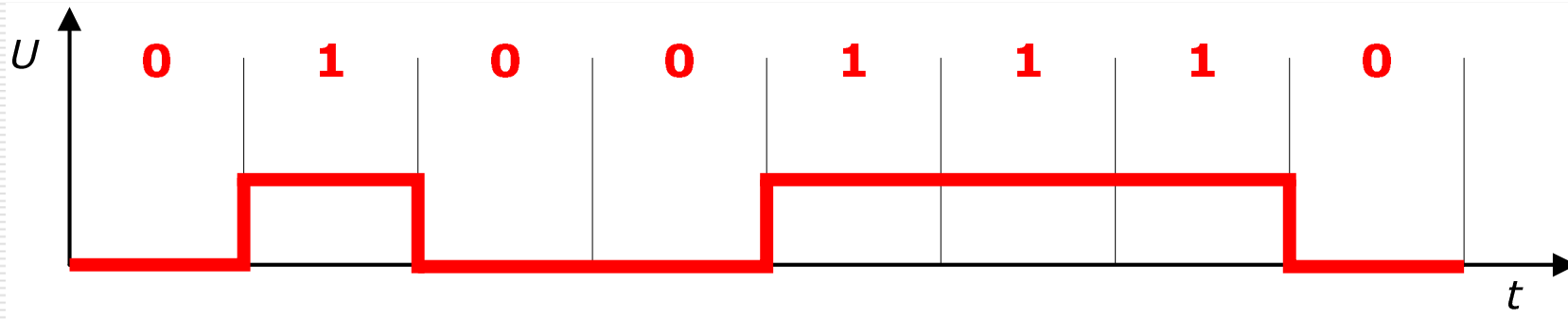


- ❑ Три основни вида



## Неполярно кодиране

- ❑ Просто, но излязло от употреба, поради недостатъци



- ❑ В кодиращите устройства напреженото ниво отговаря на **1**, а другото на **0**
- ❑ недостатъци – постояннотокова компонента и синхронизация
  - *постояннотокова компонента* - средната стойност на амплитудата на неполярния кодиран сигнал е различна от нула (компонента с нулева честота)
  - *синхронизация* - приемника не може да различи началото и края на всеки бит

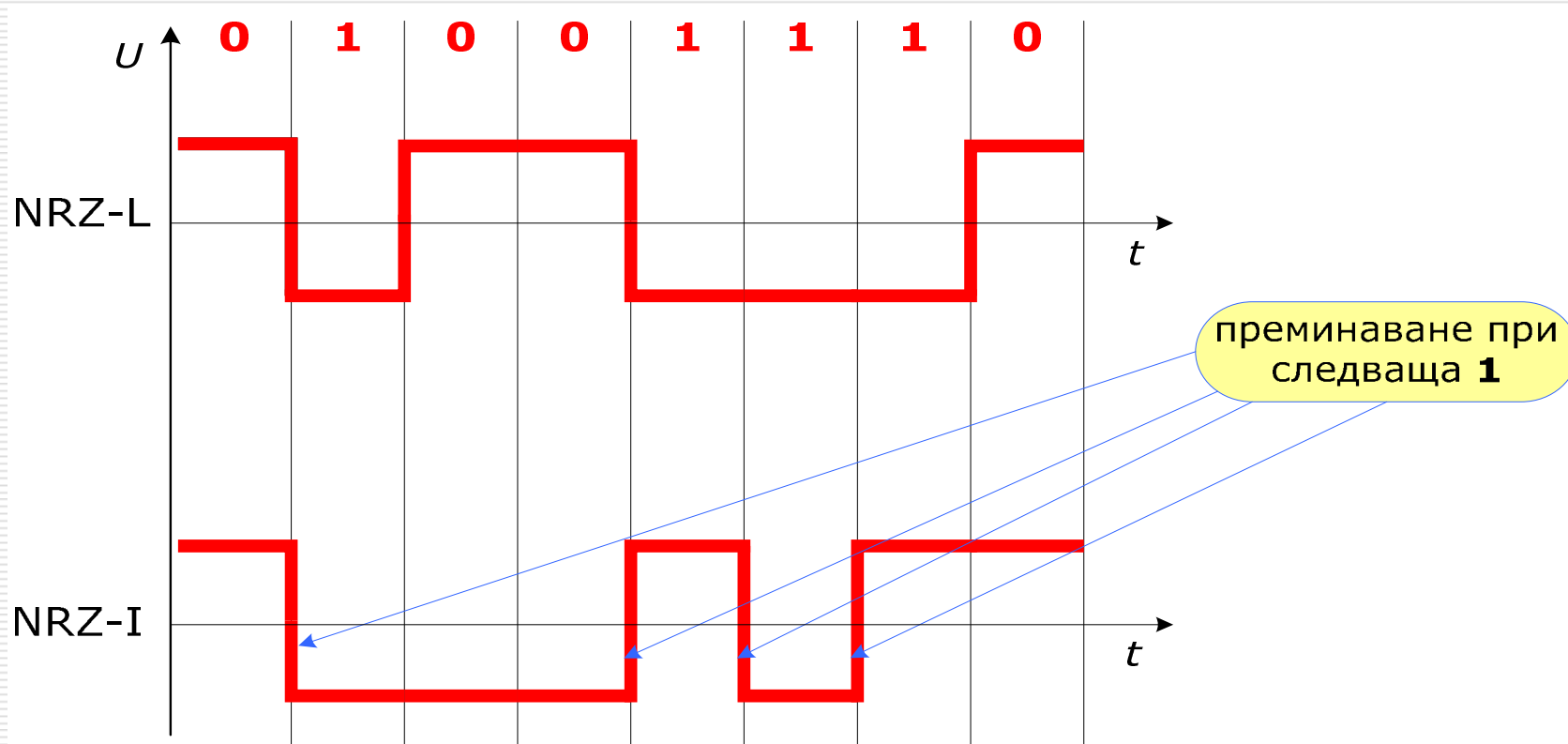
# Еднополярно кодиране

- ❑ Използва две нива на напрежение положително и отрицателно, чрез което се премахва постояннотоковата компонента
- ❑ Видове – *еднофазно* и *двуфазно*
- ❑ Двуфазно кодиране (*Manchester* и *Differential Manchester*) - решение на проблема със синхронизацията; промяна в сигнала в средата на всеки бит



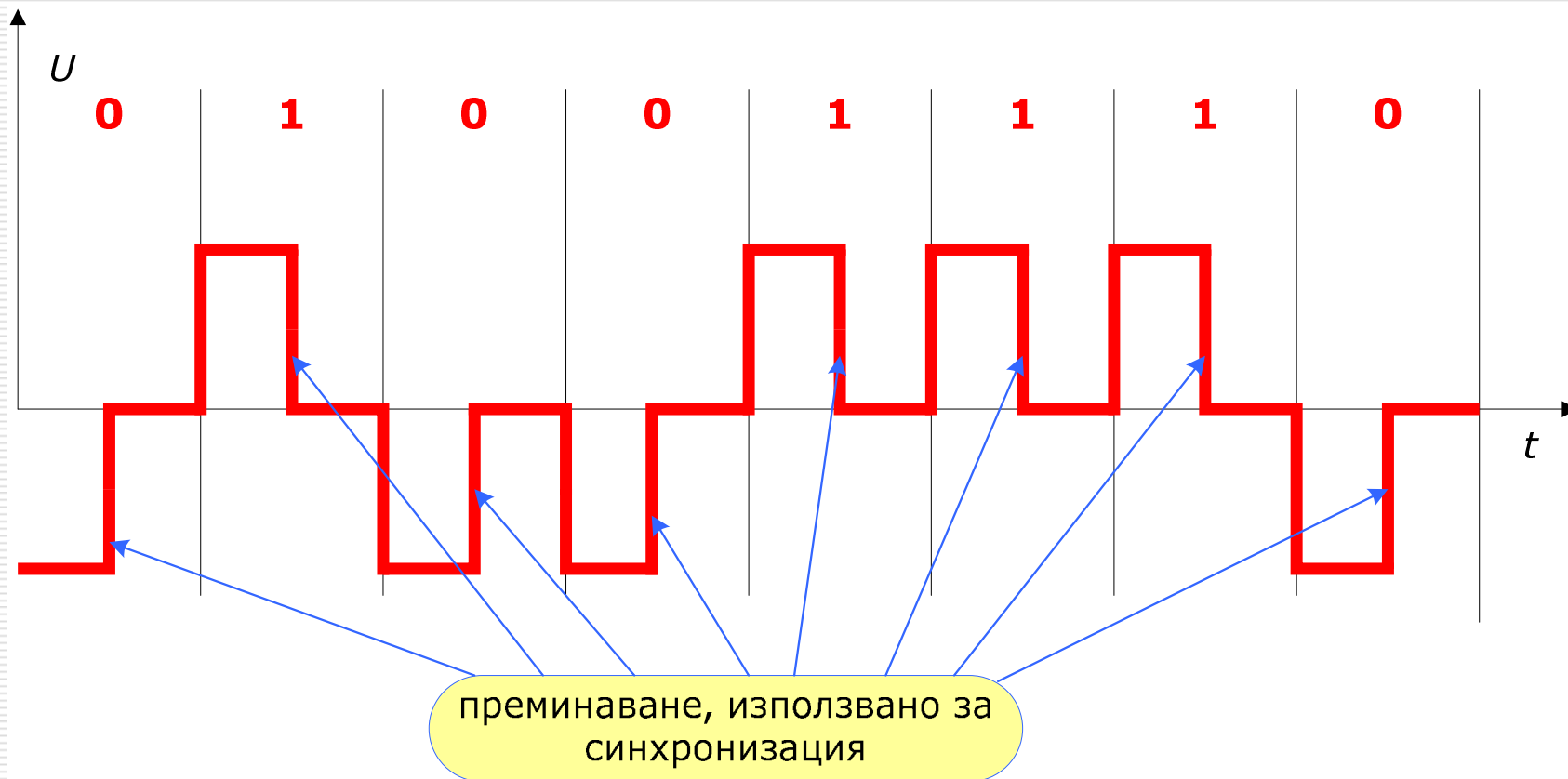
# Еднополярно кодиране

- NRZ (*Non Return to Zero*) **без връщане към нула**



# Еднополярно кодиране

- RZ (*Return to Zero*) **с връщане към нула**



## Еднополярно кодиране

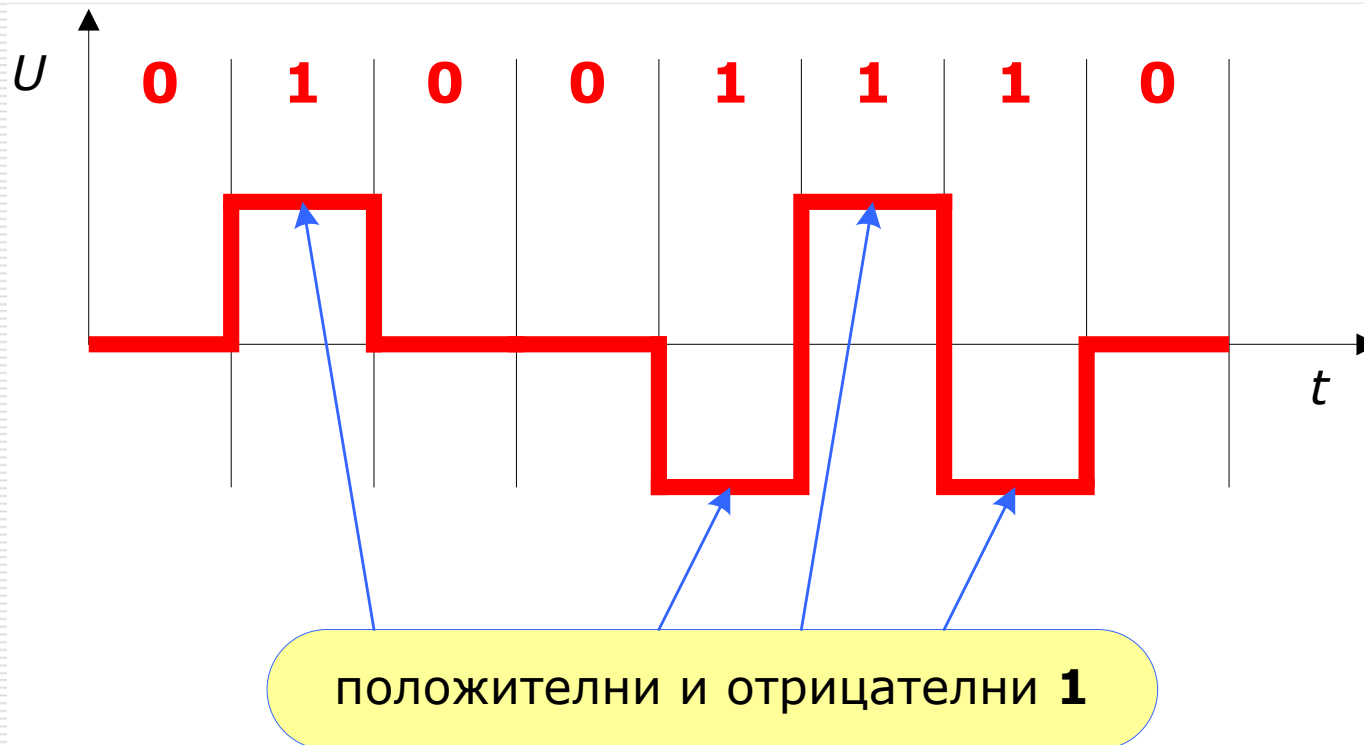
---

- NRZ-L код - нивото на сигнала зависи от типа на битовете, които представя: "+" напрежение – **0**, а "-" напрежение – **1**; възниква проблем при продължителен поток от 0 или 1
- NRZ-I код - инверсията на нивото на напрежение се представя с един бит; позволява на приемника да ресинхронизира тактовия генератор (таймер)
- RZ - три стойности: положителна, отрицателна и нула
  - ❖ **1** - преход от положително към нулево напрежение
  - ❖ **0** - преход от отрицателно към нулево напрежение

# Двуполярно кодиране

## □ AMI код (*Alternate Mark Inversion*)

- ❖ алтернативна инверсия на единицата
- ❖ **0** - нулево напрежение
- ❖ **1**-алтернативно редуване на  $\uparrow$  и  $\downarrow$  напрежения



## Двуполярно кодиране

---

- ❖ *недостатък* – не може да синхронизира дълга поредица от нули
  - ✓ премахва се чрез **биполярна единица с висока плътност** HDB-3 (*High Density Bipolar*)
  - ✓ вкарва промените в еталонния биполярен AMI сигнал всеки път, когато 4 последователни 0 са кодирани в информационния поток

# КОДИРАНЕ И МОДУЛИРАНЕ НА СИГНАЛИТЕ

---

## 1. Линейно кодиране

## 2. Дискретно-аналогова модулиране

2.1. Амплитудна манипулация

2.2. Честотна манипулация

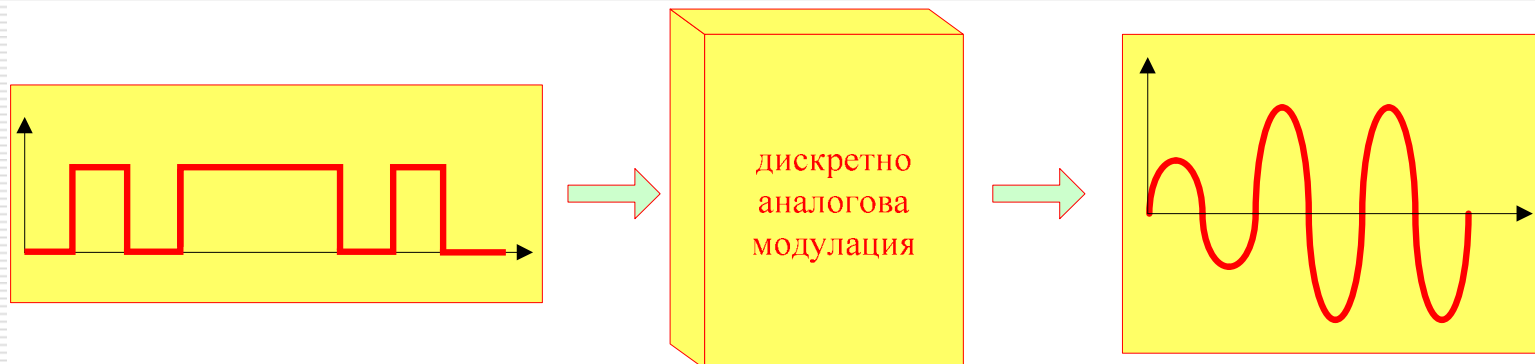
2.3. Фазова манипулация

2.4. Квадратично-амплитудна модулация

## 3. Аналогово-дискретно и аналогово-аналогово преобразуване

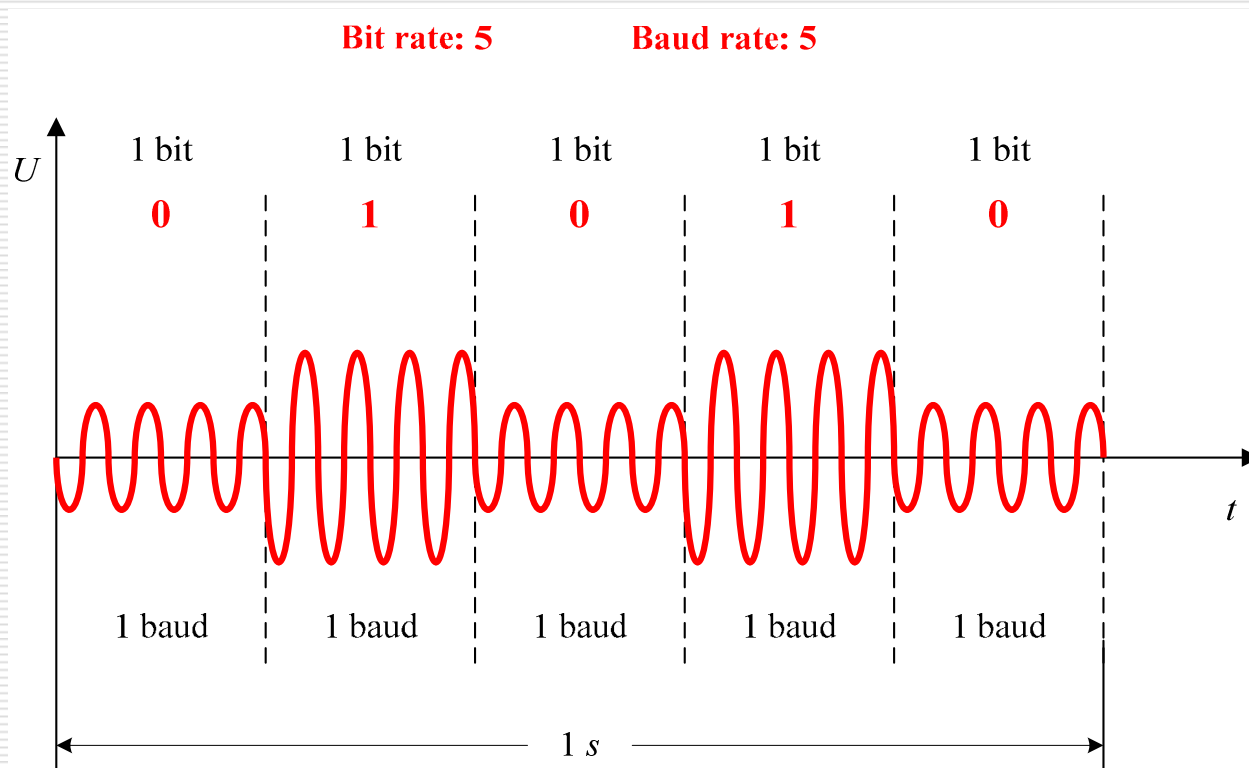
# Дискретно-аналогова модулация

- ❑ Изменение на дадена характеристика на аналоговия сигнал, базирана на информация от цифровия сигнал
- ❑ Три характеристики на аналоговия сигнал - амплитуда, честота, фаза
  - амплитудна манипулация ASK (*Amplitude Shift Keying*)
  - честотна манипулация FSK (*Frequency Shift Keying*)
  - фазова манипулация PSK (*Phase Shift Keying*)
  - квадратично-амплитудна модулация QAM (*Quadrature Amplitude Modulation*)



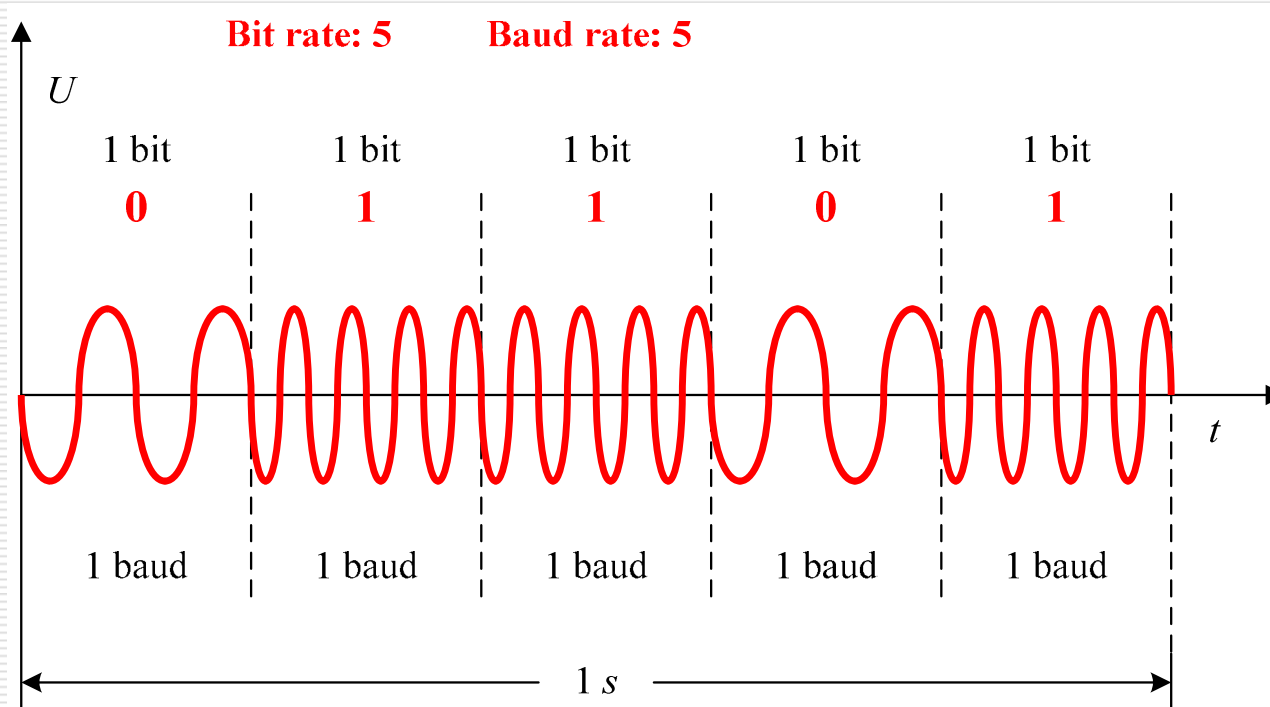
# Амплитудна манипулация ASK

- ❑ ASK → фаза и честота без промени; промяна на амплитуда
  - амплитудата - постоянна за всеки бит ~ от неговата стойност (**0/1**)
  - недостатък - ограничена скорост на предаване; без шумоустойчивост



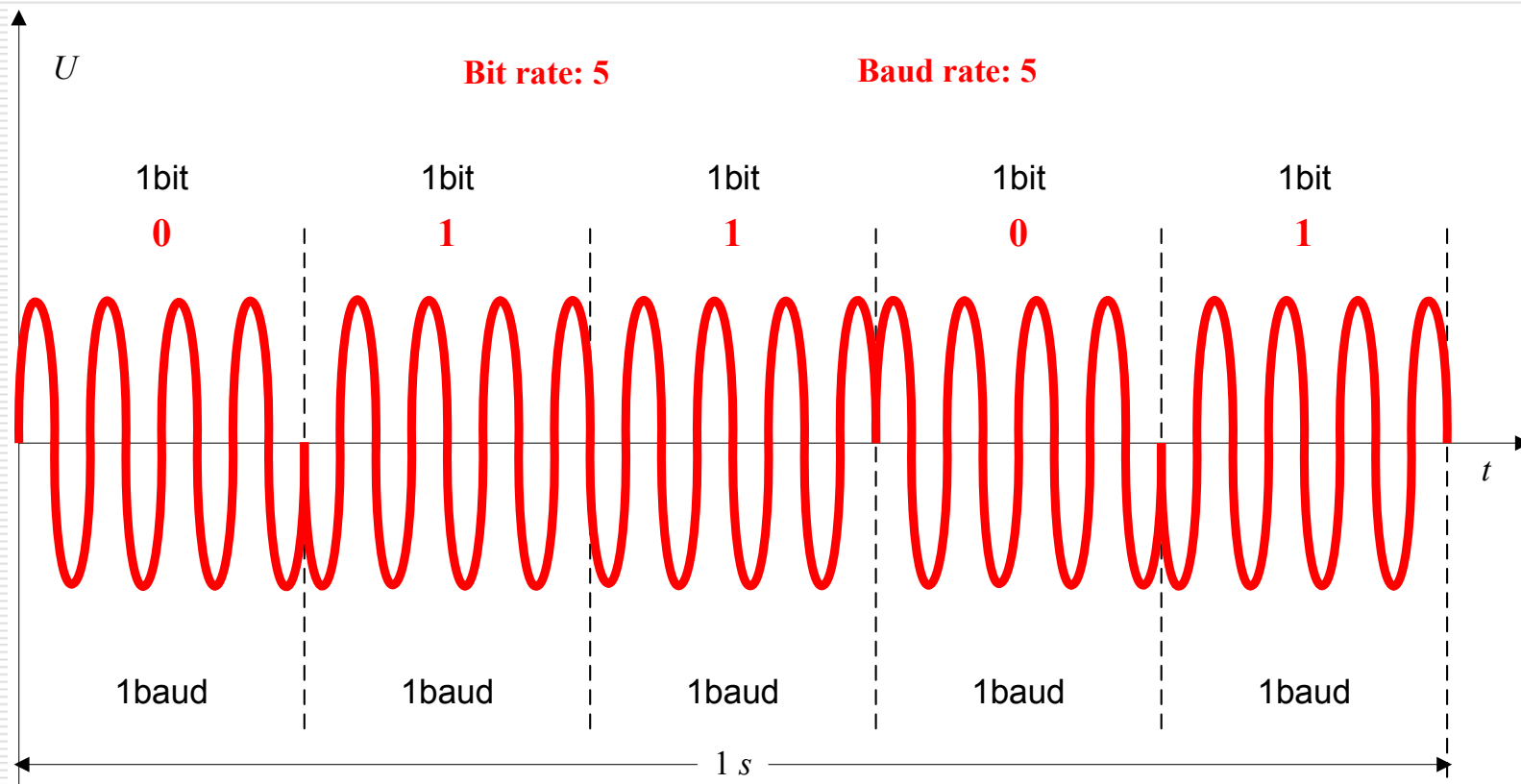
# Честотна манипулация FSK

- ❑ FSK → честотата  $\sim$  от стойността на бита (**0/1**); амплитудата и фазата - постоянни
- ❑ Приемникът следи за честотни промени през даден период
- ❑ Ограничаващ фактор - физическите възможности на носещия сигнал



# Фазова манипулация PSK

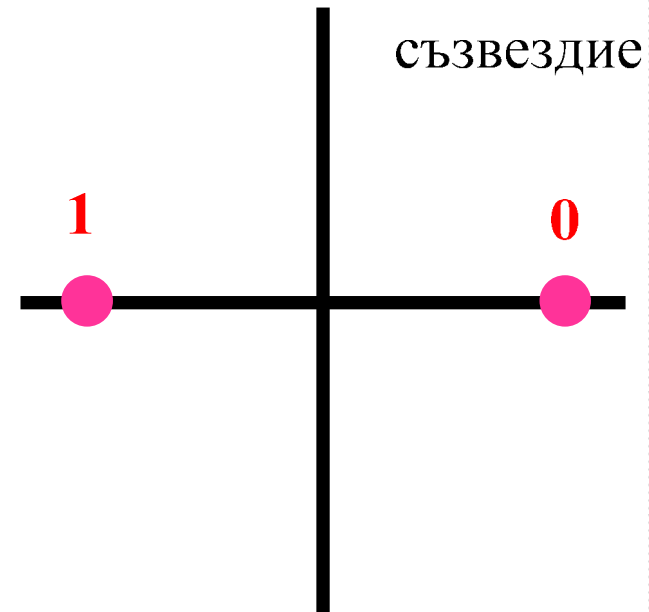
- PSK → фазата на носещата честота  $\sim$  от стойността на бита (**0/1**); амплитудата и честота - постоянни



## Фазова манипулация PSK

- ❑ Две различни фази 2-PSK - двукратна PSK манипулация; диаграма на фазовото състояние - отношението фаза/битова стойност

бит	фаза
0	0
1	180



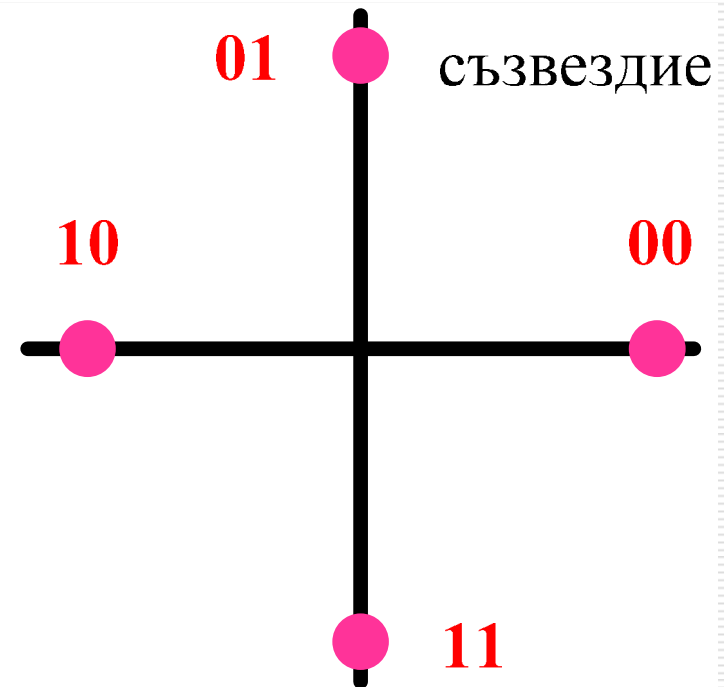
- ❑ За по-малки изменения в сигнала се използват *четири фазови изменения*, които се представят с два бита

# Фазова манипулация

## □ 4-PSK (QPSK – квадратурна фазова)

- съзвездие на сигнала  $0^\circ=00$ ;  $90^\circ=01$ ;  $180^\circ=10$ ;  $270^\circ=11$
- двойката битове - двубит
- предава два пъти по-бързо данни от 2-PSK

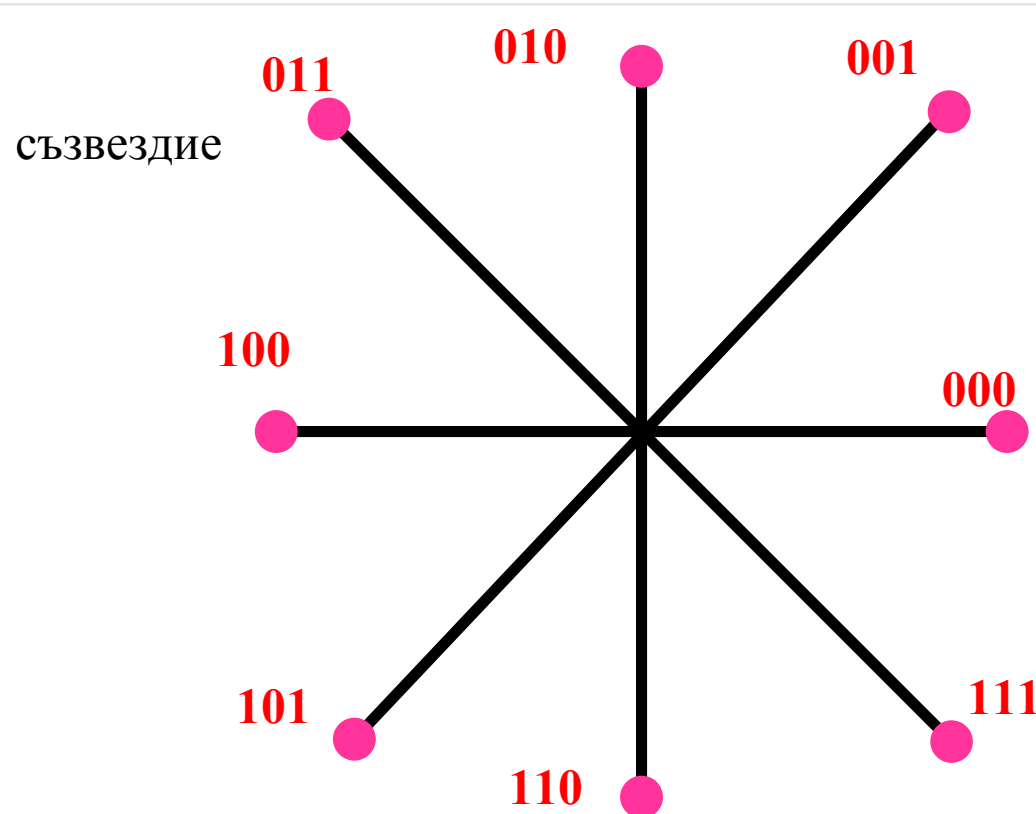
двубит	фаза
00	0
01	90
10	180
11	270



# Фазова манипулация

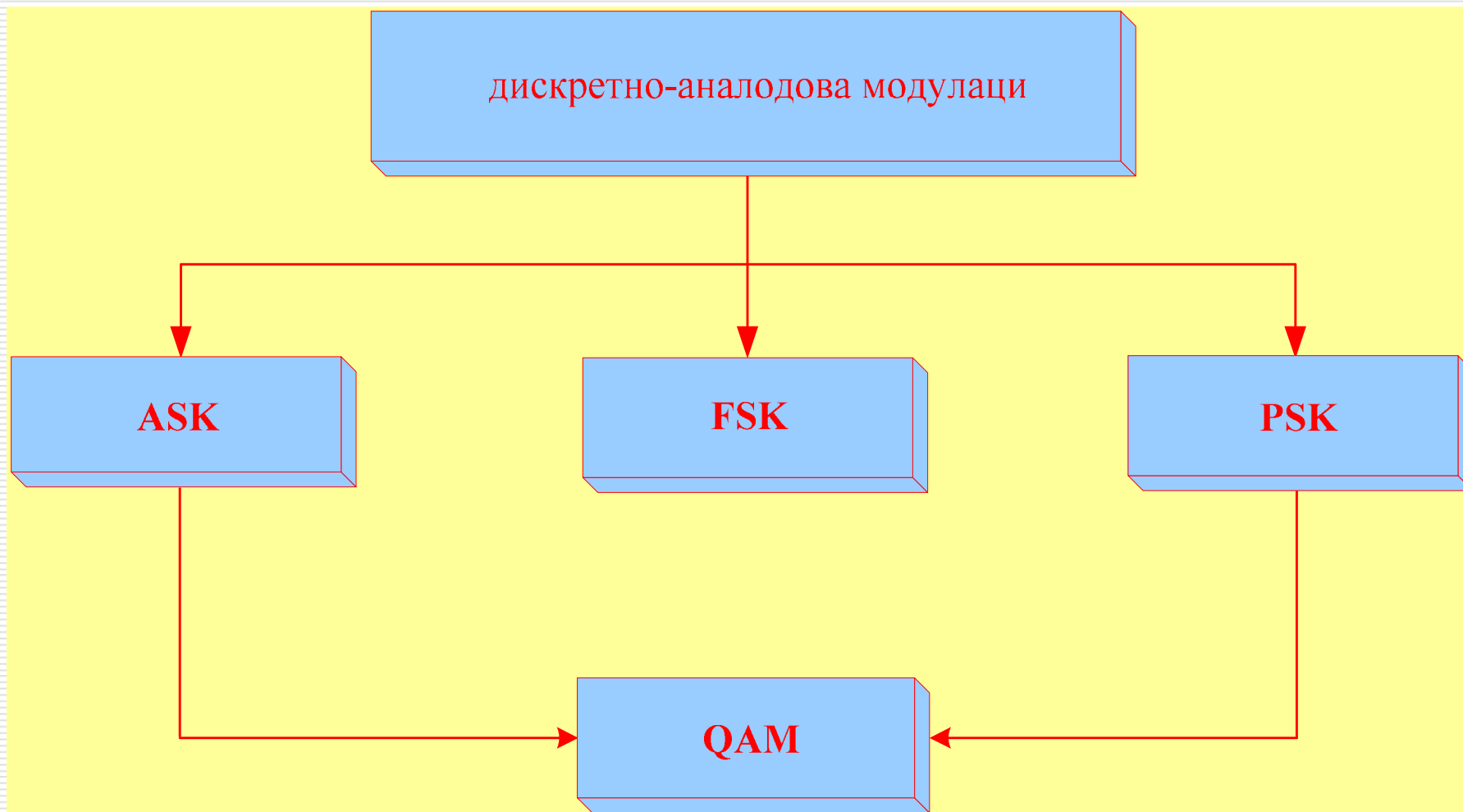
- ❑ 8-PSK осемкратна манипулация - сигналът през  $45^\circ$ 
  - представя се чрез три бита (трибютарен код)
  - осем възможни фази ( $2^3=8$ ) → три пъти по-бърза

трибит	фаза
000	0
001	45
010	90
011	135
100	180
101	225
110	270
111	315



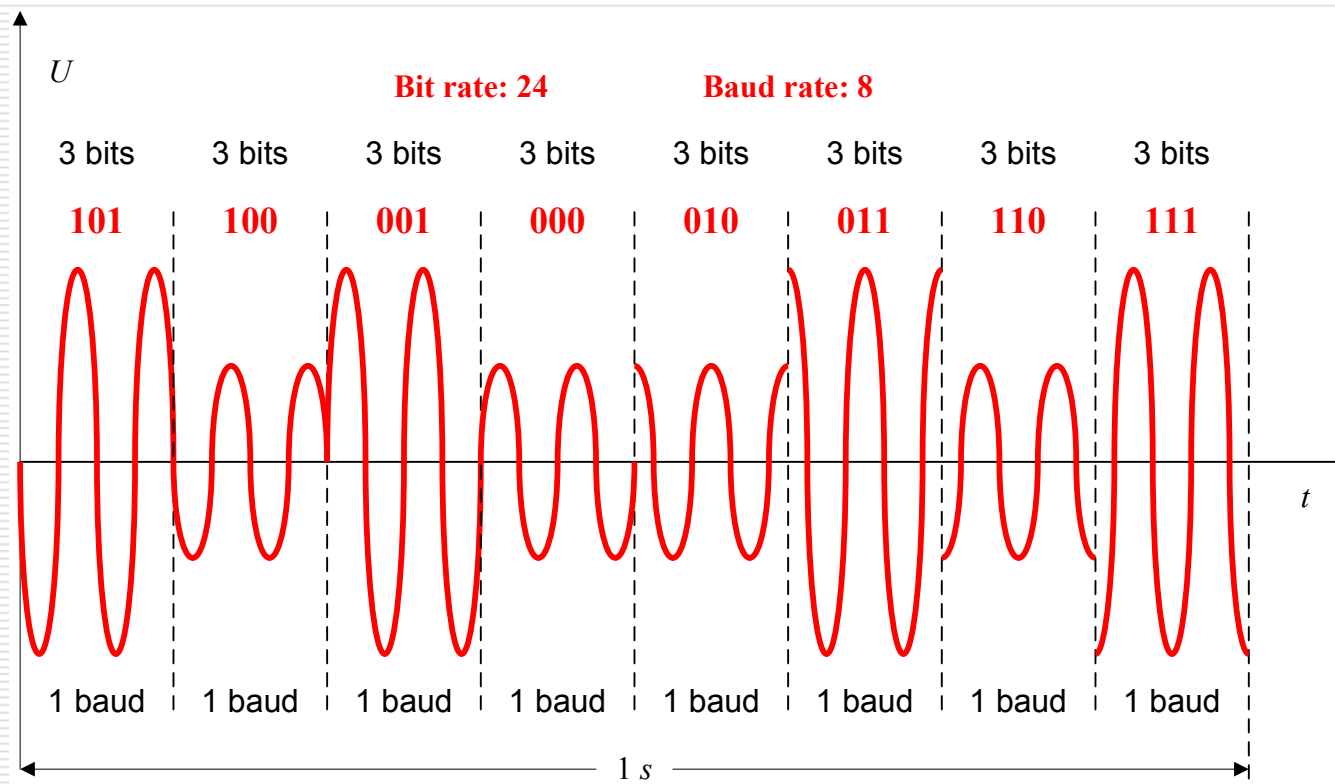
# Квадратично-амплитудна модулация QAM

- Комбинация от ASK и PSK



# Квадратично-амплитудна модулация QAM

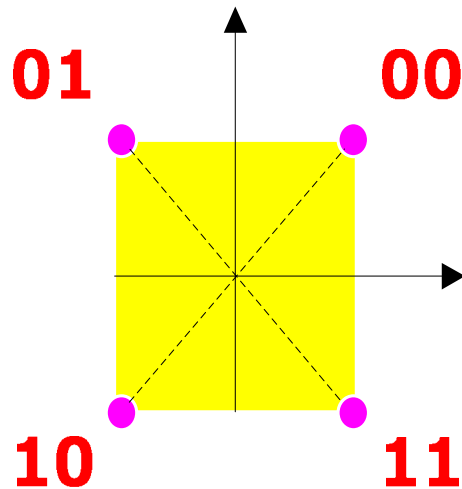
- ❑ Преобразува първичният - във ВЧ електромагнитен сигнал
- ❑ Сигналят се предава по аналогови или цифрови канали на максимално разстояние с минимална загуба
- ❑ Процес на смесване на сигнали с различен параметър - чрез променливи генератори на  $\sin$  сигнали



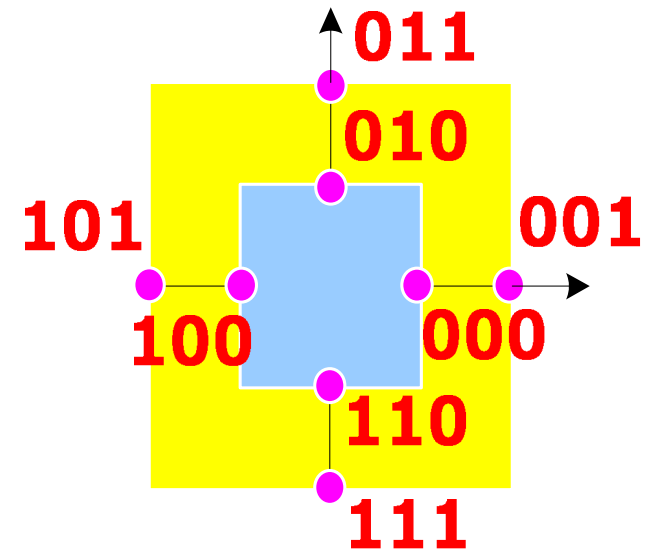
# Квадратично-амплитудна модулация

- QAM → чрез комбинация от ASK и PSK се постига максимално разграничаване между битови-, двубитови-, трибитови - сигнални единици

**4-QAM** 1 амплитуда, 4 фази



**8-QAM** 2 амплитуди, 4 фази

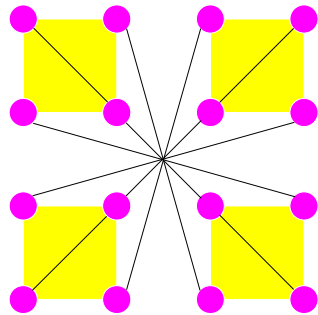


# Квадратично-амплитудна модулация

## □ Популярни 16-QAM конфигурации

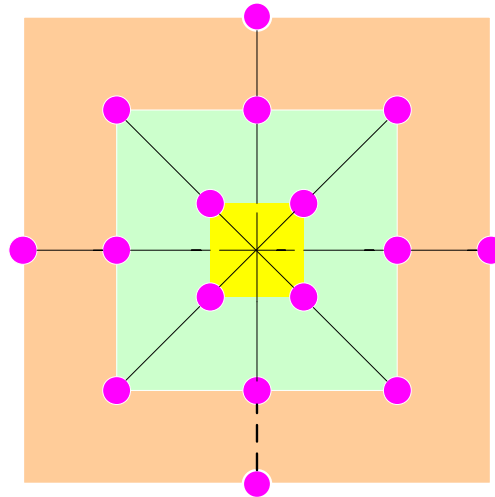
- 3 амплитуди и 12 фази (препоръка на ITU-T – шумоустойчив с голямо съотношение фаза/амплитуда)
- 4 амплитуди и 8 фази (OSI стандарт)

3 амплитуди, 12 фази



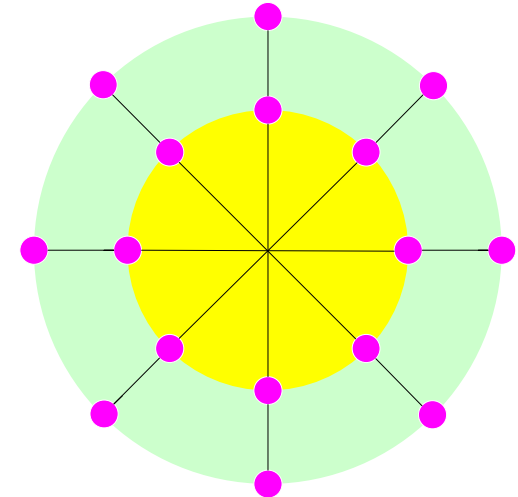
16-QAM

4 амплитуди, 8 фази



16-QAM

2 амплитуди, 8 фази



16-QAM

# Квадратично-амплитудна модулация

- ❑ Сравнение между битова и бодова скорост (*bit/ baud*)



# КОДИРАНЕ И МОДУЛИРАНЕ НА СИГНАЛИТЕ ПРИ ПРЕДАВАНЕ НА ДАННИ

---

**1. Линейно кодиране**

**2. Дискретно-аналогова модулиране**

**3. Аналогово-дискретно и аналогово-аналогово преобразуване**

3.1. Импулсна амплитудна модулация

3.2. Импулсна кодова модулация

3.3. Амплитудна модулация

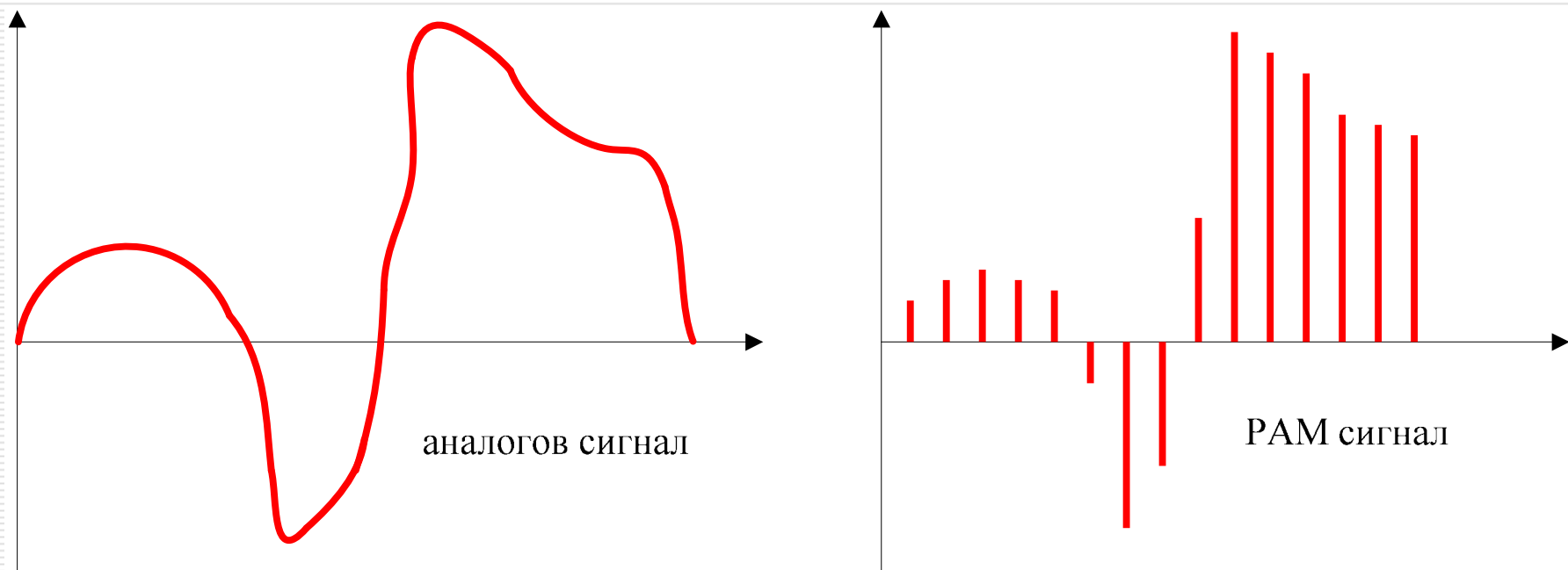
3.4. Честотна модулация

3.5. Фазова модулация

# Импулсна амплитудна модулация РАМ

## □ РАМ (*Pulse Amplitude Modulation*)

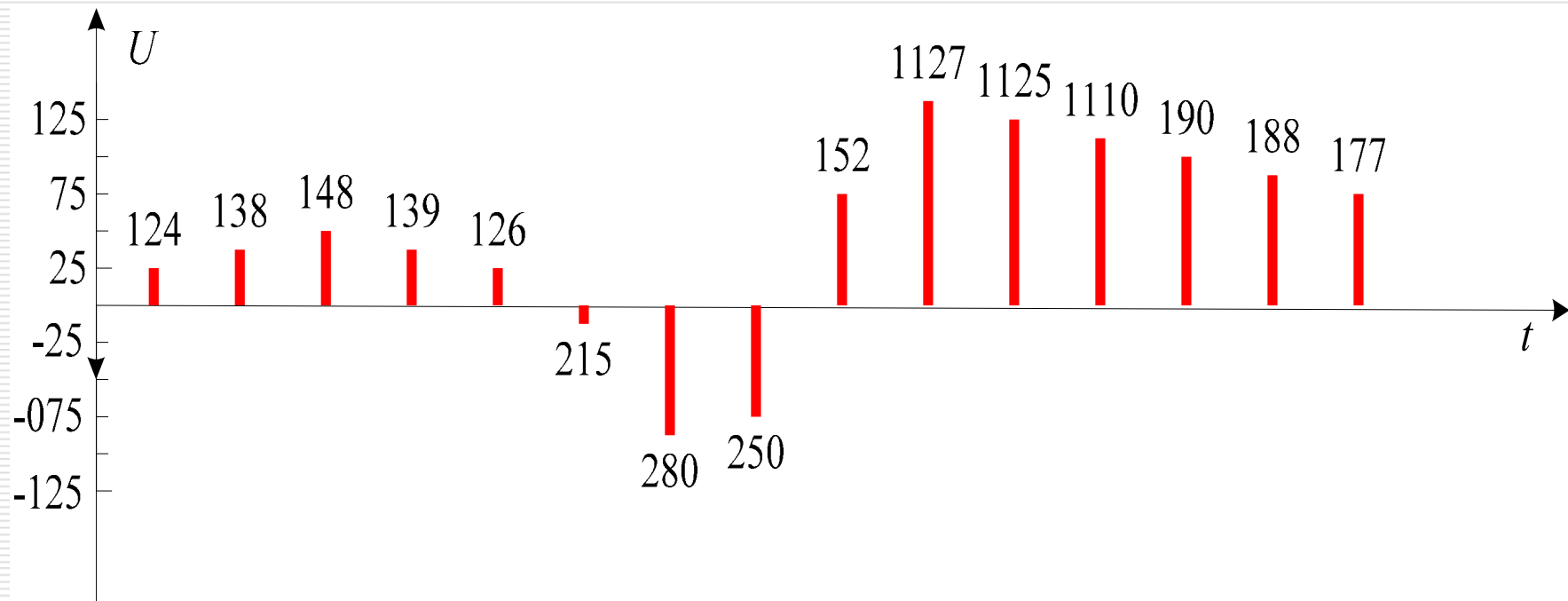
- преобразува сигнала от безкраен в дискретен брой стойности без загуба на качество
- дискретизира през равни интервали от време
- генерира серия от импулси



# Импулсна кодова модулация РСМ

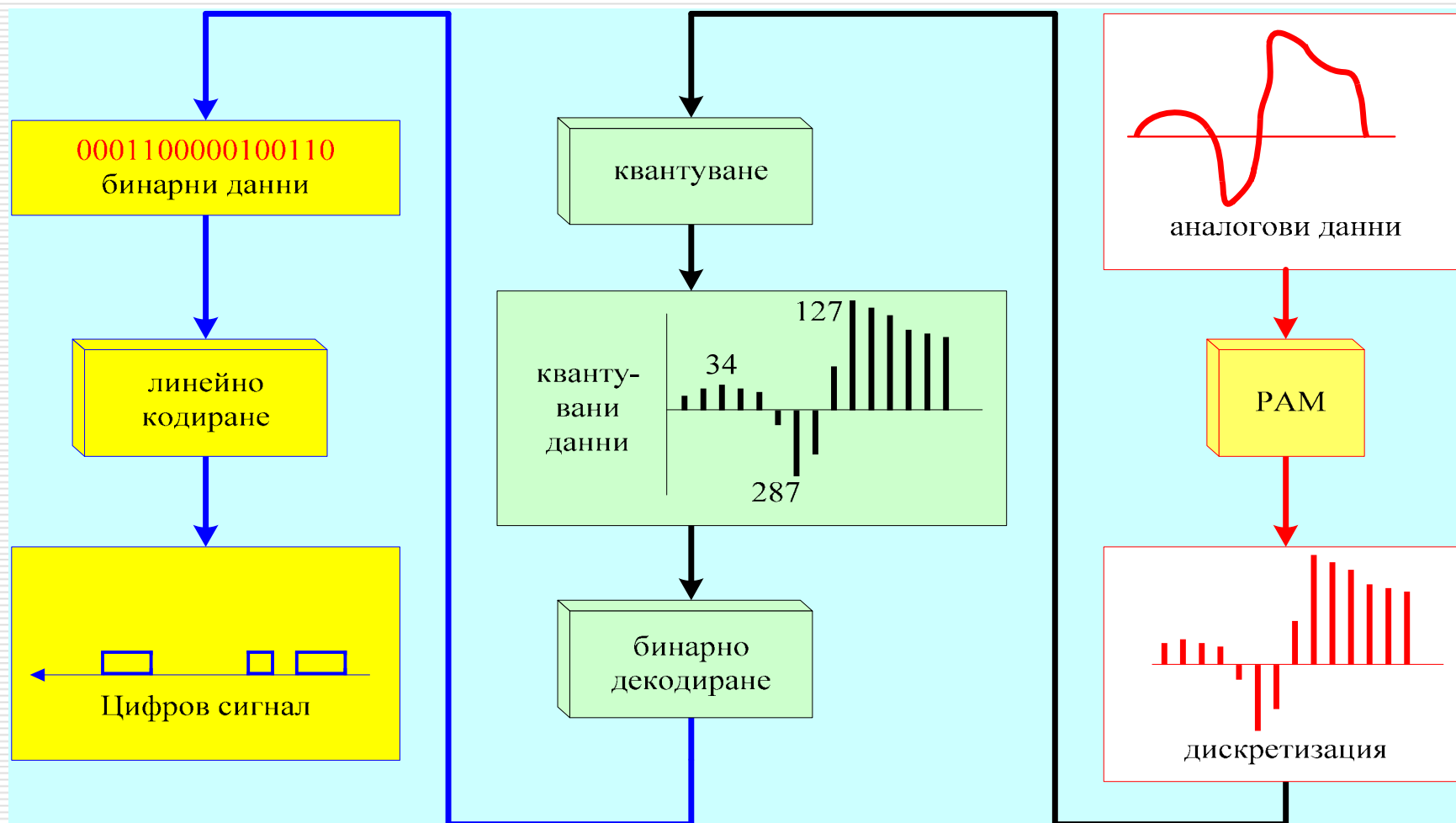
## □ РСМ (*Pulse Code Modulation*)

- преобразува импулсите в пълен цифров сигнал
- квантува импулсите, като присъединява пълната стойност на сигнала в обхвата на дискретизиране



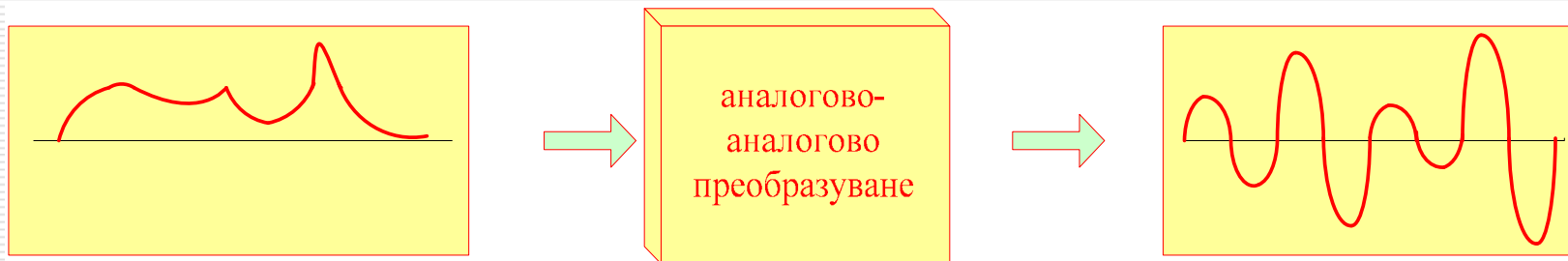
# Импулсна кодова модуляция РСМ

- ❑ Включва 4 процеса: РАМ; квантуване; двоично кодиране и линейно кодиране

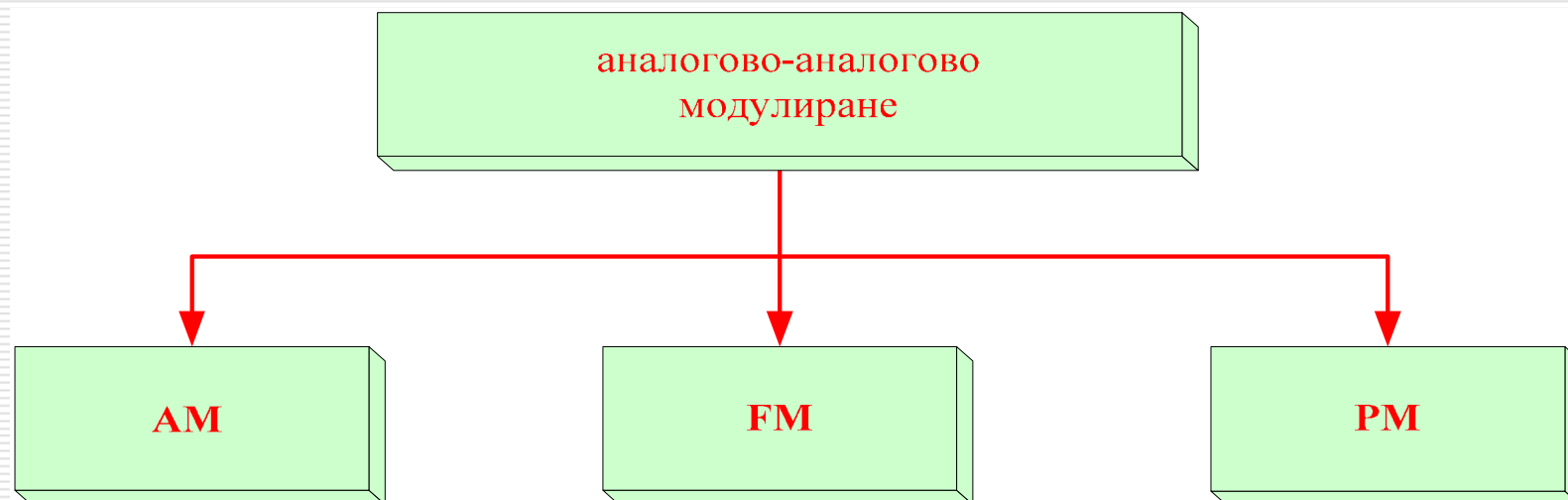


# Аналогово-аналогово преобразуване

- Изобразяване на аналоговата информация чрез аналогов сигнал



- Реализация: **амплитудна** АМ, **честотна** FM и **фазова модулация** РМ



## Амплитудна модуляция АМ

- ❑ Амплитудата на носещия сигнал се променя чрез амплитудата на модулиращия сигнал; честотата и фазата - остават непроменени
- ❑ Модулиращият сигнал се явява обвивка на носещия



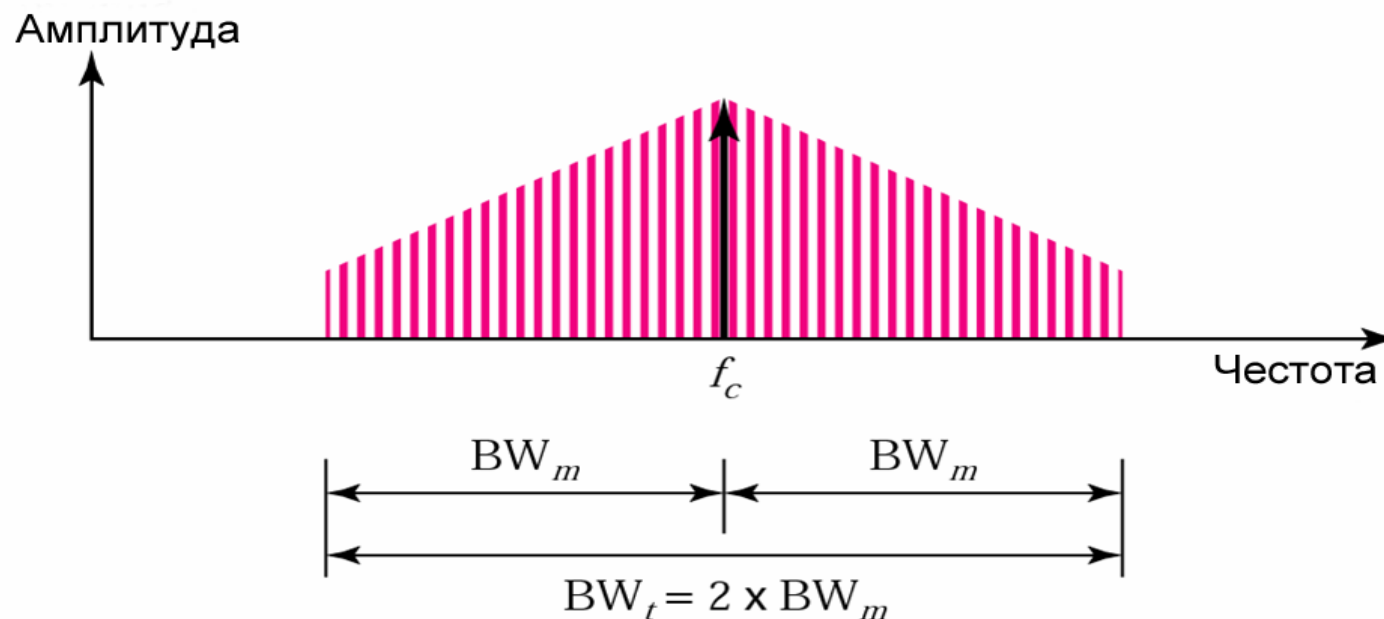
# Амплитудна модулация

- ❑ Честотна лента на АМ сигнал - двойна честотна лента на модулиращия сигнал около носещата честота; заштрихована част от графиката - честотен спектър

$BW_m$  = честотна лента на модулиращия сигнал (аудио)

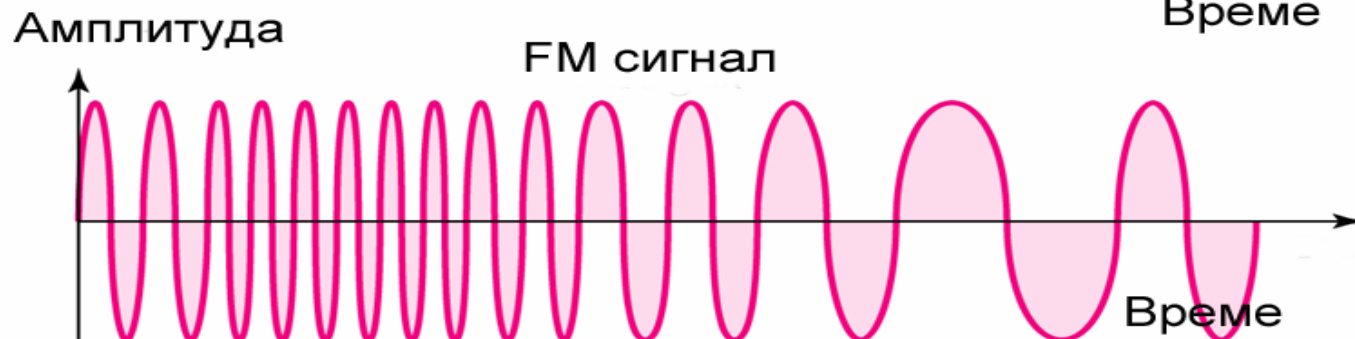
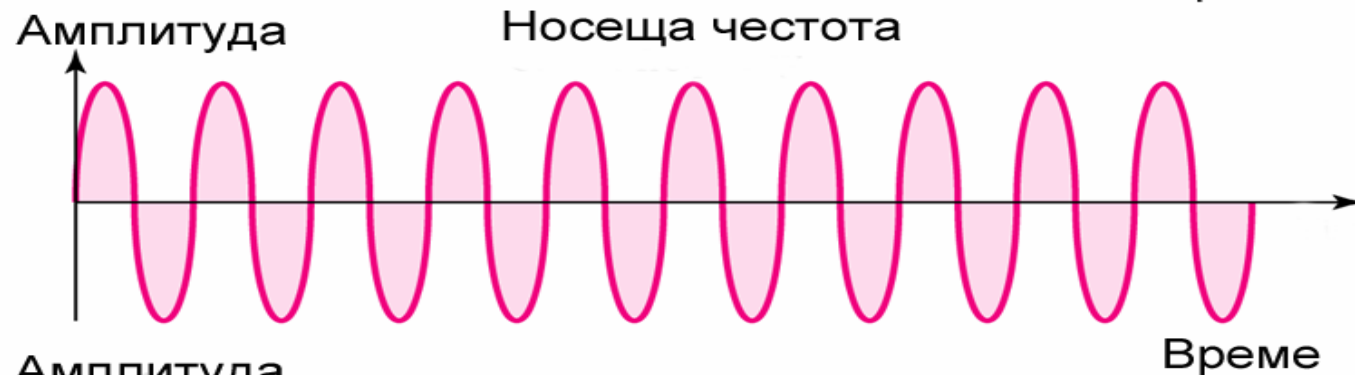
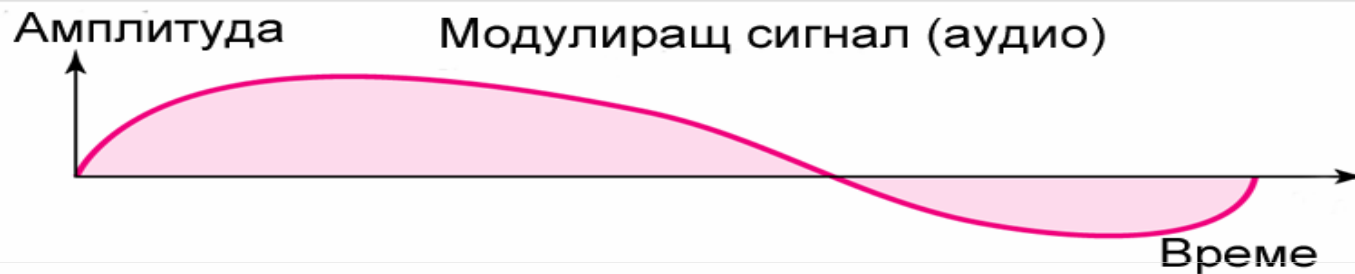
$BW_t$  = обща честотна лента (радио)

$f_c$  = носеща честота



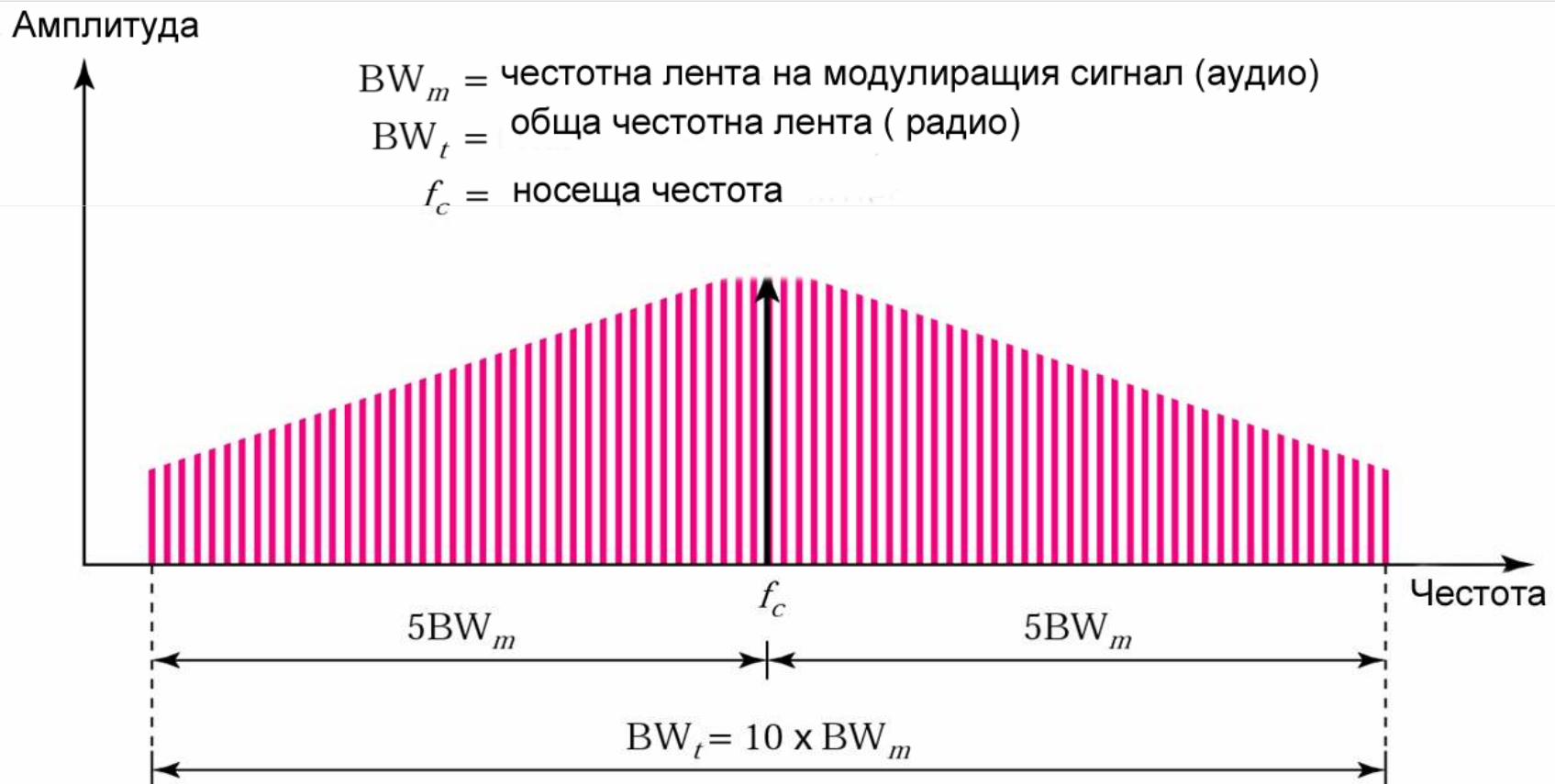
# Честотна модулация FM

- ❑ Честотата на носещия сигнал следва амплитудата на модулиращия сигнал
- ❑ Амплитуда и фаза на носещия сигнал - постоянни



# Честотна модулация

- ❑ Честотна лента на FM сигнал - 10 пъти честотната лента на модулиращия сигнал около носещата честота
- ❑ Общата честотна лента се определя от лентата на аудио сигнала



## Фазова модулация РМ

---

- Алтернатива на FM, поради по-прости хардуерни изисквания
  - Фазата на носещия сигнал следва амплитудата на модулиращия сигнал
  - Амплитуда и честота на носещия сигнал - постоянни
  - Анализът на модулирания сигнал е сходен с анализа на FM, т.е. фазовата модулация е същата като честотната модулация, но се променя на  $180^\circ$

# ГРЕШКИ ПРИ ПРЕДАВАНЕТО НА ЦИФРОВИ ДАННИ

---

## 1. Грешки при последователно предаване

1.1. Видове грешки

1.2. Асинхронно предаване

1.3. Синхронно предаване

## 2. Откриване на грешки чрез проверка за еднаквост

## 3. Циклична проверка за информационен излишък

## 4. Корекция на грешките

# Грешки при последователно предаване

---

- ❑ Цифрово предаване на данни – предимно *асинхронно* и *синхронно последователно* предаване
  - ❖ данните от източника към адресата могат да се сгрешат при предаване
  - ❖ по-често само част от съобщението се променя при предаване
- ❑ При последователно предаване **сигналните елементи** се изпращат по *линията* единично и биват:
  - ❖  $<1 \text{ bit}$  - битът се променя между предаването и приемането ( Манчестър код)
  - ❖  $1 \text{ bit}$  – предмет на разглеждане (NRZ-L и FSK)
  - ❖  $>1 \text{ bit}$  – квадратурна фазова манипулация 4-PSK

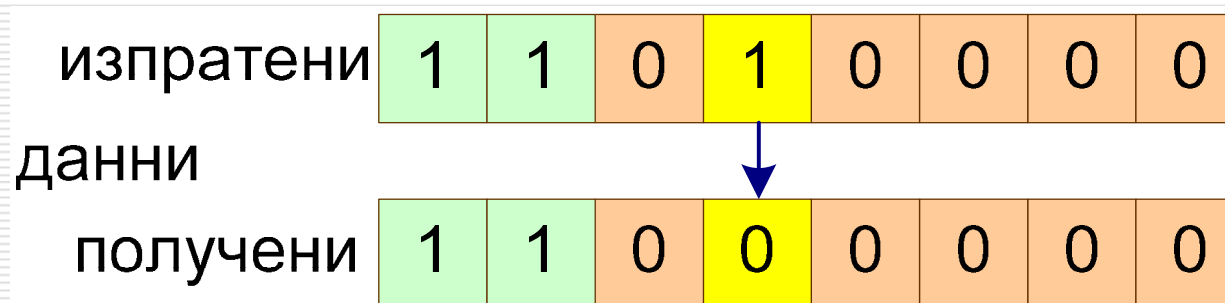
## Видове грешки

---

- ❑ Сигналят представлява *кодирани данни в двоичен код* → грешките променят смисъла на данните
  - ❖ битът се променя между предаването и приемането
  - ❖ пример: *предадена 1 - приета 0* или *предадена 0 - приета 1*
- ❑ Надеждна връзка - **откриване** и **коригиране** на грешки
  - ❖ осъществява се в канален или транспортен OSI слой
- ❑ Два вида грешки: **битова** и **динамична грешка**

## Видове грешки

- ❑ **Битова грешка** - променя **0** на **1** или **1** на **0**



- *пример:* импулсно смущение от 0,01s при предаване със скорост 1200bit/s променя до 12 бита
- малка вероятност за грешка при последователно предаване

*пример:* източник със скорост с 1Mbps → продължителност на смущението 1/1000000s = 1ms → вероятност за битова  $10^{-6}$

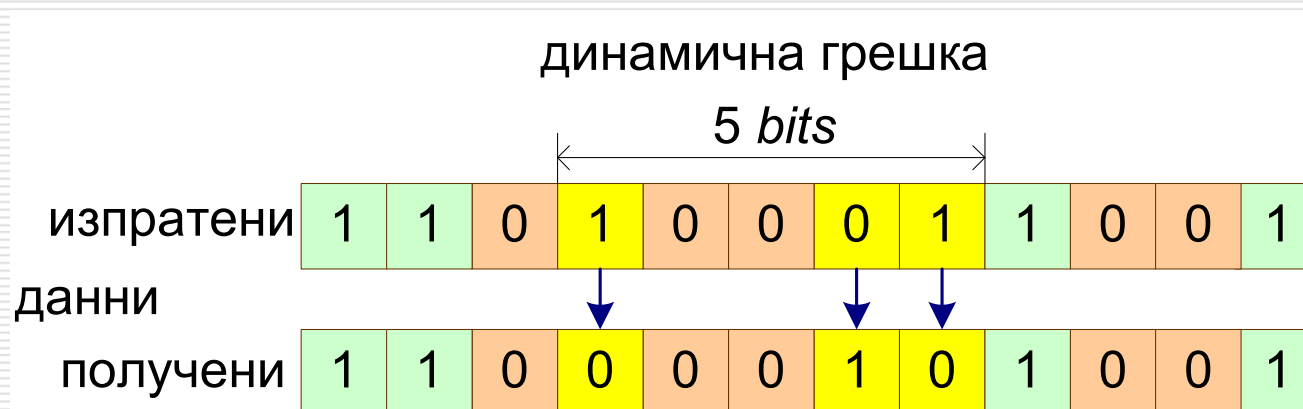
## Видове грешки

- ❑ **Динамична грешка** – промяна в два или повече бита в предаваните данни

➤ Препоръка Q.9 на ITU-T

Група от битове, в която два последователни грешни бита винаги са разделени с по-малко  $x$  верни бита

- дължината се измерва от първия до последния сгрешен бит



## Видове грешки

---

➤ характерна при последователно предаване → шума има по-дълго времетраене от един бит и въздейства на поредица от битове

➤ ефектът от грешката е по-голям при *високи скорости* на предаване

пример: предаване на данни с *1Kbps*, шум *1/100s* → *динамична грешка 10bits*; предаване на данни с *1 Mbps* → *динамична грешка 10 000 bits*

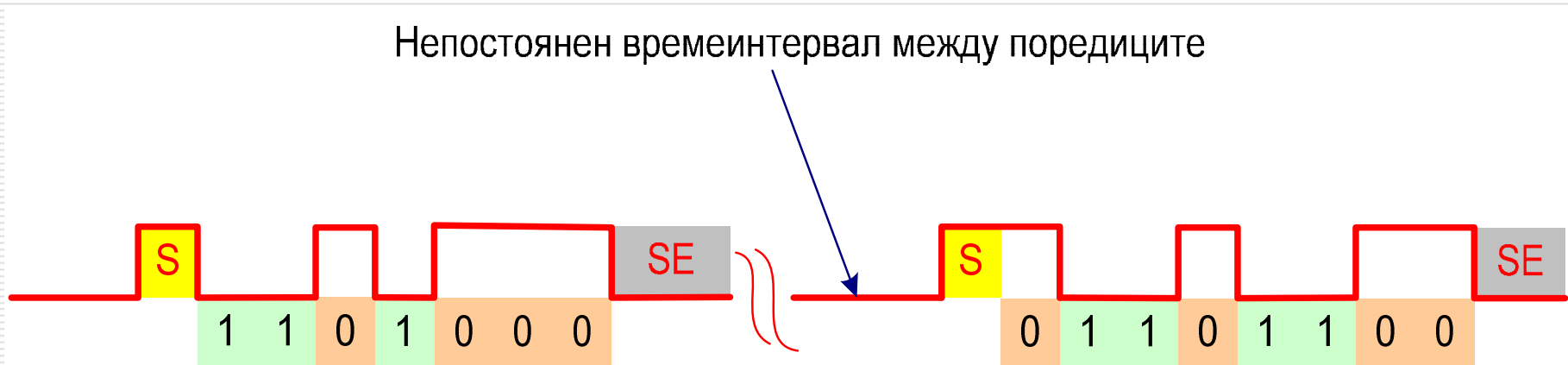
# Асинхронно предаване

- ❑ *Основна задача* - да се избегнат проблемите при предаване чрез *къса и непрекъсната* поредица от битове
- ❑ *Формат на данните*
  - ❖ данните се подреждат по време в блок от 5 до 8 бита
  - ❖ синхронизиране на блока - *само при получателя*
  - ❖ *празно състояние* двоична **1** – липса на предаване



## Асинхронно предаване

- битовете на блока се предават до P бит, който заема мястото на най-висшия бит (**0**)
  - начало на блока - *старт бит S* (**0**)
  - *стоп елемент SE* (**1**) - минималната дължина 1; 1,5 или 2 бита
- Непрекъснат асинхронен 8 bits поток
- без грешка при разлика в скоростите между предавател /приемника 5 % → 45 % разместване в осмия бит



## Асинхронно предаване

- **Пример:** ефект от грешката при приемане
  - скорост на приемаме  $10\text{kbps}$  → всеки бит с времетраене  $100\mu\text{s}$
  - приема се, че получателят е по-бърз с 6 % за всеки бит, т.е. приема битове на всеки  $94\mu\text{s}$



## Асинхронно предаване

---

➤ **грешка** като следствие от

- неправилно получен последен бит

- Изравняване на броенето на битове, т.е. ако 7-ят бит - **1**, а 8-ят - **0**, то се бърка като *S бит- рамкова грешка*

□ Необходимост от *горна граница* в блока при асинхронното предаване - 2-3 бита

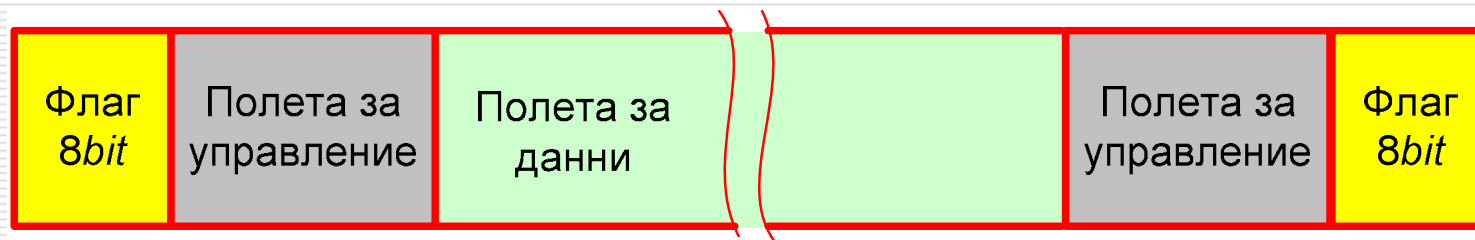
- при SE 1 *bit* за 8 bits дума - 2 от  $\forall$  10 бита не предават информация, а са необходими за *синхронизация*, т.е 20% горна граница

- *горната граница* - намалява се чрез по-големи битови блокове → нараства грешката при приемане

- по-голяма ефективност - *синхронно предаване*

## Синхронно предаване

- ❑ Синхронно предаване - битовия блок се предава без S и SE
  - блокът – с по-голяма дължина
  - синхронизация предавател/приемник – чрез *уводен и заключителен* бит
  - **рамка** (фрейм) – уводна, заключителна, управляваща информация и данни
- ❑ Рамков формат за синхронизирано предаване
  - размерът ~ от контролните процедури на данните - обикновено < 100 bits
  - увод и заключение - 8 bits флаг



## Синхронно предаване

---

- *полета за управление* - данни за протоколната дължина
- пример: **популярната схема за достъп HDLS**
  - ✓ съдържа 48 бита за управление, увод и заключение
  - ✓ при 1000 bits блок → ∇ рамка 48 bits *горна граница*
  - ✓ данни -  $1000 \times 8 = 8000$  bits
  - ✓ *процентова горна граница* -  $48/8048 = 0,6\%$

# Грешки при предаването на цифрови данни

---

**1. Грешки при последователно предаване**

**2. Откриване на грешки чрез проверка за  
еднаквост**

2.1. Информационен излишък

2.2. Процес на откриване на грешки

2.3. Проверка за еднаквост

**3. Циклична проверка за информационен  
излишък**

**4. Корекция на грешките**

## Информационен излишък

---

- ❑ Грешките при предаване - откриват се след декодиране на получените информационни данни
- ❑ Методи за откриване на грешки - основават се на ***информационен излишък***
  - концепция – в *предавателя* допълнителна информация от малка група битове в края на  $\forall$  блок
  - *приемник* - проверява цялата входяща информация
  - *проверка за достоверност* - при *потвърждение* излишната част от битове се премахват

## Информационен излишък

- За предавателя - съществуват *вероятности за грешки*, променящи на един или повече битовете във  $\forall$  рамка:
  - вероятност за битовата грешка (BER) -  $P_b$
  - вероятност рамката да се приема без грешки -  $P_1$
  - вероятност рамката да се приема без открити грешки при използване на алгоритъм за откриване -  $P_2$
  - вероятност рамката да се приема с открити грешки при използване на алгоритъм за откриване -  $P_3$
- За  $\forall$  рамка с  $F$  bits при  $P_3=0$  и  $P_b=const$ :

$$P_1 = (1 - P_b)^F \qquad P_2 = 1 - P_1$$

- ❖ при  $\uparrow$  на вероятността  $P_b \rightarrow$  вероятността  $P_1 \downarrow$
- ❖ при  $\uparrow$  на дължината  $F \rightarrow$  вероятността  $P_1 \downarrow$

## Информационен излишък

### □ Пример: цифрова мрежа с интегрирани услуги ISDN

➤ основна задача в 64-kbps канал: до 90% от 1 min наблюдавани време-интервали: BER <  $10^{-6}$

➤ при 1000 bits рамка:

за 1 ден се предават до  $5,529 \times 10^6$  bits → вероятността

за открити BER -  $P_2 = 1/(5,529 \times 10^6) = 0,18 \times 10^{-6}$

➤ за BER  $P_b = 10^{-6}$  →

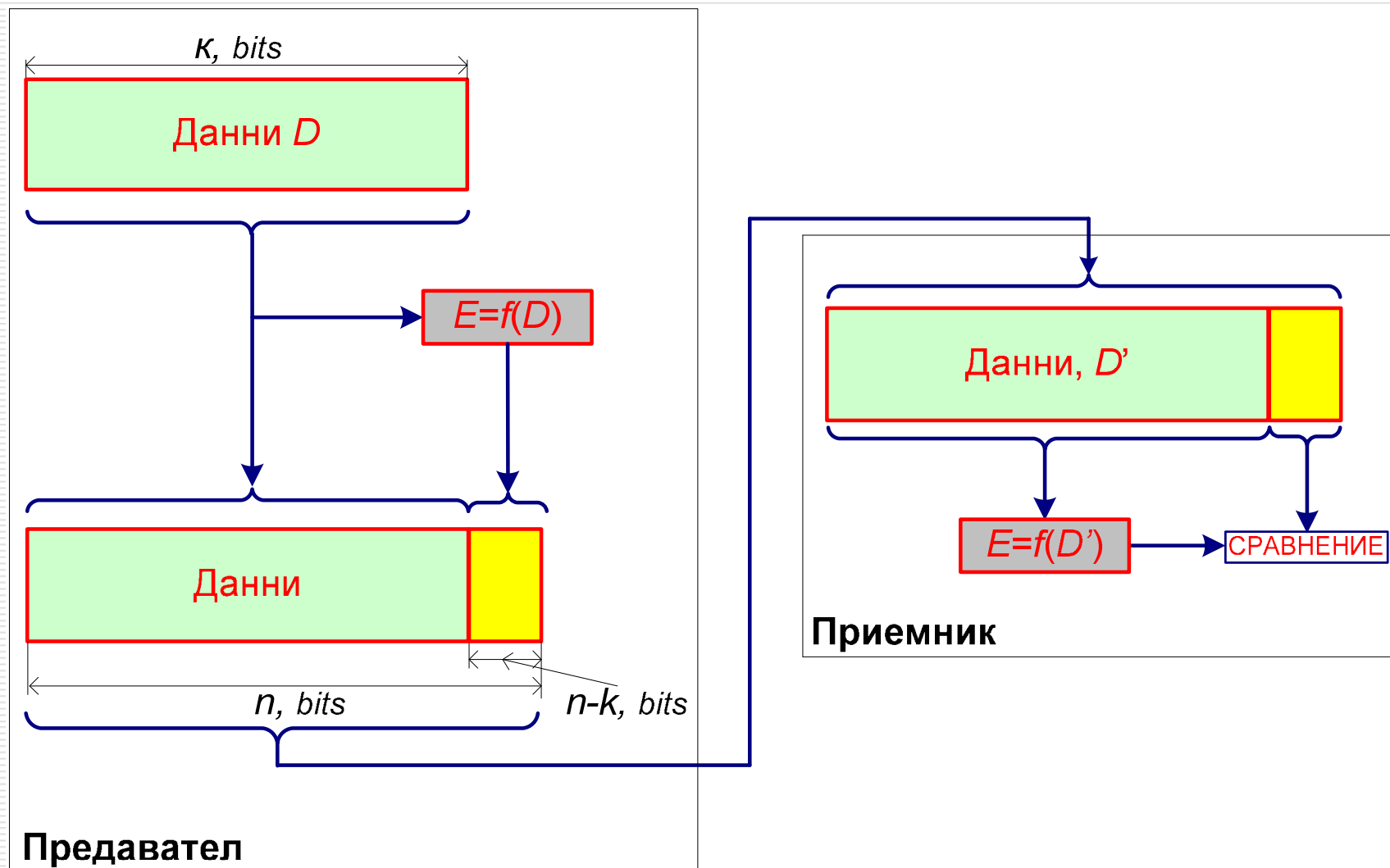
вероятността за приемане на рамката без битови грешки

$$P_1 = (0,999999).1000 = 0,999$$

вероятността за открити BER

$$P_2 = 10^{-3}, \text{ т.е. три порядъка по-голяма}$$

# Процес на откриване на грешки



# Процес на откриване на грешки

---

## ❖ *предавател*

- ✓ допълнителни  $(n-k) < k$  bits - образуват **откриващия код**  $E$  като функция на предадените битове
- ✓ изчисляват се **контролни битове**, които се прибавят към данните  $D$

## ❖ *приемник*

- ✓ разделя пристигащата рамка на  $k$  bits данни и  $(n-k)$  bits откриващ код  $E$
- ✓ **сравнява** приетите  $(n-k)$  bits и получения контролен код  $E=f(D')$
- ✓ **открива** грешка при несъответствие

# Процес на откриване на грешки

---

□ Методи за проверка на информационния излишък

## 1. Проверка за еднаквост PC (*Parity Check*)

- *вертикален контрол за излишък VRC*
- *проверка по четност и нечетност LRC*

## 2. Циклически контрол CRC (*Cyclic Redundancy Check*)

- използва се **контролна сума** FCS (*Frame Check Sequence*)
- VRC, LRC и CRC - в каналния OSI слой
- FCS се използва и в по-горните OSI слоеве

## Проверка за еднаквост

### ❑ Вертикален контрол за излишък VRC

- най-използван механизъм, защото е най-евтин
- *разряд за проверка на четността* – добавя се излишен бит към  $\forall$  рамка
- прочита се общия брой **1**
- пример: **предаване на думата *world* (свят)**

❖ предавател- в ASCII код

1110111 1101111 1110010 1101100 1100100

**w**

**o**

**r**

**l**

**d**

❖ първите четири символа - четен брой **1** →  
проверка по четност **0**

## Проверка за еднаквост

❖ символ  $d$  - нечетен брой **1** → проверка по четност **1**

11101110 11011110 11100100 11011000 1001001

w

o

r

1

d

### □ **Проверка за нарушаване на четност/нечетност LRC**

- информационните блокове се организират в таблица от редове и стълбове
- данните се поставят в редове и се проверяват по четност
- за всеки стълб се прави вертикален контрол и се създава допълнителен ред
- проверка в приемника - блокът се отхвърля, ако не изпълнява правилото за четност

# Проверка за еднаквост

➤ пример: предаване на 32 бита – в таблица (4x8)



# Грешки при предаването на цифрови данни

---

1. Грешки при последователно предаване
2. Откриване на грешки чрез проверка за еднаквост
3. Циклична проверка за информационен излишък
  - 3.1. Принципи на цикличната проверка
  - 3.2. Сума по модул 2
  - 3.3. Полиноми
  - 3.4. Цифрова логика
4. Корекция на грешките

## Принципи на цикличната проверка

---

- ❑ **Цикличната проверка** CRC - най-мощното средство за контрол на грешките с информационен излишък
- ❑ Основава се на *проверка по четност* чрез двоично делене вместо събиране на битове
  - *k*-bits блок от данни
  - *предавател*- генерира *контролна сума FCS* -  $(n-k)$  bits
  - *приемник*- дели пристигащата рамка на FCS; при *резултат без остатък* – **няма грешка при предаване**
- ❑ Необходими свойства на CRC цикличната проверка
  - рамката да се дели точно на даден делител
  - остатъкът да има един бит по-малко от делителя

# Сума по модул 2

---

□ Методи за генериране за циклична проверка

■ **сума по модул 2**

■ **полиноми**

■ **цифрова логика**

□ Сума по модул 2

➤ използва *двоично събиране без пренос*  $\cong$  изключващо ИЛИ (XOR)

➤ примери

$$\begin{array}{r} 1111 \\ + \\ \hline 1010 \\ \hline 0101 \end{array}$$

$$\begin{array}{r} 1111 \\ - \\ \hline 0101 \\ \hline 1010 \end{array}$$

$$\begin{array}{r} 11001 \\ \times \\ \hline 11 \\ \hline 11001 \\ \hline 11001 \\ \hline 101011 \end{array}$$

## Сума по модул 2

---

- дефинирани променливи
  - $n$  bits рамка за предаване -  $T$
  - $k$  bits от информационната рамка  $T$  -  $D$
  - последните  $(n-k)$  bits от  $T$  - контролна сума  $F$
  - шаблон от  $(n-k+1)$  bits – дефинира делителя  $P$
- Основна задача при сумиране по модул 2 – деленето  $T/P$  да е **без остатък**

$$T = 2^{n-k} D + F$$

- в *предавателя* - частно  $Q$  и остатък  $R$  (винаги поне с 1 бит по-малък от делителя  $P$ )

$$\frac{2^{n-k} D}{P} = Q + \frac{R}{P}$$

## Сума по модул 2

---

- в приемника - деление с контролната сума

$$\frac{T}{P} = \frac{2^{n-k} D + R}{P} = \frac{2^{n-k} D}{P} + \frac{R}{P}$$

- задача →  $T$  да е точно делим на  $P$
- определяне на грешките - в  $n$ -битова рамка се представят като поле от **1** за всеки грешен бит → приета рамка  $T_r$ ; предадена рамка  $T$ ; шаблон за грешки  $E$ ;  $\oplus$  - изключващо ИЛИ

$$T_r = T \oplus E$$

## Сума по модул 2

---

### □ Пример

#### ■ задание

**съобщение**  $D$  ( $k=10$  bits)  $\rightarrow$  1010001101

**шаблон**  $P$  (6 bits)  $\rightarrow$  110101

**дължина на рамката на предаване**  $T - n = 15$  bits

**контролна сума**  $R - (n-k)=5$  bits;  $R = ?$

■ намира се произведението  $2^5 \times D \rightarrow$  101000110100000

■ произведението  $2^5 \times D$  се дели на шаблона  $P$  - получават се частното  $Q$  и остатъкът  $R$





## Полиноми

- CRC генераторът се представя като алгебричен многочлен
  - ❖ стойностите на полинома се представят като двоични коефициенти на променливата  $X$
  - ❖ пример  $D = 110011 \rightarrow D(X) = X^5 + X^4 + X^1 + X^0$
  - ❖ аритметичните операции - също по модул 2

$$\frac{X^{(n-k)}D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$T(X) = X^{(n-k)}D(X) + R(X)$$

# Полиноми

## □ Пример (същия като слайд 28)

### ■ задание

**съобщение**  $D$  ( $k=10$  bits)  $\rightarrow$  1010001101

$$D(X) = X^9 + X^7 + X^3 + X^2 + 1$$

**шаблон**  $P$  (6 bits)  $\rightarrow$  110101

$$P(X) = X^5 + X^4 + X^2 + 1$$

**контролна сума**  $R$  -  $(n-k)=5$  bits;  $R \rightarrow ?$

### ■ получава се $R \rightarrow$ 01110,

т.е.  $R(X) = X^3 + X^2 + X^1$

### ■ резултат - чрез полиномно деление

## Полиноми

---

- ❑ Грешката  $E(X)$  не може да се открие, ако се дели на  $P(X)$
- ❑ Доказва се, че могат да се открият някои грешки  $E(X)$  дори, когато не са делими на шаблона  $P(X)$
- ❑ Използват на стандартни полиноми за шаблони  $P(X)$ 
  - ❖ предаване на ATM хедър – CRC-8 =  $X^8+X^2+X+1$
  - ❖ предаване на 6 bits думи - CRC-12 =  $X^{12}+X^{11}+X^3+X^2+X+1$
  - ❖ предаване на 8 bits думи
    - за Европа CRC-16 =  $X^{16}+X^{15}+X^2+1$
    - за САЩ CCITT =  $X^{16}+X^{12}+X^5+1$
  - ❖ синхронно предаване в IEEE 802 LAN - CRC-32 =

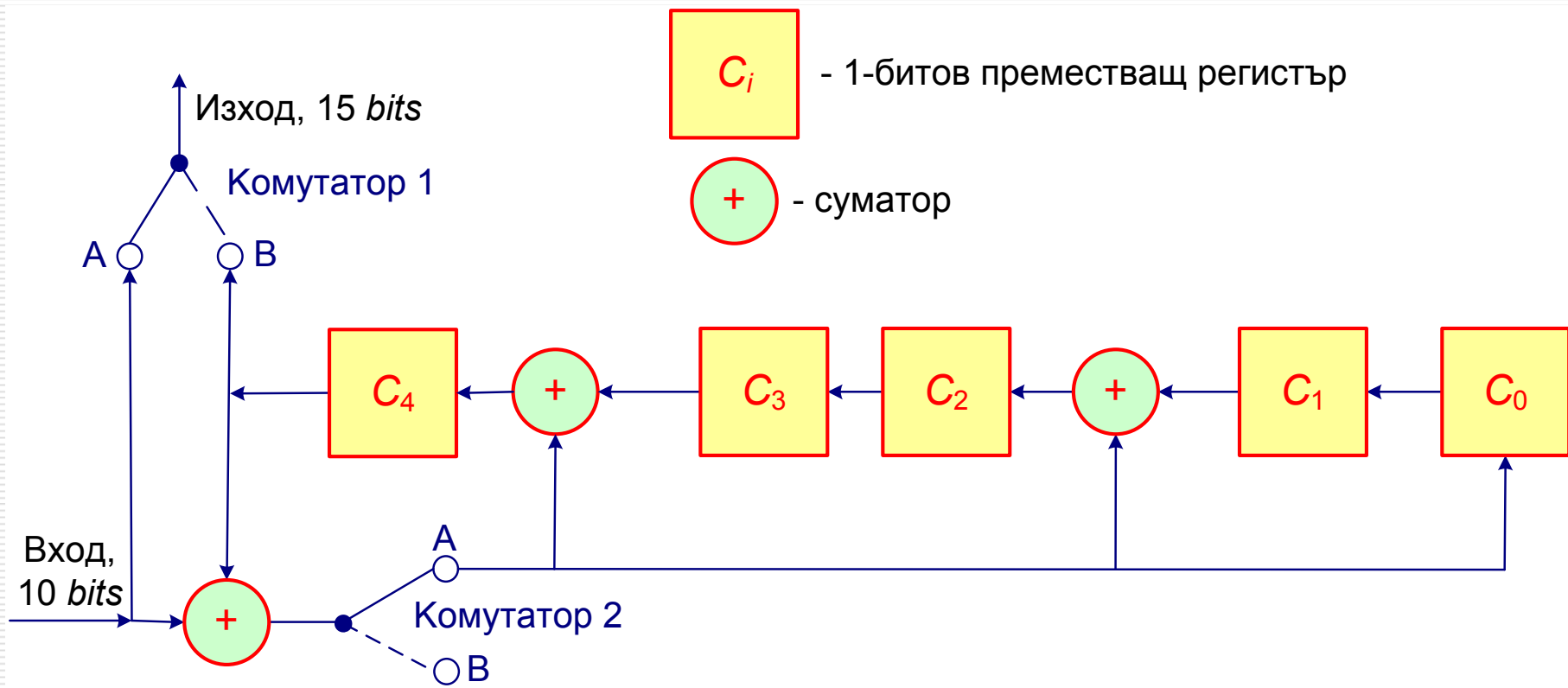
## Цифрова логика

---

- ❑ CRC генераторът се представя чрез схема, съдържаща изключващо ИЛИ и преместващ регистър
  - *преместващ регистър* - последователност от 1bit запомнящи клетки, които се преместват последователно чрез тактуване
  - регистърът съдържа  $(n-k)$  bits равни на дължината на контролната сума FCS
  - повече от  $(n-k)$  елементи за изключващо ИЛИ
  - наличието/липсата на *суматор* съответства на наличието /липсата на полином в шаблона  $P(X)$  (изключение полиномните членове 1 и  $X^{n-k}$ )

# Цифрова логика

- Пример: преместващ регистър за делене с полинома  $P(X) = X^5 + X^4 + X^2 + 1$



# Цифрова логика

■ Данни  $D=1010001101$

стъпка	$C_4$	$C_3$	$C_2$	$C_1$	$C_0$	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	ИЗХОД
иниц.	0	0	0	0	0	1	1	1	1
1	1	0	1	0	1	1	1	1	0
2	1	1	1	1	1	1	1	0	1
3	1	1	1	1	0	0	0	1	0
4	0	1	0	0	1	1	0	0	0
5	1	0	0	1	0	1	0	1	0
6	1	0	0	0	1	0	0	0	1
7	0	0	0	1	0	1	0	1	1
8	1	0	0	0	1	1	1	1	0
9	1	0	1	1	1	0	1	0	1
10	0	1	1	1	0				

## Корекция на грешките

---

- ❑ В някои протоколни архитектури *корекцията на грешки съвпада с откриването на грешки*
  - например, протоколи за управление на връзките (HDLC) и транспортни протоколи (TCP)
  - използва се *откриващ код* за предаване на блока
  - за коригиране - *приемникът иска от предавателя **повтаряне на цялата информация***
- ❑ Подходът е неадекватен за безжични мрежи (причини)
  - висока скорост на предаване – *мултиплициране* на грешки, които имат малка вероятност
  - много по-голямо закъснение при разпространението от времето за предаване на единична рамка

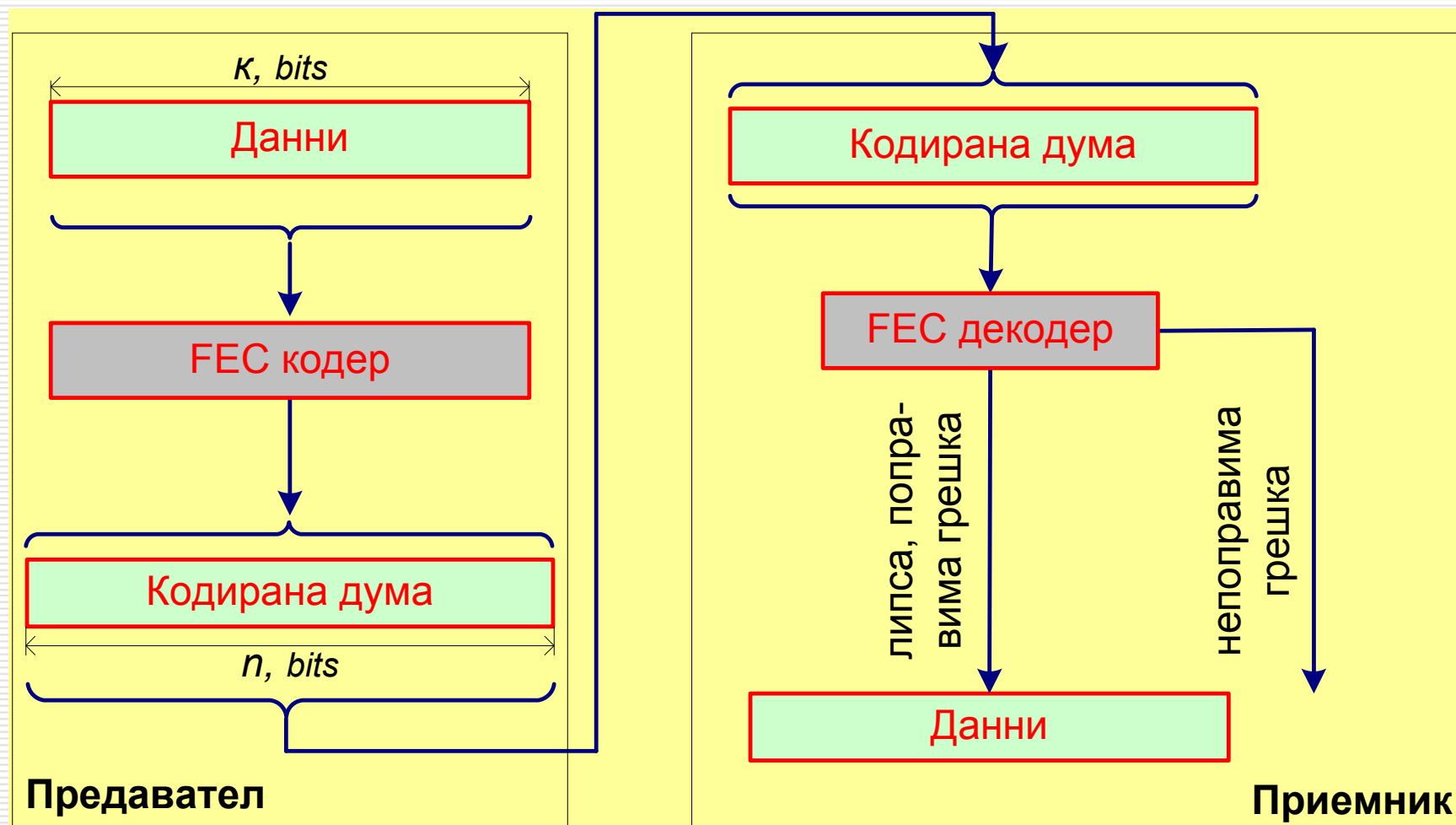
# Корекция на грешките

---

- ❑ Приемникът използва **коригиращ код**
  - ❖ по-сложен от *кода за откриване на грешки*
  - ❖ извършва повторно подреждане в ред на по-голям брой битове
  - ❖ не е ефективен при многобитови грешки и се използват за корекция *само* на едно-, дву- и триразрядни грешки
- ❑ Процес на коригиране на грешките
  - ❖ сравнение при приемане на  $k$ -bits блок от данни с  $n$ - bits **кодова дума** ( $n > k$ )
  - ❖ използване на FEC (*forward error correction*) кодиращо и декодиращо устройство

# Корекция на грешките

- ❑ Процес на коригиране на грешката



## Корекция на грешките

---

□ Четири възможни случая при FEC декодирането:

- 1. Няма битова грешка** - на изхода на декодерът се получава *оригиналния* блок от данни
- 2. Откриване и коригиране** на образци от грешки - декодера съпоставя този *оригиналния* и *приетия* блок от данни
- 3. Откриване без коригиране** на образци от грешки - декодера докладва за *некоригируема* грешка
- 4. Не откриване** на грешки – в редки случаи декодерът не намира различия при съпоставянето на *оригиналния* и *приетия* блок от данни

# Корекция на грешките

- Принцип на **кодирането** за коригиране
  - използва се **разстоянието на Хеминг**  $d(v_1, v_2)$  - между  $n$ -bits последователности с брой на битовете  $v_1$  и  $v_2$
  - пример:  $v_1 = 011011$  и  $v_2 = 110001 \rightarrow d(v_1, v_2) = 3$
- Изпращане на блокове от данни с дължина  $k$  бита
  - всеки  $k$ -bits блок се изпраща поотделно
  - сравнява се всяка  $k$ -bits последователност с уникална  $n$ -bits дума
- Пример:
  - задание -  $k=2; n=5$

Блок данни	Кодова дума
00	00000
01	00111
10	11001
11	11110

## Корекция на грешките

---

- приемник – блок данни с кодова дума 00100
- не валидна кодова дума → приемникът открива грешка – от 1 до 5 бита
- необходима корекция на 00100 за преобразуване във валидна кодова

кодова дума	коригирани битове
00000	1
00111	2
11001	4
11110	3

- най-вероятна кодова дума, която е изпратена е 00000 и съответства на блок от данни 00
- разстояния на Хеминг:  $d(00000, 00100)=1$ ;  $d(00111, 00100)=2$ ;  $d(11001, 00100)=4$ ;  $d(11110, 00100)=3$

# Корекция на грешките

---

- Проектирането на кодови блокове включва следните правила
  1. За зададени стойности на  $n$  и  $k$ , се търси най-голяма стойност на  $d_{\min}$
  2. Кодът трябва относително лесно да се кодира и декодира, изисква минимална памет и време за обработка
  3. Броят на допълнителните битове  $(n-k)$  трябва да е *малък*, за да се намали обхвата на грешките
  4. Броят на допълнителните битове  $(n-k)$  трябва да е *голям* за да се намали честотата на грешките

# ПРОТОКОЛИ ЗА АВТОМАТИЧНО УПРАВЛЕНИЕ НА КАНАЛНИЯ СЛОЙ

---

## 1. Автоматично управление на потока

1.1. Управление чрез спиране и изчакване

1.2. Управление чрез плъзгащ прозорец

## 2. Автоматично управление на грешката

## 3. Протокол за управление от високо ниво

# Автоматично управление на потока

---

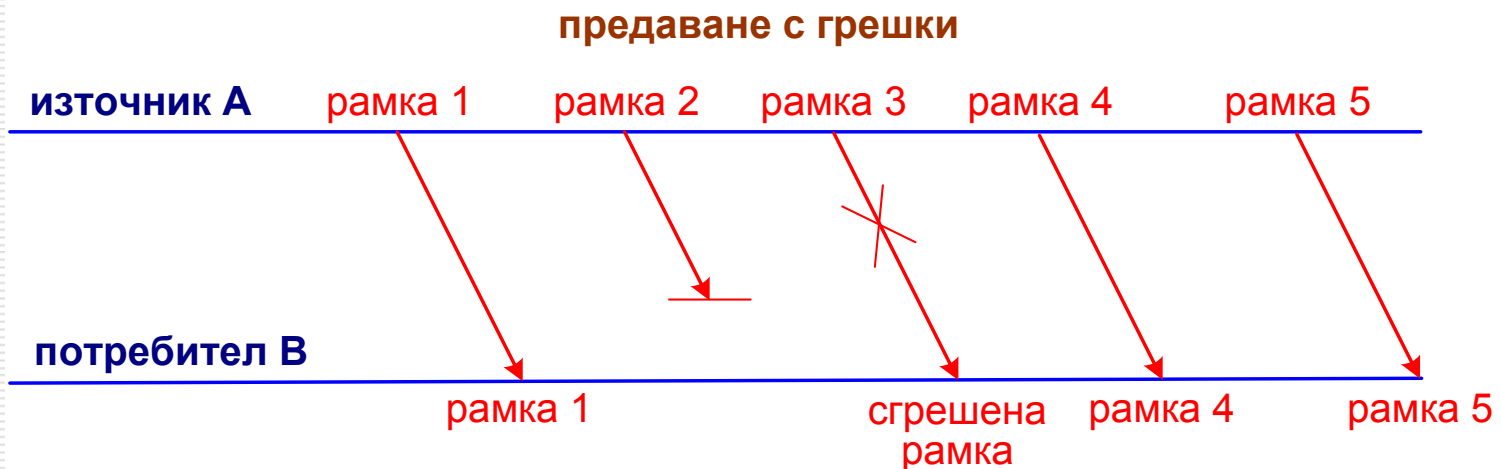
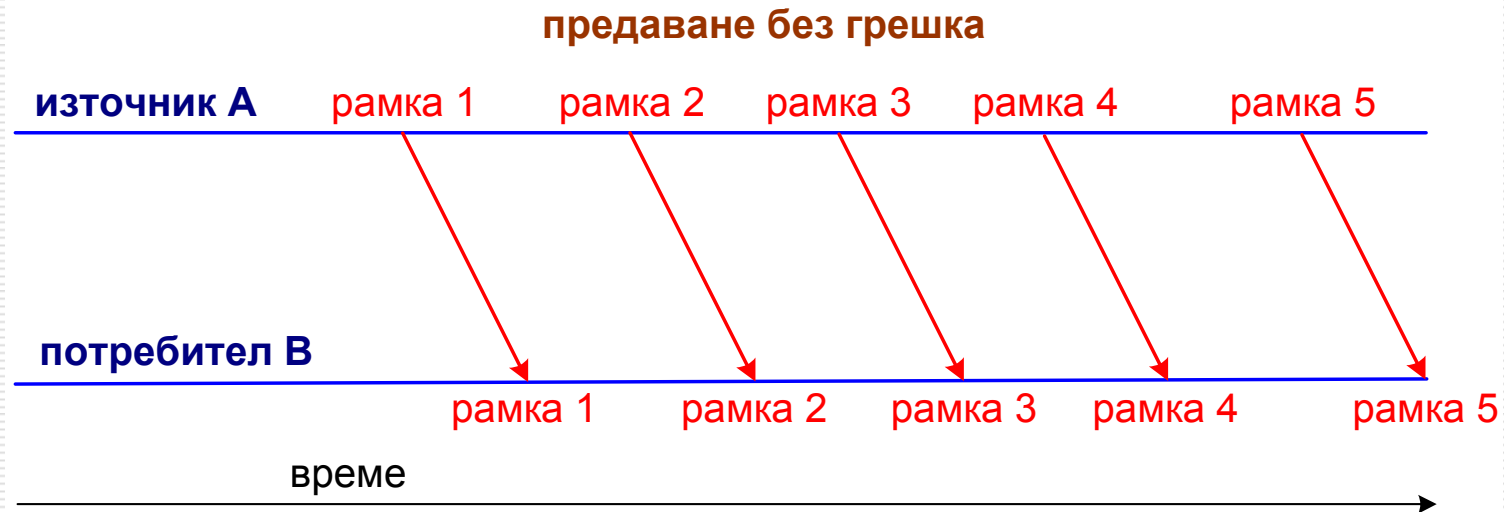
- Основни изисквания за ефективно предаване на данните
  - ❖ **синхронизация на фреймовете** – разпознаване на началото и края на всеки фрейм
  - ❖ **автоматично управление на потока** - приемникът регулира потока от данни без да препълва своя буфер за чакане
  - ❖ **автоматично управление на грешки** – откриване и коригиране на битовите и динамични грешки
  - ❖ **адресиране** - определяне на идентичността на потребителите в компютърната мрежа
  - ❖ **автоматично управление на данните** в канала за връзка – разделяне на управляващите от информационните битове
  - ❖ **автоматизирано управление на канала** – стартиране, настройване и прекратяване на продължителен обмен на данни

# Автоматично управление на потока

---

- ❑ **Автоматично управление на потока** (*Flow Control*)
  - метод, чрез който източника на данни не претоварва потребителя
  - приемането на данните се реализира чрез буфер със зададена максимална дължина
  - при отсъствие на управление *буферът* на приемника се препълва с данни и част от тях се отхвърлят
- ❑ Механизъм на управлението при *предаване без грешки*
  - ✓ информацията се изпраща като поредица от рамки (данни + управляваща информация)
  - ✓ времето за предаване  $\sim$  на дължината на рамката
- ❑ Механизъм на управлението при *предаване с грешки*

# Автоматично управление на потока



## Управление чрез спиране и изчакване

---

- Управление чрез **спиране и изчакване** (*Stop-and-wait*)
  - приемникът получава дадена рамка и изпраща на потвърждение до източника
  - източникът изчаква получаването на това потвърждение, след което изпраща следваща рамка
  - приемникът може да спре потока от данни като задържи потвърждението
- Източникът разделя цялата информация на по-малки съобщения, които се предават чрез много рамки
  - размерът на буфера е ограничен
  - при продължително предаване вероятността за грешки е по-голяма
  - при по-малки рамки грешките се откриват по-рано и се препредава по-малък обем от информация

## Управление чрез спиране и изчакване

---

- ❑ Желателно е всяка изпращаща станция (например, LAN) да не използва за дълъг период *поделената линия за връзка*, защото предизвикват големи закъснения
- ❑ **Битова дължина на линията**  $B$  –  $\sim$  ефективността на предаване

$$B = R \times \frac{d}{V}$$

$R$  - скорост на предаване на линията, bps

$d$  - дължина на линията, m

$V$  - скорост на разпространение, m/s

- ❑ **Закъснението при разпространение**  $a$  - времето за предаване на един бит от предавателя към приемника

$$a = \frac{B}{L}$$

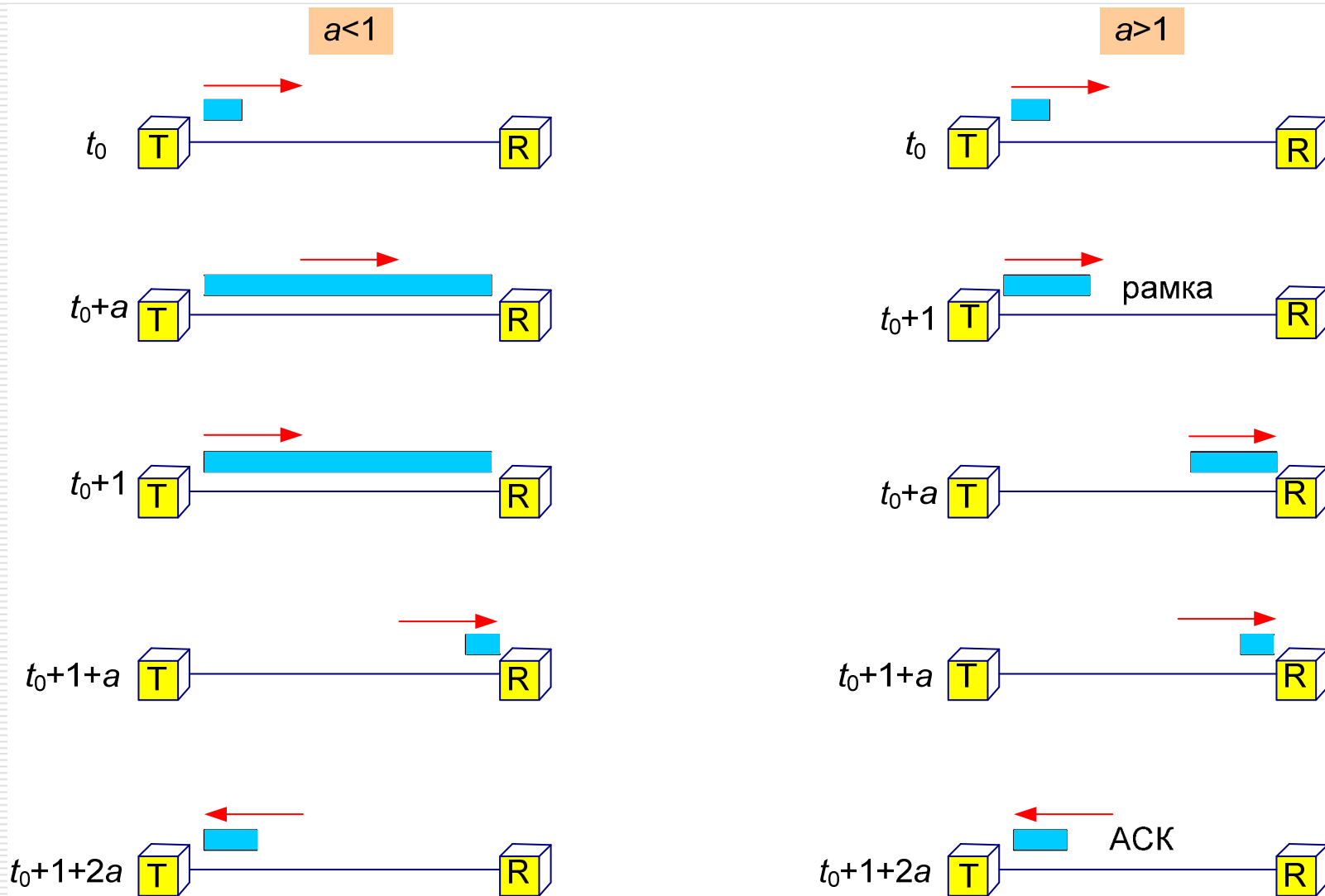
$L$  - дължина на рамката в битове

## Управление чрез спиране и изчакване

---

- времето за разпространение е по-малко от времето за предаване  $a < 1$ 
  - рамката е достатъчно дълга и първите битове пристигат до местоназначението си преди източникът да е завършил предаването
- времето за разпространение е по-голямо от времето за предаване  $a > 1$ 
  - източникът завършва предаването на рамката преди първите битове да са достигнали до приемника
  - 4 момента на процеса на предаване на рамката
  - връщането на потвърждението АСК
- методът не е подходящ за високи скорости на предаване и за големи разстояния между А и В

# Управление чрез спиране и изчакване



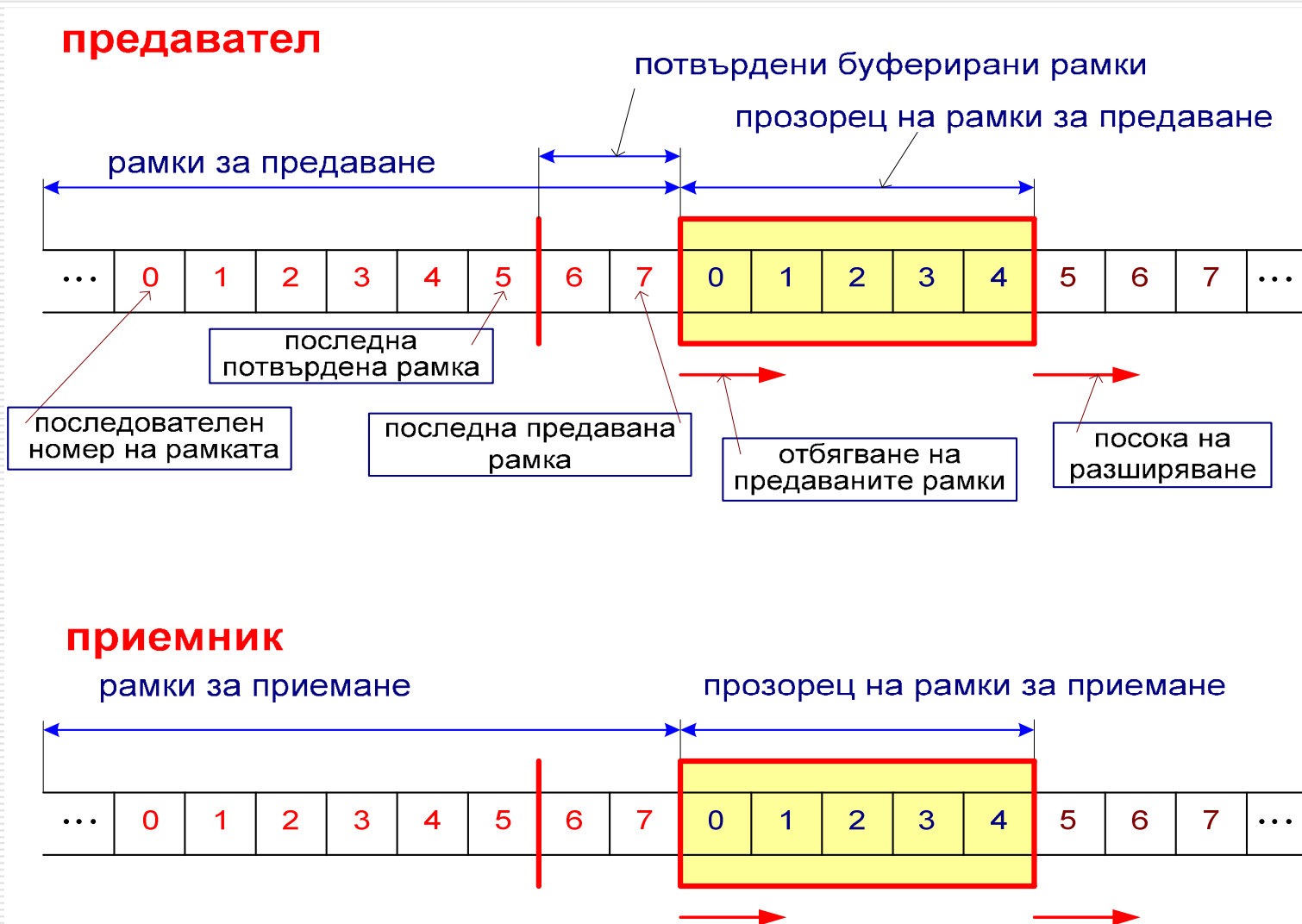
## Управление чрез плъзгащ прозорец

---

- Същност на метода за управление
  - управление на линия в режим *пълнен дуплекс* - за два сървъра – **A** и **B**
  - потребител **B** приема до  $W$  рамки в буфера, които източник **A** може да изпрати без потвърждение
  - всяка рамка има пореден номер, чрез който се следи За да следи нейното потвърждаване при приемане от **B**
  - потвърждението подсказва скрито, че **B** е готов за получаване следващите  $W$  рамки → приемане на последователност от рамки
  - **A** - списък с поредни номера на рамки, готови за изпращане, а **B** - списък с поредни номера, готови за приемане → тези списъци *плъзгащ прозорец*

# Управление чрез плъзгащ прозорец

- Пример: 3 bits последователност от рамки



## Управление чрез плъзгащ прозорец

---

- Пример: 3 bits последователност от рамки (продължение)
  - рамките са номерирани от 0 до 7
  - правоъгълникът - показва рамките, които могат да се изпратят- 5 рамки
  - при предаване на рамка прозорецът *намалява*, а при получаване – *нараства*
  - рамките между вертикалната черта и прозореца са изпратени, но не са потвърдени
  - максималнодопустимият размер на прозореца за дадена дължина не е задължителен
  - приемникът може да приеме до 7 рамки след последата потвърдена
  - за  $k$  bits последователност поредицата от рамки е от 0 до  $2^k-1$ , т.е. максималният размер на прозореца е  $2^k-1$

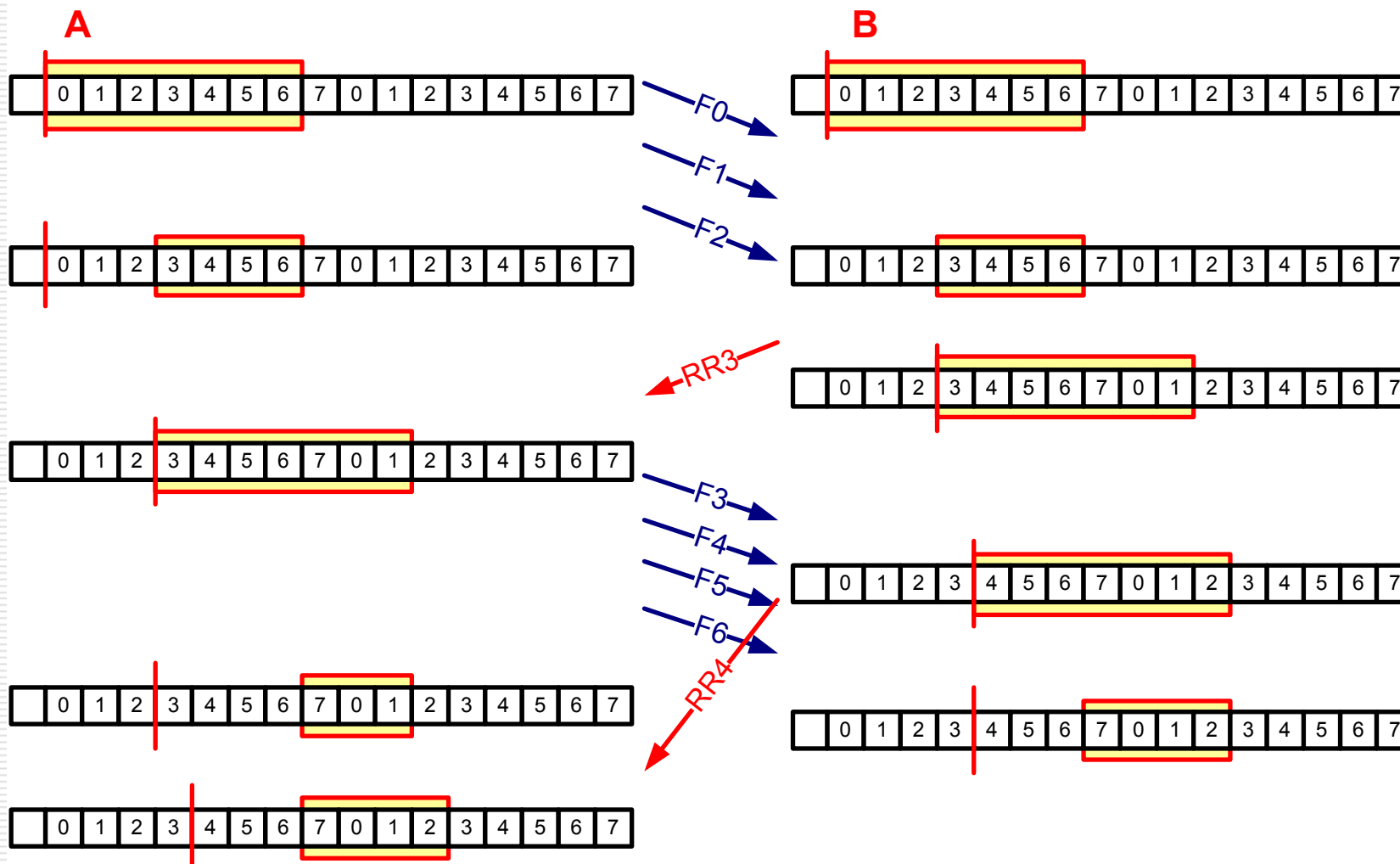
## Управление чрез плъзгащ прозорец

---

- *Протоколът за управление на потока* позволяват на приемния сървър да спре потока от рамки като изпрати съобщението **«неподготвен за получаване»** RNR (*receive not ready*)
  - ✓ потвърждава предишните приети рамки
  - ✓ забранява бъдещия трансфер
  - ✓ пример: съобщението RNR5 - приемникът е получил всички рамки до 4<sup>-та</sup> включително
  - ✓ ново предаване - след потвърждение за повторно отваряне на прозореца

# Управление чрез плъзгащ прозорец

- Двупосочно предаване - режим *пълен дуплекс*



## Управление чрез плъзгащ прозорец

---

- поддържат се 2 прозореца – един за предаване и един за приемането
- ефективно предаване - всяка рамка включва поле, което съдържа поредния номер на тази рамка и поле, което съдържа поредния номер на рамката, използвана за потвърждение
- за потвърждение –отделна **потвърждаваща рамка** RN (*receive ready*) или RNR (*receive not ready*)
- изпращане на информация без потвърждение – повтаря се номера на последното изпратено потвърждение

# Протоколи за автоматично управление на каналния слой

---

1. Автоматично управление на потока

**2. Автоматично управление на грешката**

2.1. Автоматично повторение на заявката

2.2. Повторение чрез спиране и изчакване

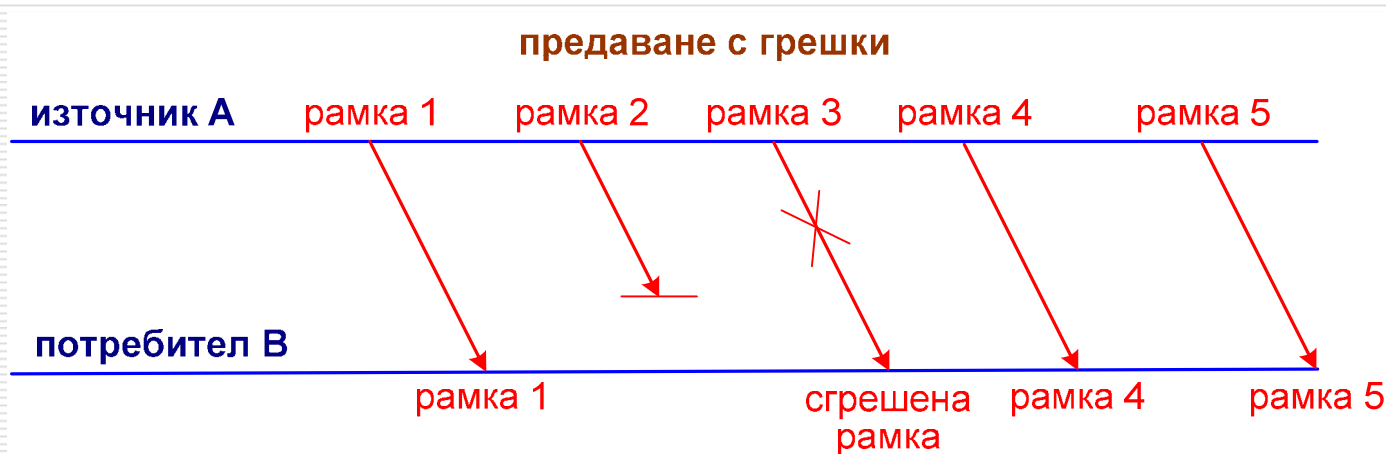
2.3. Повторение чрез отиване и връщане

2.4. Повторение чрез избирателно отхвърляне

3. Протокол за управление от високо ниво

## Автоматично повторение на заявката

- ❑ Управлението на грешката е средство за откриване и коригиране на грешките при предаването на рамките
- ❑ Данните *предават* чрез на последователни рамки, които се *приемат* по реда на изпращане, т.е за всяка предадена рамка е възможно закъснение при приемане
- ❑ 2 вида грешки при предаването на данни
  - ❖ загубена рамка - не достига приемната страна
  - ❖ повредена рамка – някои от битовете са сгрешени при приемане



## Автоматично повторение на заявката

---

- ❑ Алгоритъм на *управлението на грешката*
  - ❖ *откриване на грешките* при предаване и приемане на информационни данни
  - ❖ *положително потвърждаване* - приемникът декларира всички успешно получени (без грешка) рамки
  - ❖ *препредаване след изчакване* – източникът препредава отново рамките, които не са потвърдени за определено време след изпращането
  - ❖ *отрицателно потвърждение и препредаване* – приемникът връща отрицателно потвърждение за рамките, в които е открита грешка и източникът препредава отново
- ❑ Обобщение на тези правила- **автоматично повторение на заявката** ARQ (*automatic repeat request*)

## Автоматично повторение на заявката

---

- ❑ Основна задача на управлението на грешката – подобряване на надеждността на всеки канал за връзка с помощта на ARQ
- ❑ 3 стандартизирани версии на ARQ, основани на методите и техническите средства за управление на потока
  - ❖ **повторение чрез спиране и изчакване** (*Stop-and-Wait ARQ*)
  - ❖ **повторение чрез многократно отиване и връщане** (*Go-Back-N ARQ*)
  - ❖ **повторение чрез избирателно отхвърляне** (*Selective-Reject ARQ*)

## Повторение чрез спиране и изчакване

---

- ❑ Stop-and-Wait ARQ се използва при *управлението на потока чрез спиране и изчакване*
  - източникът предава дадена рамка, след което изчака **потвърждение** ACK (*acknowledgment*)
  - таймер в приемника за откриване на грешки и отхвърляне на сгрешените или загубени рамки
  - дадена рамка се изпраща повторно, ако края на времето, зададено в таймера не е получено потвърждение
  - източникът трябва да пази копие на изпратената рамка до получаване на потвърждение





## Повторение чрез отиване и връщане

---

- приемник- потвърждава постъпващите рамки чрез готово приемане RR (*receive ready*)
  - при грешка в рамката - отрицателно потвърждение REJ (*reject*)
  - отхвърля всички следващи рамки, докато грешната рамка не се получи правилно
  - източникът получава REJ и препредава всички следващи рамки
- Възможности за предаване между станции **A** и **B**
1. Повредена (невалидна) рамка (отхвърляне от получателя)  
→ REJ4 – изпраща повторно рамка4
  2. Повредено RR съобщение - RR7 –предадени до рамка6
  3. Таймерът на **A** изтича - RR рамка с  $P=1$  бит, **B** прочита тази рамка и изпраща на **A** RR1
- Максимален размер на прозореца - ограничен до  $(2^3-1)$

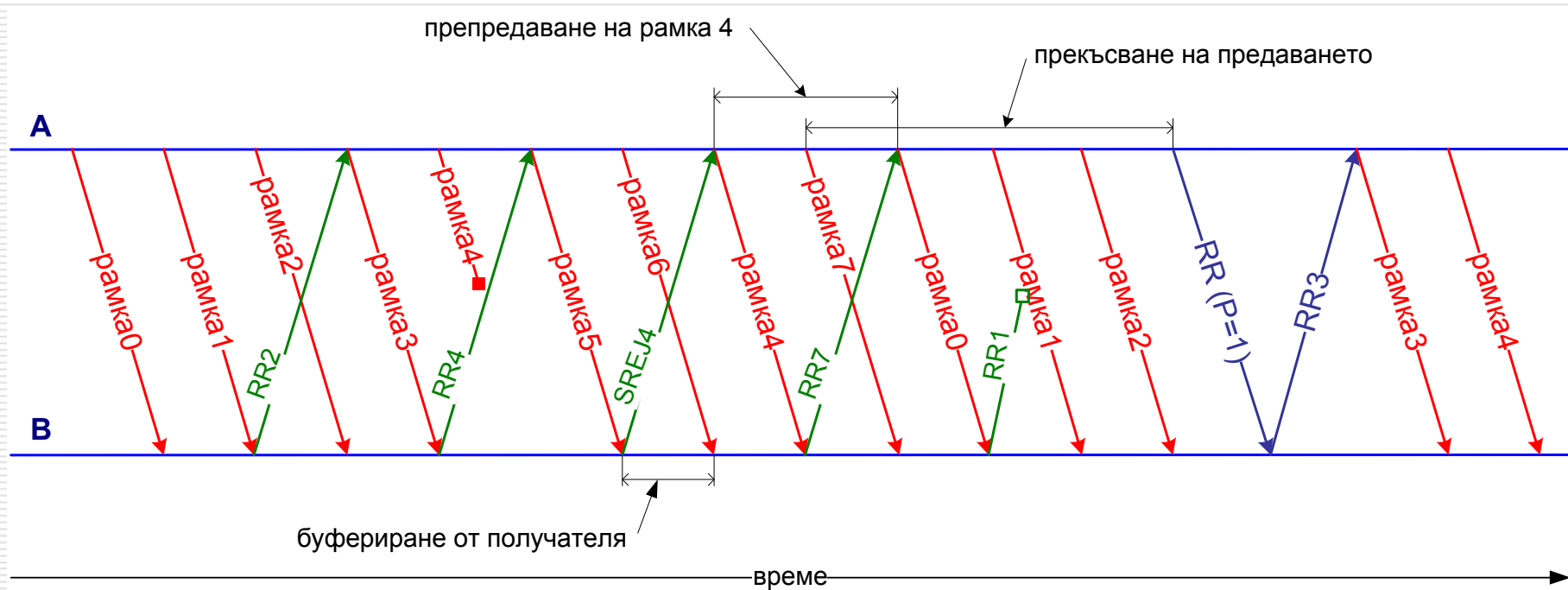
## Повторение чрез избирателно отхвърляне

---

- ❑ Автоматично повторение чрез избирателно отхвърляне (*Selective-Reject ARQ*) –препраща рамки с отрицателно потвърждение SREJ или рамки с изтекъл таймер
- ❑ По-ефективен от Go-Back-N ARQ - намалява препращаната информация
- ❑ Използва се по-рядко
  - Приемник - буфер за запазване на SREJ потвърждението, докато грешната рамка се препраща; логика за повторно вмъкване на рамката на подходящото място
  - Предавател – логика за изпращане на сгрешената рамка в съответствие с последователността

# Повторение чрез избиращелно отхвърляне

- ❑ Размерът на прозореца е по-малък от този при многократно повторение (Go-Back-N ARQ)
- ❑ Пример: прозореца, състоящ се от 3-bits рамкова последователност



## Повторение чрез избирателно отхвърляне

---

- ❖ **A** изпраща рамки от 0 до 6 към **B**
  - ❖ **B** получава 7-те рамки и изпраща общо потвърждение чрез съобщението RR7
  - ❖ заради смущения, съобщение RR7 е загубено
  - ❖ **A** изчаква и препредава рамка0
  - ❖ **B** вече е задала на прозореца за приемане на рамки 7,0,1,2,3,4 и 5; **B** предполага, че рамка7 е загубена и приема рамка0
- Основен проблем - прозорците на приемната и предавателната страна се застъпват → решение максималната дължина на прозореца  $\leq$  от половината последователни рамки (4 не потвърдени рамки извън прозореца)

# Протоколи за автоматично управление на каналния слой

---

1. Автоматично управление на потока
2. Автоматично управление на грешката
- 3. Протокол за управление от високо ниво**
  - 3.1. Основни характеристики
  - 3.2. Структура на полетата
  - 3.3. Фази на HDLC операцията

## Основни характеристики

---

- ❑ HDLC протокол (*High-Level Data Link Control Protocol*) - основен протокол на каналния OSI слой, стандартизиран по ISO 3009 и ISO 4335
- ❑ Разнообразни приложения - определят 3 типа *станции*, 2 *конфигурации* на канала и 3 *режима на предаване*
  - типове станции
    - ✓ **първична** – отговаря за управлението на операциите на канала; рамките, свързани с нея – команди
    - ✓ **вторична** – оперира под управлението на първичната станция; рамките, свързани с нея – отговори
    - ✓ **комбинирана** – обединява характеристиките на първичната и вторичната; работи с команди и отговори

## Основни характеристики

---

- конфигурации на канала
  - ✓ **не балансирана** – състои се от една първична и една или повече вторични станции; поддържа режими на предаване - дуплекс и полудуплекс
  - ✓ **балансирана** -състои се от 2 комбинирани станции; поддържа режими на предаване - дуплекс и полудуплекс
- режими на предаване
  - ✓ **с нормален отговор** NRM (*Normal Response Mode*) – при не балансирана конфигурация; първичната станция предава данни към вторичната, а вторичната станция само отговаря на командите от първичната

## Основни характеристики

---

### ➤ режими на предаване

- ✓ **асинхронен балансиран** ABM (*Asynchronous Balanced Mode*) – при балансирана конфигурация;  $\forall$  от комбинираните станции предават данни без да е необходимо разрешение от другата
- ✓ **с асинхронен отговор** ARM (*Asynchronous Response Mode*) – при не балансирана конфигурация; вторична станция предава данни без явно разрешение от първичната; първичната станция отговаря за линията, стартиране на трансфера, откриване на грешките и логическото прекъсване на връзката

# Структура на полетата

- ❑ HDLC използва синхронно предаване
- ❑ **Формат на рамката** – флагове, адрес, управление, контролна сума FCS и информационно поле



## ❑ Полета на флаговете

- определят границите на рамката - заглавна част (*header*) и опашка (*trailer*)
- уникална поредица от нули и единици – 01111110 – може да се използва като завършващ и като стартов флаг на следващата рамка

# Структура на полетата

- нарушаване на синхронизацията – при поява на комбинация 01111110 в полето от данни
- решение на този проблем - чрез процедурата *уплътняване* (запълване) на битовете → вмъква допълнителен нулев бит след всяко появяване на пет последователни единици в дадена рамка
- пример за уплътняване на битове

оригинални данни

1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

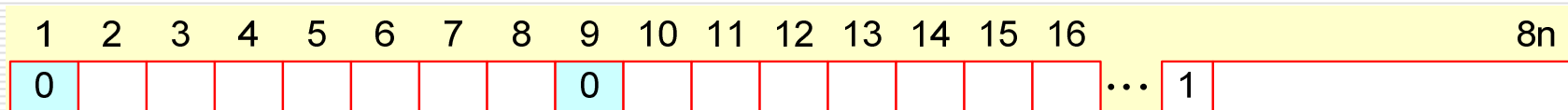
уплътняване на битовете

1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

# Структура на полетата

## □ Адресно поле

- идентифицира вторичната станция, която предава/получава дадена рамка
- не е необходимо при връзка между две точки (*point-to-point*), но се включва заради придържане към протокола
- дължина 8 бита или се използва разширен формат, при който дължината на адреса е кратна на 7 бита
- най-левият бит на всеки октет определя броя на октетите в адреса



- първичната станция използва единичен октет 11111111 → приема се за адрес на всички станции (*all-stations address*)

# Структура на полетата

## □ Управляващо поле

- HDLC има 3 типа рамки с различен формат
  - **информационни рамки** (*I-frames*) - ARQ механизъм с плъзгащ прозорец
  - **наблюдавани рамки** (*S-frames*) -ARQ механизъм без плъзгащ прозорец
  - **неизброими рамки** (*U-frames*) - функции за допълнително управление на канала първите един/два бита определят типа на рамката
- битове са организирани в под полета
  - P/F бит (*poll/final*) - в управляващите рамки P=1 изисква рамка за отговор; в рамките за отговор F=1 показва предаден отговор
  - N(S) и N(R)–изпратен (получен) последователен брой
  - S – наблюдавани функционални битове
  - M – неизброими функционални битове

# Структура на полетата

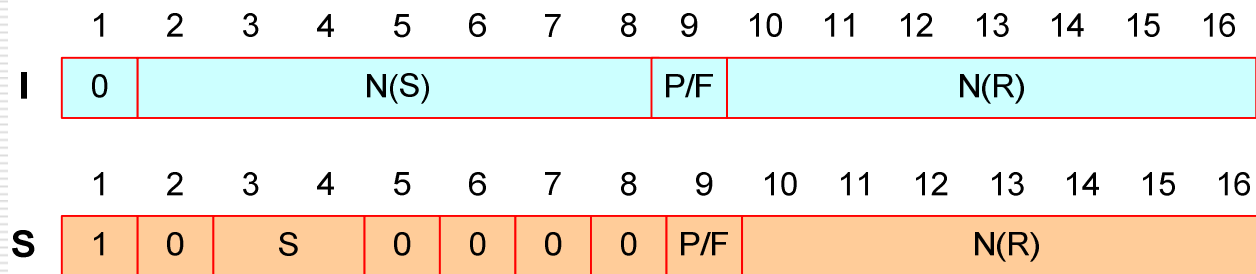
➤ 8 bits формат на управляващото поле

	1	2	3	4	5	6	7	8
<b>I</b>	0	N(S)			P/F	N(R)		
<b>S</b>	1	0	S	P/F	N(R)			
<b>U</b>	1	1	M	P/F	M			

- ✓ основното управляващо поле за S-frames и I-frames използва 3 bits последователни номера
- ✓ U-frames съдържат винаги 8 bits управляващо поле

# Структура на полетата

- 16 bits формат на управляващото поле



## ❑ Информационно поле

- представено само в I-frames и в някои U-frames
- полето съдържа битова последователност от пълен брой октети
- променлива дължина на информационното поле

## ❑ Поле на контролната сума FCS

- нормално се използва 16 bits CRC-CCITT код
- използва се и 32 bits CRC-32 код

## Фази на HDLC операцията

---

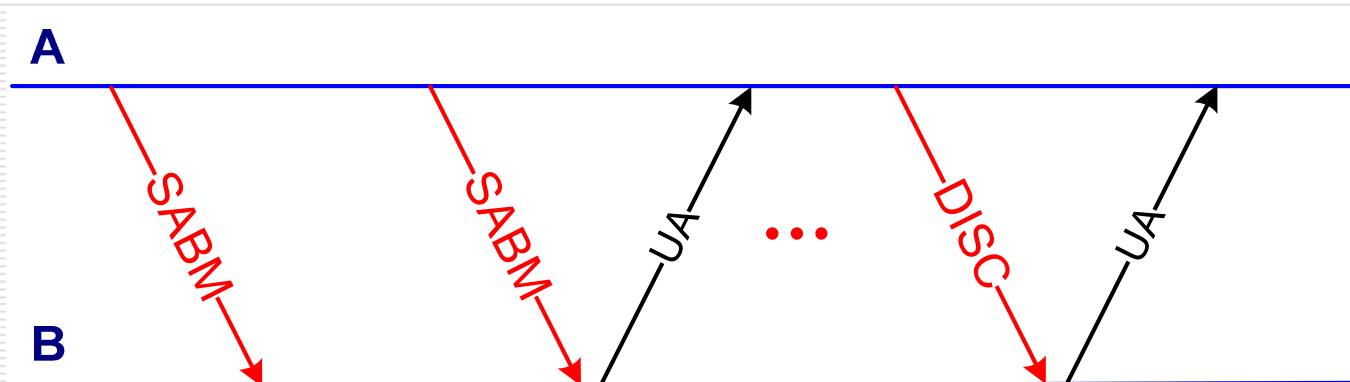
- ❑ HDLC операцията представлява обмен на S-,I- и U-frames между две станции
- ❑ Операцията се състои от 3 основни фази
  - *стартране на канала* за обмен на рамки - една от двете страни; договарят се опциите, които ще се използват
  - *обмен на данни и управляваща информация*-управление на грешката и управление на потока
  - *сигнализиране за прекратяване* на операцията - една от двете страни

## Фази на HDLC операцията

### □ Примери за операция

#### ➤ *установяване и разпадане на канал*

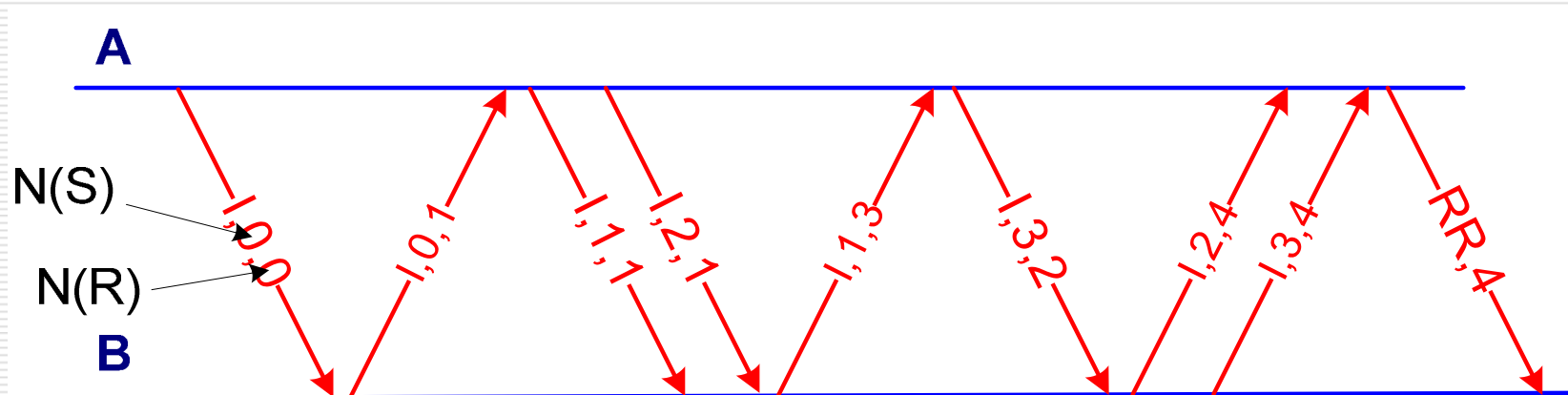
- ✓ HDLC протокола на **A** подава команда SABM към **B** и стартира таймер
- ✓ **B** връща отговор UA и установява началните стойности на променливите и броячите
- ✓ Прекъсване **A** изпраща командата DISC, а **B** отговаря с UA



## Фази на HDLC операцията

### ➤ пълен дуплекс на I рамка

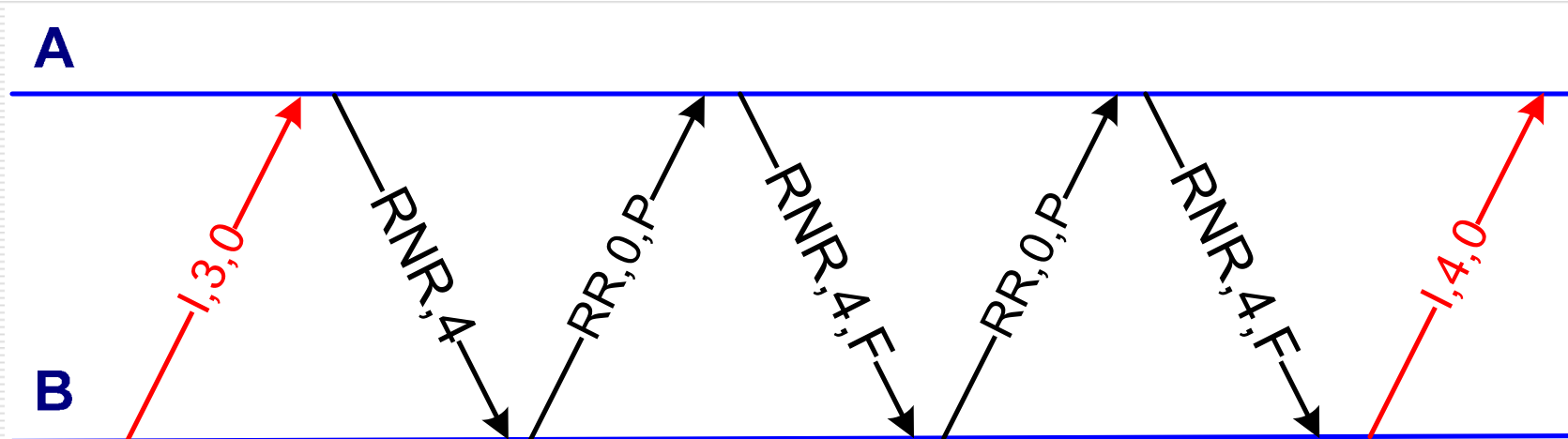
- ✓ **A** изпраща I-frames без входящи данни (I,0,0) - повтаря се последователността (I,1,1; I,2,1)
- ✓ **B** получава поредица от I-frames -в изходящата рамка се използва следващия пореден номер (I,1,3)
- ✓ обменът на данни засяга и управляващи рамки (RR4)



## Фази на HDLC операцията

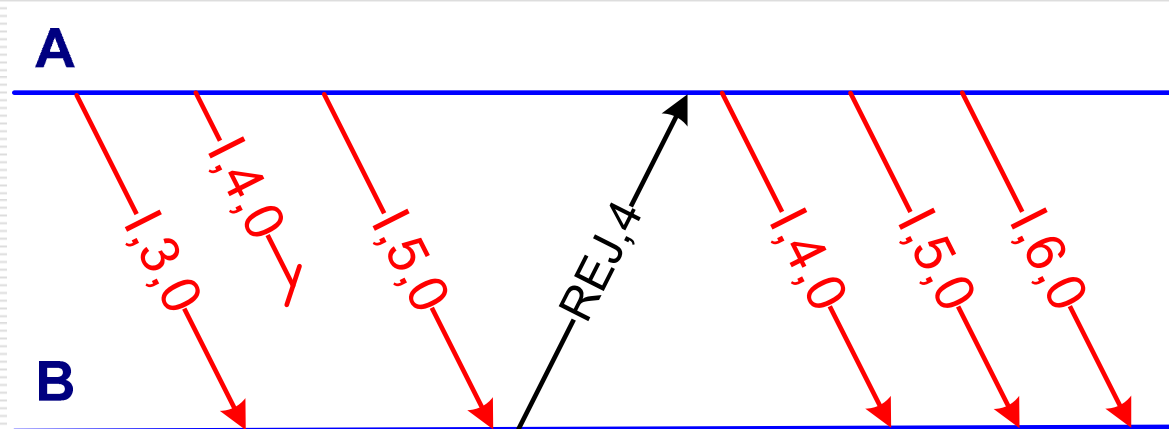
### ➤ **заето състояние**

- ✓ HDLC не може да обработи I-frames със скоростта на пристигане, т.е. **В** не е в състояние да приеме данните
- ✓ препълва се входящия буфер и се спират I-frames чрез RNR команда



## Фази на HDLC операцията

- **Възстановяване на грешка с командата REJ**



- **Възстановяване на грешка с прекъсване на времето**

