



Comunicação de Dados

Transmissão Digital

Prof. Hermes Irineu Del Monego, D.E.E.C.
hmonego@utfpr.edu.br

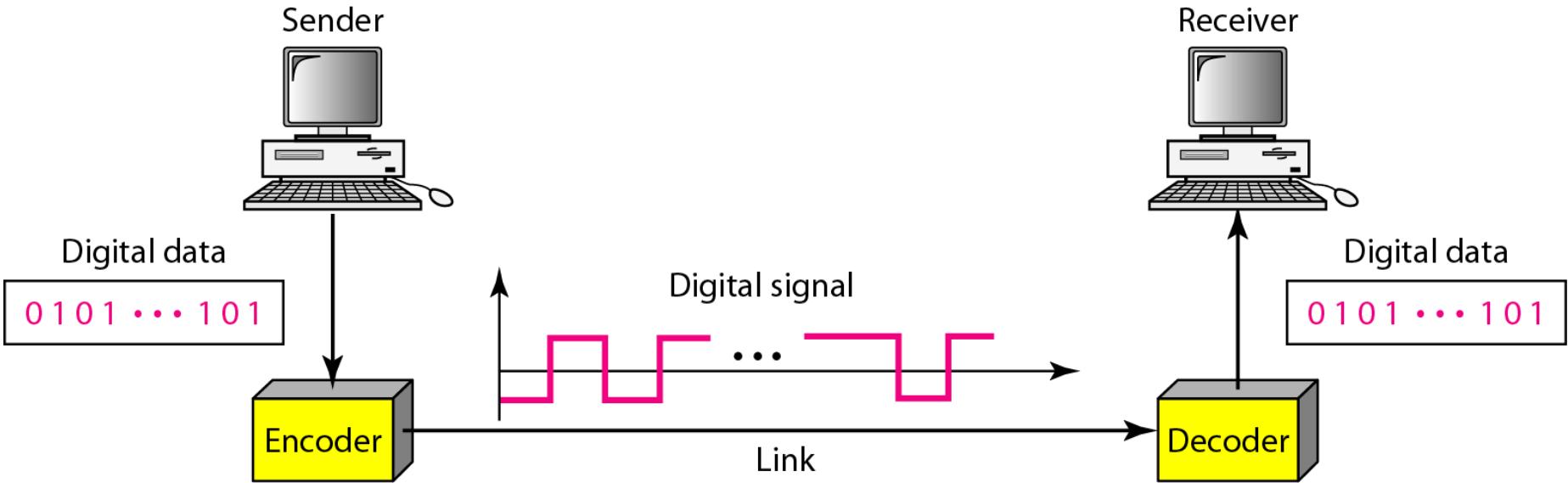
Conversão Digital/Analógica

Dissemos que os dados podem ser digitais ou analógicos. Dissemos ainda que os sinais que representam dados também podem ser digitais ou analógicos. Veremos como podemos representar dados digitais por meio de sinais digitais. A conversão envolve três técnicas: codificação de linha, codificação de blocos e mistura de sinais. A codificação de linha é sempre necessária; a codificação de blocos e a mistura de sinais não necessariamente.

Codificação e decodificação de linha

Codificação de linha é o processo de conversão de dados digitais em sinais digitais. Partimos do pressuposto de que os dados, na forma de texto, números, imagens, áudio ou vídeo, são armazenados na memória do computador como sequências de bits. A codificação de linha converte uma sequência de bits em um sinal digital. No emissor, os dados digitais são codificados em um sinal digital; no receptor, os dados digitais são recriados, reconvertendo-se o sinal digital. A Figura abaixo ilustra o processo.

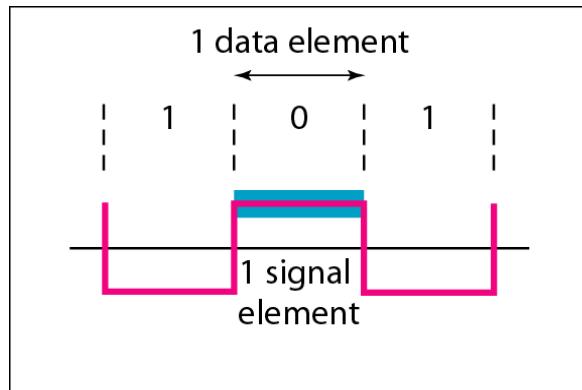
Codificação e decodificação de linha



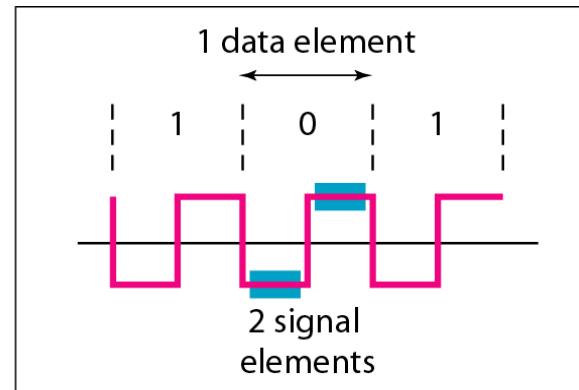
Elemento de Sinal versus Elemento de Dados

Façamos a distinção entre um elemento de dados e um elemento de sinal. Em comunicação de dados, o objetivo é transmitir elementos de dados. Um elemento de dados é a menor entidade capaz de representar uma informação: trata-se do bit. Na comunicação de dados digitais, elementos de sinal transportam elementos de dados. Um elemento de sinal é a menor unidade (em termos de tempo) de um sinal digital. Em outras palavras, os elementos de dados são aquilo que precisamos transmitir; os elementos de sinal são aquilo que podemos transmitir. Os elementos de dados são transportados; os elementos de sinal são os portadores.

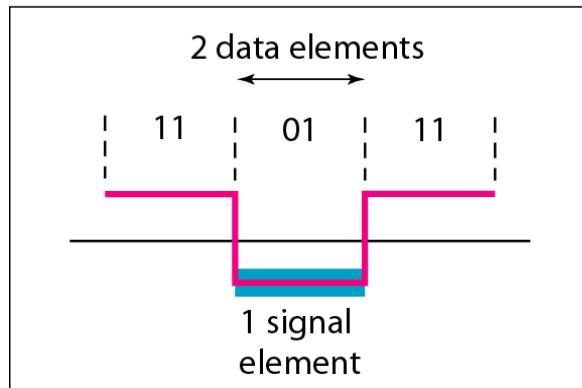
Elemento de Sinal versus Elemento de Dados



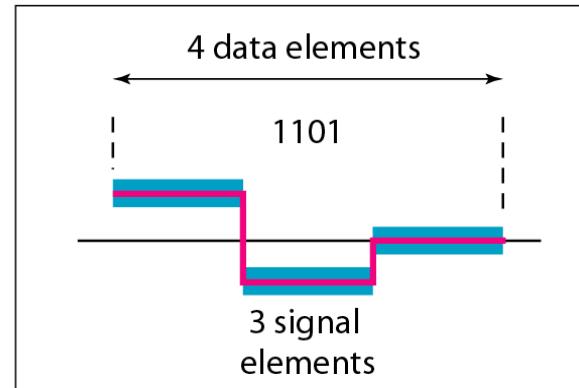
a. One data element per one signal element ($r = 1$)



b. One data element per two signal elements ($r = \frac{1}{2}$)



c. Two data elements per one signal element ($r = 2$)



d. Four data elements per three signal elements ($r = \frac{4}{3}$)

Elemento de Sinal versus Elemento de Dados

- a) da figura, um elemento de dados é transportado por um elemento de sinal ($r = 1$)
- b) da figura, precisamos de dois elementos de sinal (duas transições) para transportar Cada elemento de dados ($r = 1/2$). Veremos mais adiante que um elemento de sinal extra é necessário para garantir sincronização
- c) da figura, um elemento de sinal transporta dois elementos de dados ($r = 2$)
- d), um grupo de 4 bits é transportado por um grupo de três elementos de sinal ($r = 4/3$). Para cada método de codificação de linha que discutirmos, forneceremos o valor de r .

Taxa de Dados versus Taxa de Sinal

A taxa de dados define o número de elementos de dados (bits) enviados na forma de 1. A unidade é bits por segundo (*bps*). A taxa de sinal é o número de elementos de sinal enviados na forma de 1. A unidade é *baud*. Existem várias terminologias comuns usadas no jargão técnico. A taxa de dados é algumas vezes chamada taxa de bits; a taxa de sinal é muitas vezes denominada taxa de pulsos, taxa de modulação ou taxa de transmissão.

Podemos formular a relação entre taxa de dados e taxa de sinal como: $S=c \times N \times 1/r$ *baud*, onde N é a taxa de dados (*bps*); c é o fator de caso, que varia para cada caso; S é o número de elementos de sinal (taxa de transmissão) e r é o número de elementos de dados.

Exemplo 1

Um sinal transporta dados nos quais um elemento de dados é codificado como um elemento de sinal ($r = 1$). Se a taxa de bits for de 100 kbps , qual é o valor médio da taxa de transmissão se c estiver entre 0 e 1?

Exemple 1

Um sinal transporta dados nos quais um elemento de dados é codificado como um elemento de sinal ($r = 1$). Se a taxa de bits for de 100 kbps , qual é o valor médio da taxa de transmissão se c estiver entre 0 e 1?

Solução:

Supondo que o valor médio de c seja $1/2$. A taxa de transmissão é então

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100,000 \times \frac{1}{1} = 50,000 = 50 \text{ baud}$$

Taxa de Dados versus Taxa de Sinal

Embora a largura de banda real de um sinal digital seja infinita, a largura de banda efetiva é finita. Quando falarmos de largura de banda de um sinal digital, precisamos nos lembrar que estamos falando sobre essa largura de banda efetiva. podemos dizer que a largura de banda (intervalo de frequências) é proporcional à taxa de sinal (taxa de transmissão).

A largura de banda mínima pode ser dada por $B_{\min} = c \times N \times 1/r$
Podemos calcular a taxa de dados máxima se a largura de banda do canal for dada. $N_{\max} = 1/c \times B \times r$

Exemplo 2

A taxa de dados máxima de um canal é $N_{máx} = 2 \times B \times \log_2 L$ (definido pelo teorema de Nyquist). Isso está de acordo com a fórmula para $N_{máx}$?

Exemplo 2

A taxa de dados máxima de um canal é $N_{\max} = 2 \times B \times \log_2 L$ (definido pelo teorema de Nyquist). Isso está de acordo com a fórmula para N_{\max} ?

Solução:

Um sinal com L níveis pode, na verdade, transportar $\log_2 L$ bits por nível. Se cada nível corresponder a um elemento de sinal e supormos o caso médio ($c = 1/2$), teremos então:

$$N_{\max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L$$

Falta de sincronismo

Afastamento em Relação à Referência Inicial: Ao decodificar um sinal digital, o receptor calcula a media efetiva de potência do sinal recebido. Essa média é chamada referência inicial. A potência do sinal de entrada é comparada em relação a essa referência para determinar o valor do elemento de dados. Uma longa *string* de 0s ou 1s pode provocar uma mudança (desvio) na referência (afastamento em relação à referência inicial) tornando difícil para o receptor decodificar corretamente o sinal. Um bom método de codificação de linha precisa evitar o afastamento em relação à referencia inicial.

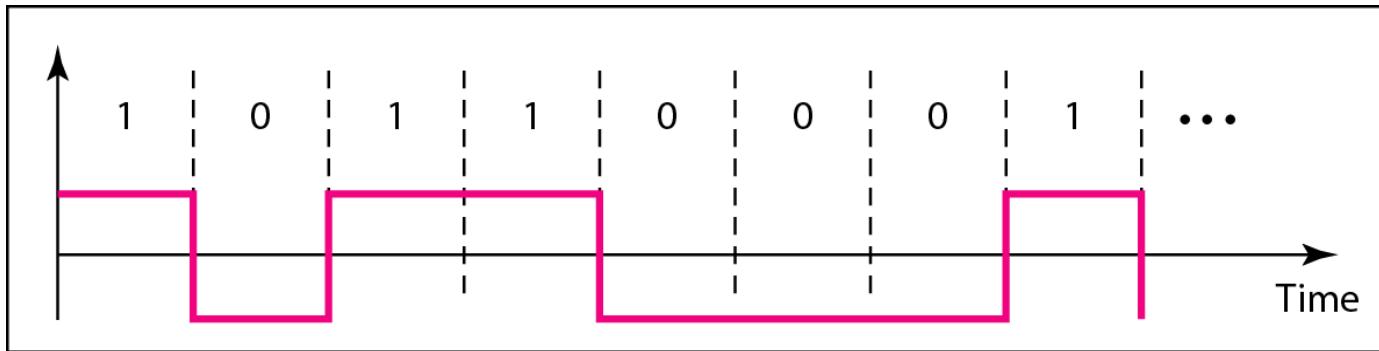
Falta de sincronismo

Componentes DC: Quando o nível de tensão de um sinal digital for constante por certo período, o espectro cria frequências muito baixas (resultados da análise de Fourier). Essas frequências em torno de zero, chamadas componentes DC, constituem um problema para um sistema que não seja capaz de deixar passar baixas frequências ou para um sistema que use acoplamento elétrico (por meio de um transformador). Por exemplo, uma linha telefônica não é capaz de deixar passar frequências abaixo de 200 Hz. Da mesma forma, um link de longa distância poderá usar um ou mais transformadores para isolar eletricamente partes distintas de uma linha. Para tais sistemas, precisamos de uma estratégia para eliminar os componentes DC.

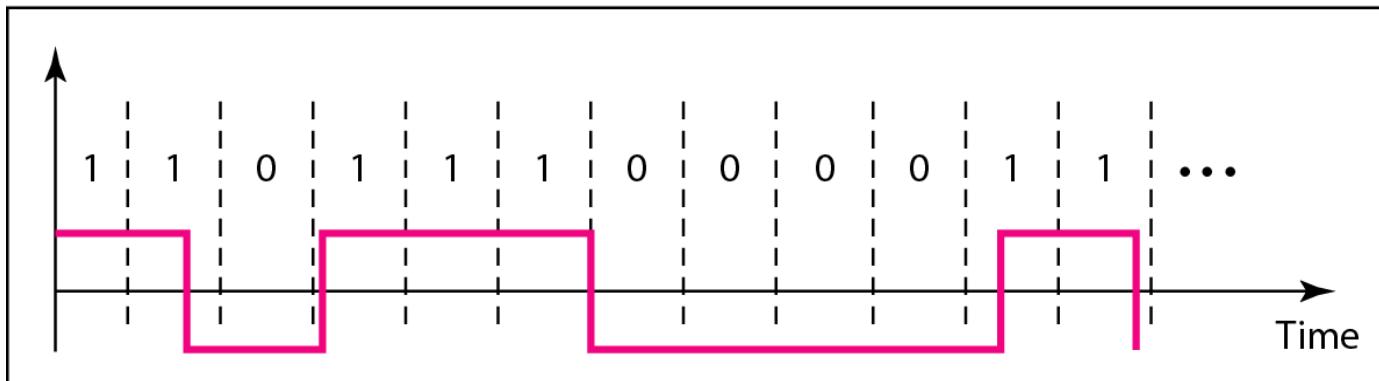
Falta de sincronismo

Auto sincronização: Para interpretar corretamente os sinais recebidos do emissor, os intervalos de bits do receptor devem corresponder exatamente aos intervalos de bits do emissor. Se o *clock* do receptor for mais rápido ou mais lento, os intervalos de bits recebidos não coincidirão e pode ser que o receptor interprete os sinais de forma incorreta. A Figura baixo mostra uma situação na qual o receptor tem uma duração de bits menor. O emissor envia 10110001, ao passo que o receptor recebe 110111000011. Um sinal digital auto sincronizado inclui informações de sincronismo nos dados transmitidos. Isso pode ser obtido se existirem transições no sinal que alertem o receptor sobre o início, meio ou fim de um pulso.

Efeito da Falta de sincronismo



a. Sent



b. Received

Example 3

Em uma transmissão digital, o *clock* do receptor é de 0,1% mais rápido que o do emissor. Quantos bits extras por segundo o receptor recebe se a taxa de dados for de 1 *kbps*? E no caso de 1 Mbps?

Example 3

Em uma transmissão digital, o *clock* do receptor é de 0,1% mais rápido que o do emissor. Quantos bits extras por segundo o receptor recebe se a taxa de dados for de 1 *kbps*? E no caso de 1 Mbps?

Solução:

A 1 kbps, o receptor recebe 1.001 bps em vez de 1.000 bps.

1000 bits sent	1001 bits received	1 extra bps
----------------	--------------------	-------------

A 1 Mbps o receptor recebe 1.001.000 bps em vez de 1.000.000 bps.

1,000,000 bits sent	1,001,000 bits received	1000 extra bps
---------------------	-------------------------	----------------

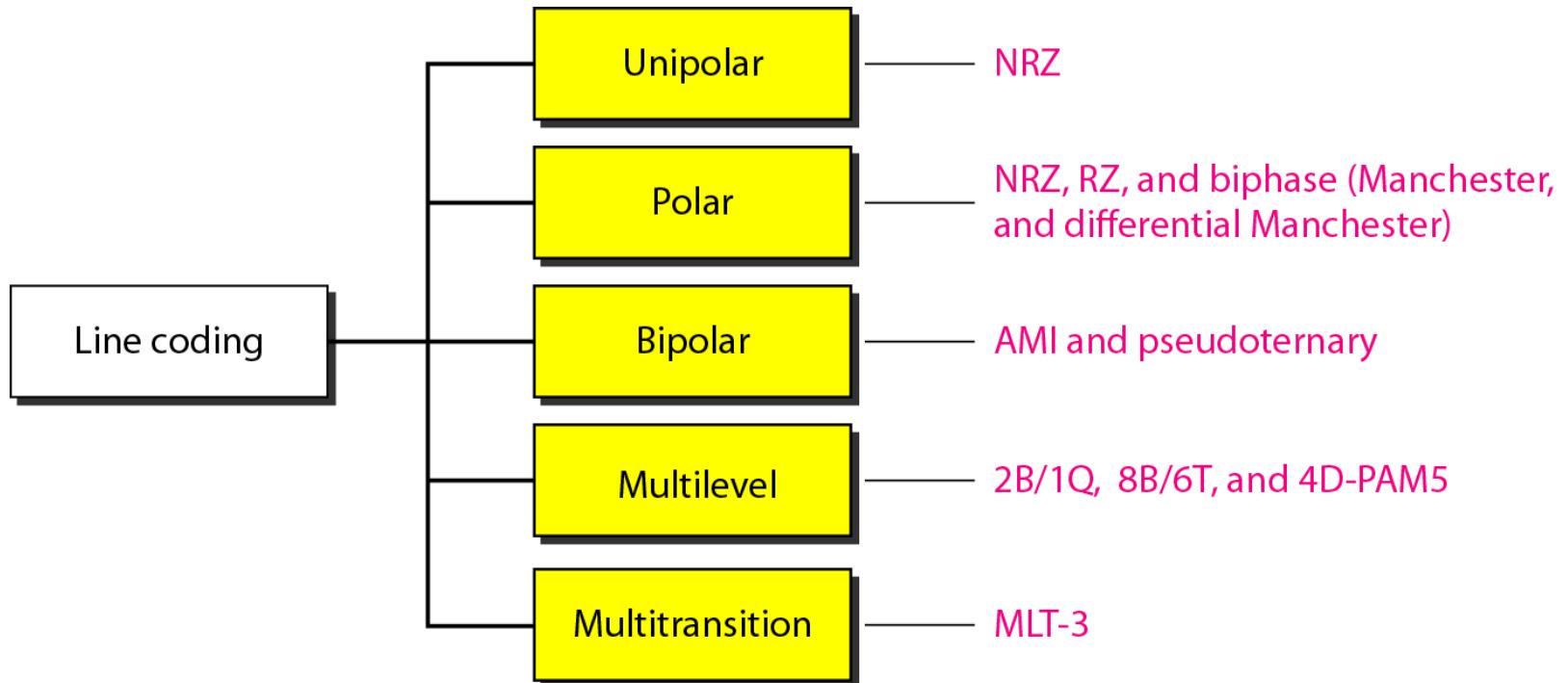
Métodos de Codificação de Linha

Detecção de Erros Embutidos: É desejável possuir recursos de detecção de erros embutidos no código gerado para detectar parte ou todos os erros ocorridos durante uma transmissão.

Imunidade a Ruído e Interferência: Outra característica de código desejável é um código que seja imune a ruídos e outras interferências. Alguns métodos de codificação que iremos tratar apresentam essa capacidade.

Complexidade: Um método complexo é mais dispendioso de implementar que um simples. Por exemplo, um método que use quatro níveis de sinal é mais difícil de interpretar que um com apenas dois níveis.

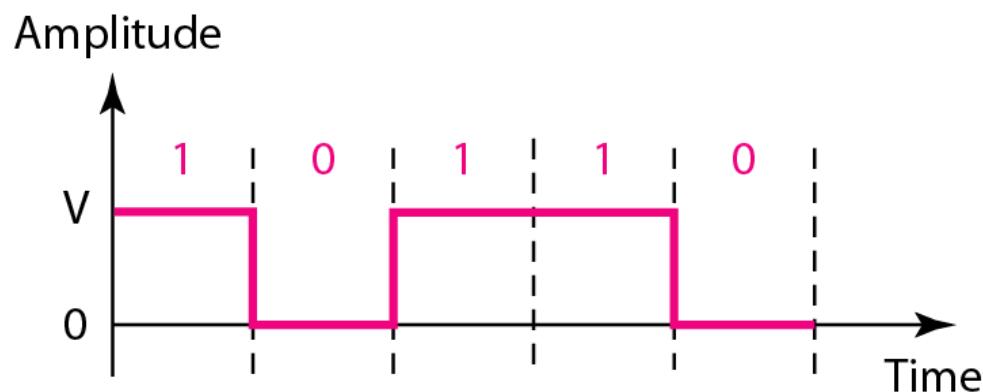
Métodos de Codificação de Linha



Esquema NRZ Unipolar

NRZ (Non-Return-to-Zero, ou seja, Sem Retorno a Zero)

Tradicionalmente, um método unipolar foi desenvolvido como um método NRZ (sem retorno a zero) no qual a tensão positiva define o bit 1 e a tensão zero define o bit 0. Ele é chamado NRZ porque o sinal não retorna a zero no meio do bit. A Figura abaixo mostra um método NRZ unipolar.



$$\frac{1}{2}V^2 + \frac{1}{2}(0)^2 = \frac{1}{2}V^2$$

Normalized power

Esquemas Polares

Em métodos polares, as tensões se encontram em ambos os lados do eixo. Por exemplo, o nível de tensão para 0 pode ser positivo e o nível de tensão para 1 pode ser negativo.

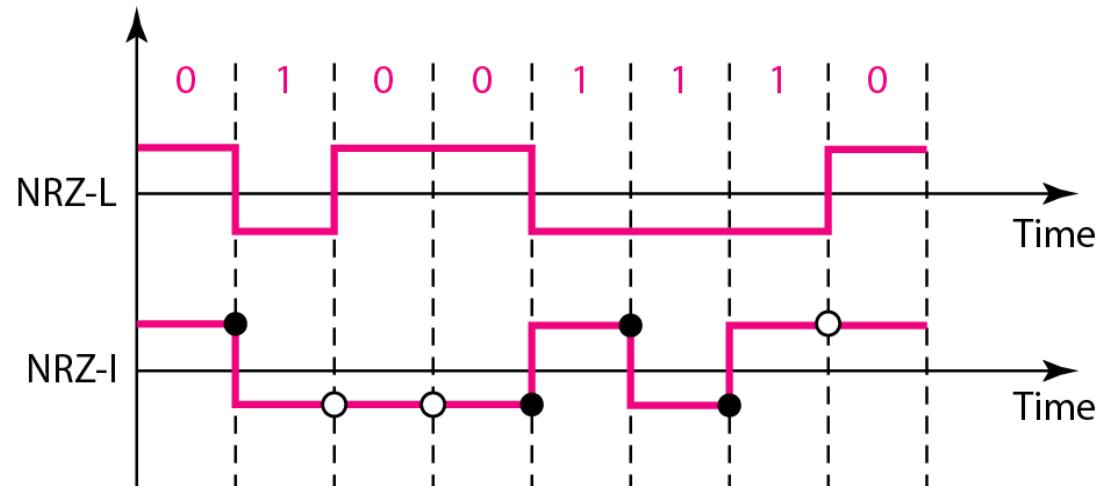
Esquema NRZ Unipolar

NRZ (Sem Retorno a Zero) Na codificação NRZ polar, usamos dois níveis de amplitude de tensão. Podemos ter duas versões de NRZ polar: NRZ-L e NRZ-I, conforme pode ser visto na Figura abaixo. A figura também mostra o valor de r , a taxa de transmissão média, e a largura de banda.

Na variante, **NRZ-L (NRZ-Level)**, o nível da tensão determina o valor do bit.

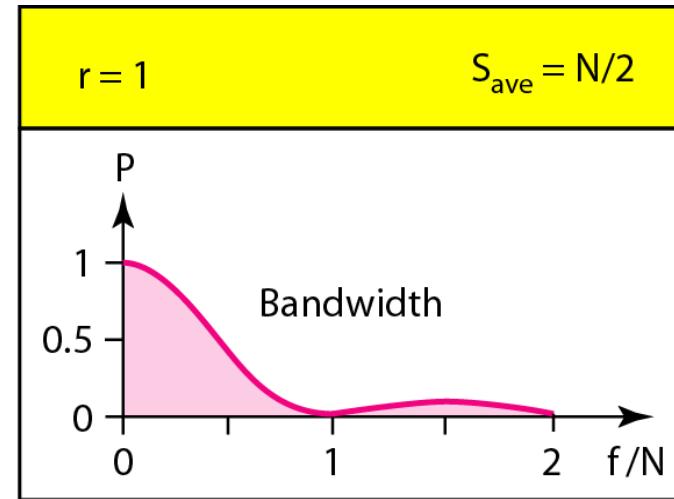
Na variante, **NRZ-I (NRZ-Invert)**, a mudança ou falta de mudança no nível da tensão determina o valor do bit a ser transmitido. Esse método não modifica o sinal de saída quando envia bit “0”, e inverte o sinal do estado anterior quando enviando bit “1”.

Esquema NRZ-L e NRZ-I



○ No inversion: Next bit is 0

● Inversion: Next bit is 1



Tanto o método NRZ-L quanto o NRZ-I apresentam a taxa média de sinal igual a $N/2$ Bd.

Tanto o método NRZ-L quanto o NRZ-I apresentam problemas de componentes DC.

Example 3

Um sistema usa o método NRZ-I para transferir dados a 10 Mbps. Qual é a taxa média de sinal e a largura de banda mínima?

Example 4

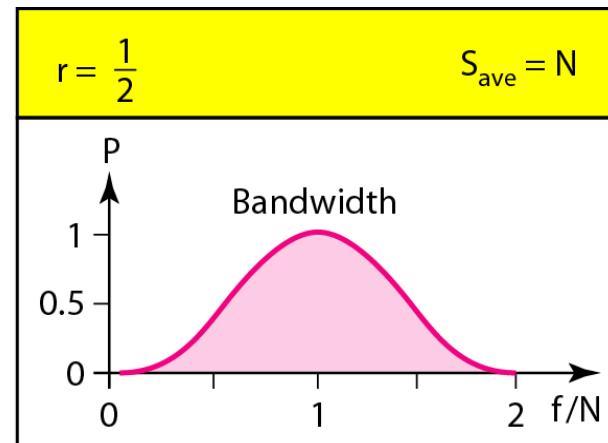
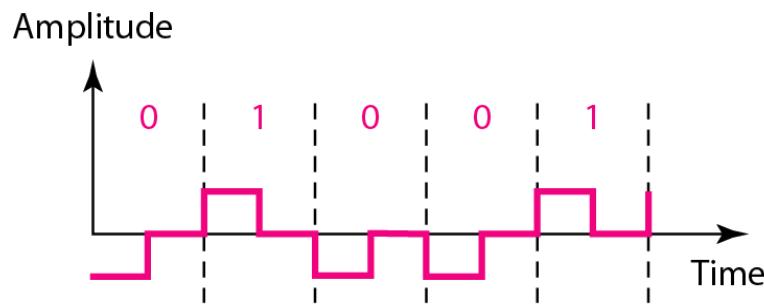
Um sistema usa o método NRZ-I para transferir dados a 10 Mbps. Qual é a taxa média de sinal e a largura de banda mínima?

Solução

A taxa média de sinal é $S = N/2 = 500$ kbauds. A largura de banda mínima para essa taxa de transmissão média é $B_{min} = S = 500$ kHz.

Esquema Polar ZR

O principal problema na codificação NRZ ocorre quando os clocks do emissor e do receptor não estão sincronizados. O receptor não sabe quando um bit terminou e o próximo bit está começando. Uma solução seria o método RZ (*return-to-zero*, ou seja, retorno a zero) que usa três valores: positivo, negativo e zero. No método RZ, o sinal muda não entre bits, mas sim durante o bit.



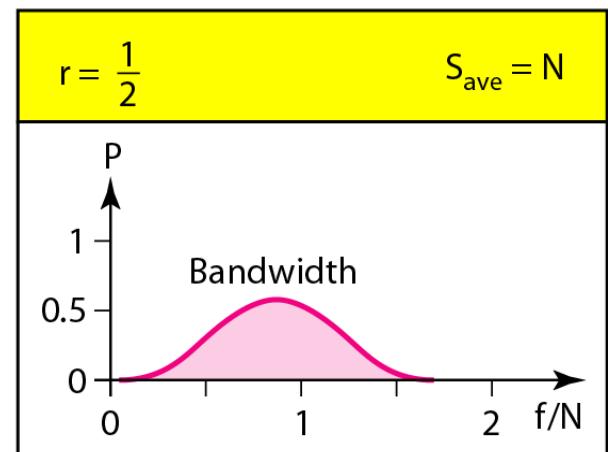
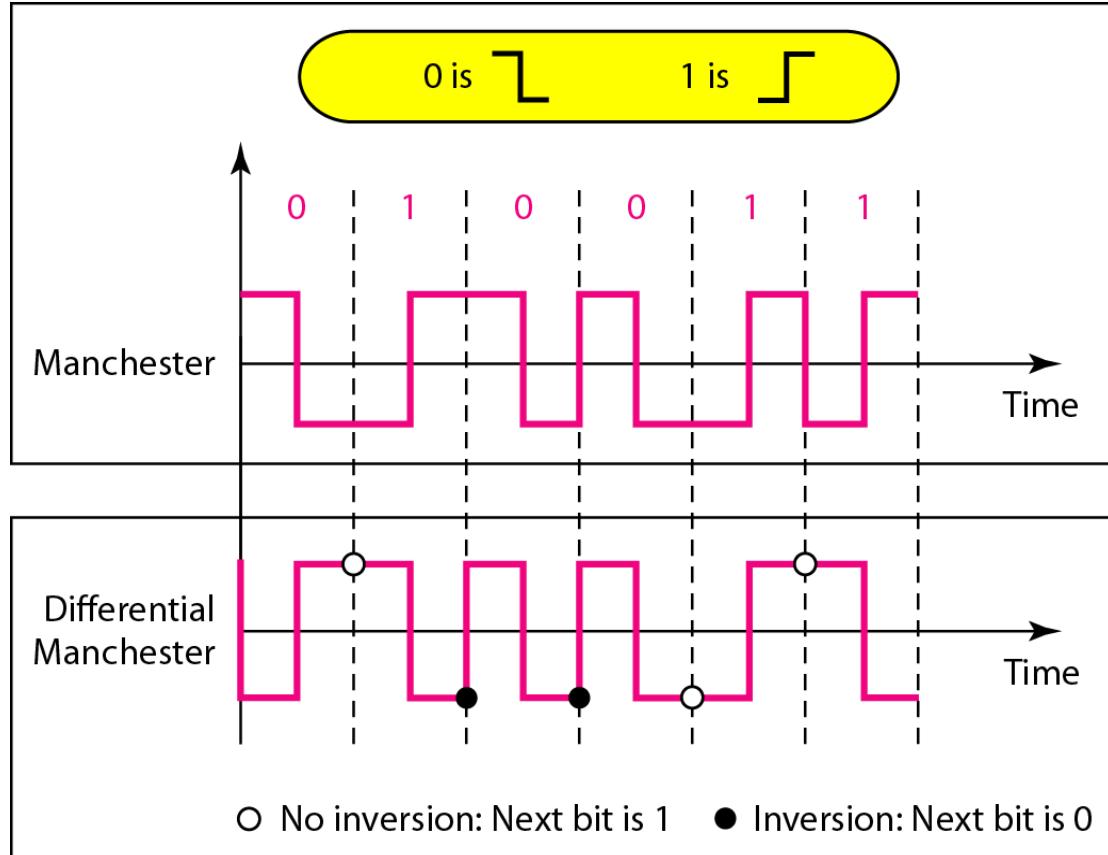
Bifásico: Manchester

O Manchester reúne os conceitos do método RZ (transição no meio do bit) e do método NRZ-L são combinados no método Manchester. Na codificação Manchester, a duração do bit é dividida em duas metades. A Tensão permanece em um nível durante a primeira metade e se desloca para o outro nível na segunda metade. A transição no meio do bit fornece sincronismo.

Bifásico: Manchester Diferencial

O método Manchester diferencial combina os conceitos dos métodos RZ e NRZ-I. Existe sempre uma transição no meio do bit, mas os valores são determinados no inicio dele. Se o próximo bit for 0, ocorre uma transição; caso o bit seguinte seja 1, não ocorre nenhuma transição. A Figura abaixo mostra os métodos de codificação Manchester e Manchester diferencial.

Manchester e Manchester diferencial



Nas codificações Manchester e Manchester diferencial, a transição no meio do bit é usada para sincroniza

A largura de banda mínima requerida para os métodos Manchester e Manchester diferencial é o dobro do NRZ.

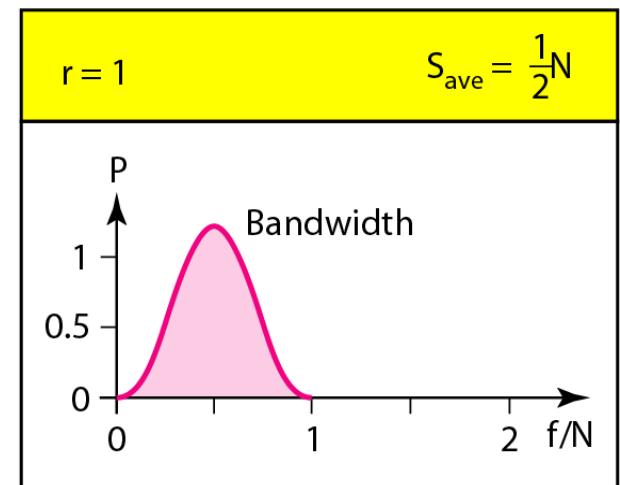
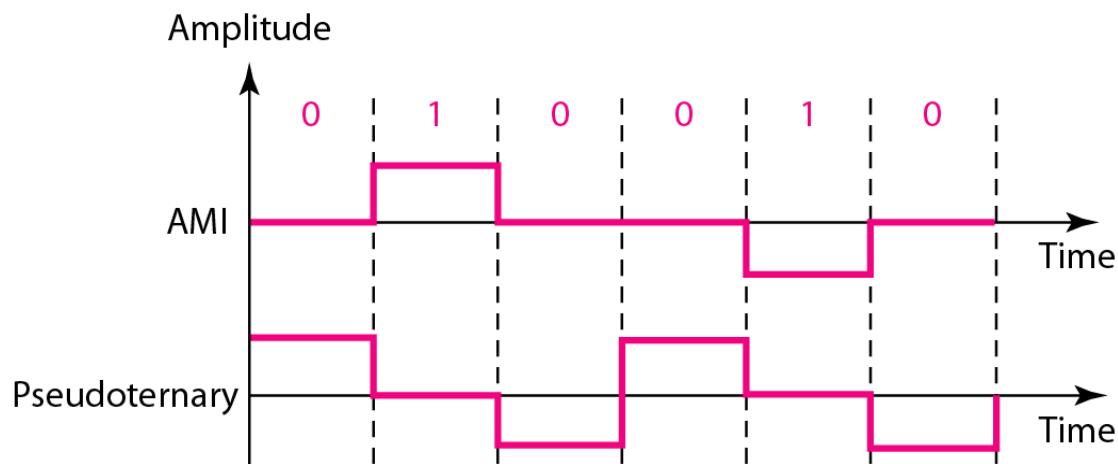
Esquemas Bipolares

Na codificação **bipolar** (algumas vezes denominada binária multinível), existem três níveis de tensão: positivo, negativo e zero. O nível de tensão para um elemento de dados se encontra em zero, ao passo que o nível de tensão para o outro elemento fica alternando entre valores positivos e negativos.

AMI e Pseudoternária

A Figura mostra duas variações de codificação bipolar: AMI e pseudoternária. Um método de codificação bipolar comum é o chamado AMI — inversão de marca alternada (*alternate mark inversion*). No termo inversão de marca alternada, a palavra marca provem da telegrafia e significa 1. Portanto, AMI significa inversão de 1 alternado. Uma tensão neutra zero representa o 0 binário. Os 1s binários são representados alternando-se tensões positivas e negativas. Uma variante da codificação AMI é denominada pseudoternária, na qual o bit 1 é codificado como uma tensão zero e o bit 0 é codificado como tensões alternadas positiva e negativa.

AMI e Pseudoternária



Esquemas Multinível

O desejo de aumentar a velocidade de dados ou diminuir a largura de banda necessária resultou na criação de muitos métodos alternativos. O objetivo é aumentar o numero de bits por *baud* codificando-se um padrão de m elementos de dados em um padrão de n elementos de sinal.

Em métodos $mBnL$, um padrão de m elementos de dados é codificado como um padrão de n elementos de sinal, em que $2^m \leq L^n$.

Esquema 2B1Q

O método quaternário (2B1Q), usa padrões de dados de tamanho 2 e codifica os padrões de 2 bits como um elemento de sinal pertencente a um sinal de quatro níveis. Nesse tipo de codificação, temos: m=2, n=1 e L=4 (quaternário). A Figura abaixo mostra um exemplo de um sinal 2B1Q.

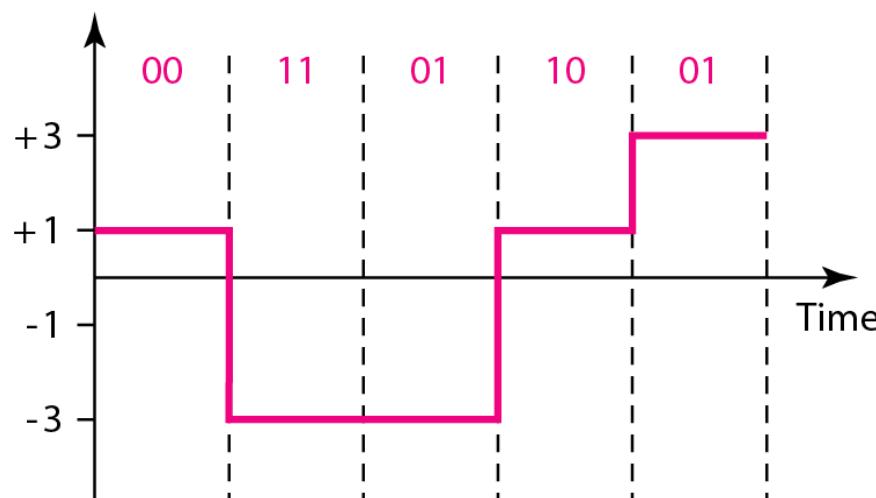
A taxa media de sinal de 2B1Q é $S=N/4$. Isso significa que, usando 2B1Q, podemos enviar dados duas vezes mais rápido que utilizando NRZ-L. Entretanto, o método 2B1Q usa quatro níveis de sinal diferentes, o que significa que o receptor tem de decodificar quatro limiares diferentes. A redução na largura de banda tem seu preço. Nesse método, não existe nenhum padrão de sinal redundante, pois $2^2 = 4$.

Esquema 2B1Q

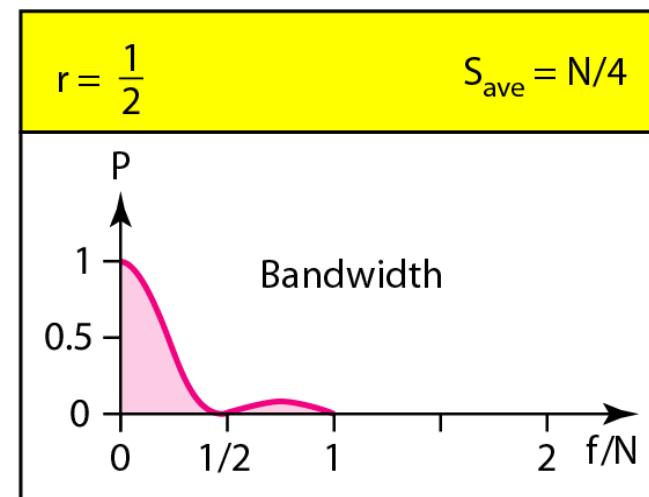
Previous level:
positive Previous level:
negative

Next bits	Next level	Next level
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

Transition table



Assuming positive original level



Esquemas 8B6T

Método interessante é o oito binário, seis ternário (8B6T). Esse código é usado em cabos 100BASE-4T. Nesse tipo de método, podemos ter $2^8=256$ padrões de dados diferentes e $3^6=478$ padrões de sinal diferentes.

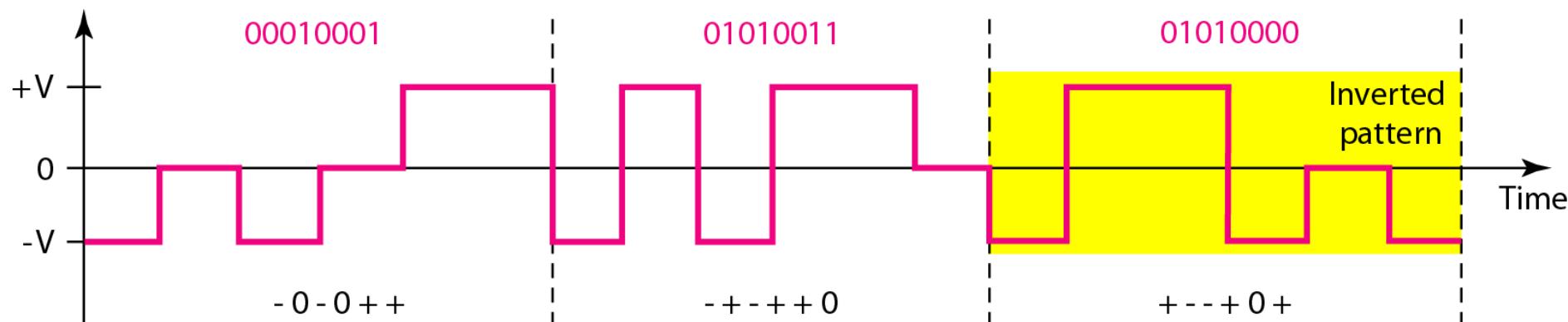
Existem $478-256=222$ elementos de sinal redundantes que fornecem sincronismo e detecção de erros. Parte da redundância também é utilizada para fornecer equilíbrio DC. Cada padrão de sinal tem peso 0 ou valores +1 DC. Isso significa que não existe nenhum padrão com peso -1.

Para tornar todo o fluxo equilibrado em termos DC, o emissor controla o peso. Se dois grupos de peso 1 forem encontrados, o primeiro será enviado como ele é, enquanto o próximo será totalmente invertido para dar peso -1.

Esquemas 8B6T

A Figura abaixo mostra um exemplo de três padrões de dados codificados como padrões de três sinais. Os três níveis possíveis de sinal são representados como $-$, 0 e $+$. O primeiro padrão de 8 bits 00010001 é codificado como o padrão de sinal $-0-0++$ com peso 0 ; o segundo padrão de 8 bits 01010011 é codificado como $-+-+-+0$ e peso $+1$. O terceiro padrão de bits deve ser codificado como $+-+-+0+$ com peso $+1$. Para criar equilíbrio DC, o emissor inverte o sinal real. O receptor pode reconhecer facilmente que se trata de um padrão invertido, pois o peso é -1 . O padrão é invertido antes da decodificação.

Esquemas 8B6T



A taxa média de sinal do método é, teoricamente, $S_{\text{médio}} = 1/2 \times N \times 6/8$; na prática, a largura de banda mínima é muito próxima de $6 \times N/8$.

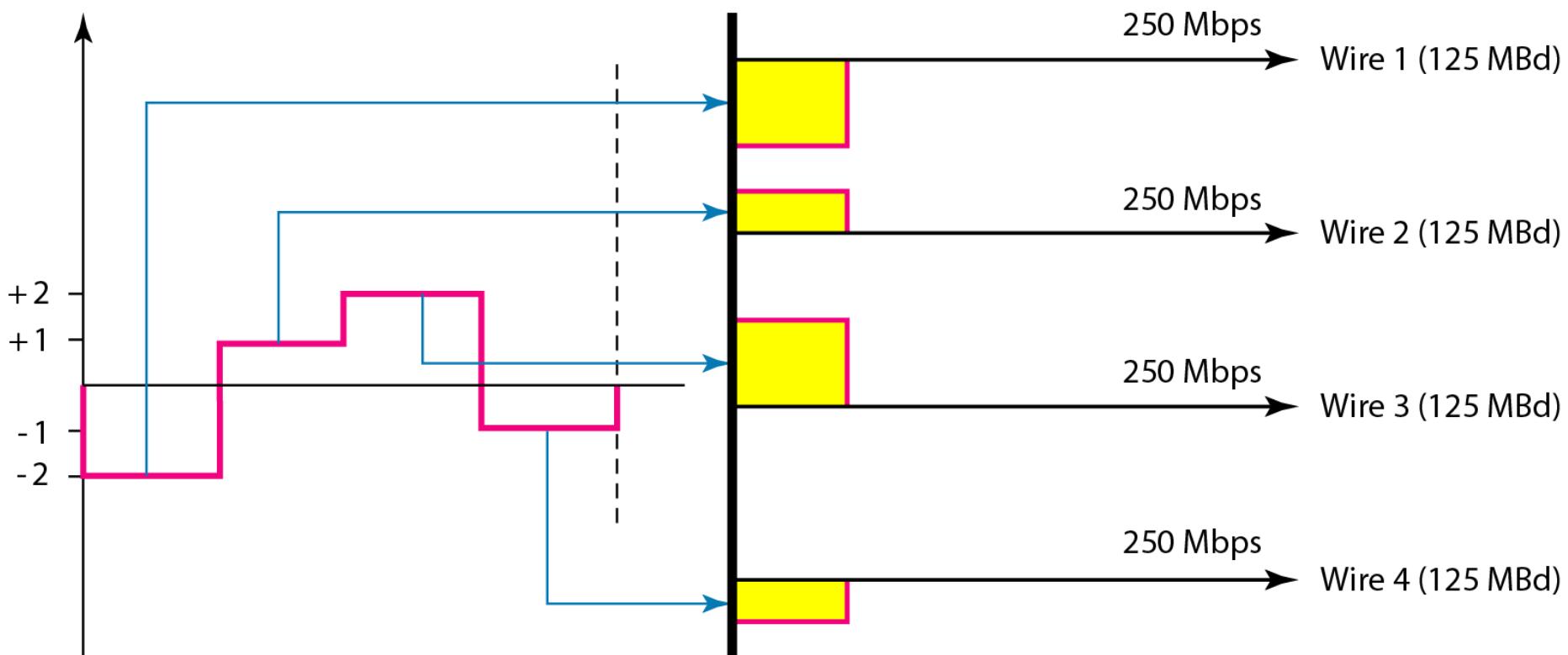
Esquemas 4D-PAM5

Modulação de amplitude de pulso com cinco níveis e quatro dimensões (4D-PAM5). O 4D significa que os dados são enviados através de quatro fios ao mesmo tempo. Ele usa cinco níveis de tensão como -2 , -1 , 0 , 1 e 2 . Entretanto, um nível, o nível 0 , é usado apenas para detecção antecipada de erros. Se assumirmos que o código é unidimensional, os quatro níveis criam algo similar ao método 8B4Q. Em outras palavras, uma palavra de 8 bits é convertida em um elemento de sinal de quatro níveis diferentes. A pior taxa de sinal para essa versão imaginaria de uma dimensão é $N \times 4/8$ ou $N/2$.

Esquemas 4D-PAM5

00011110

1 Gbps

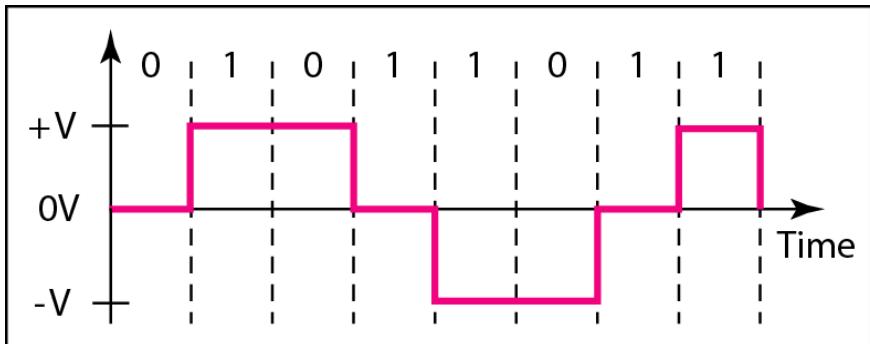


Transmissão Multilinha MLT-3

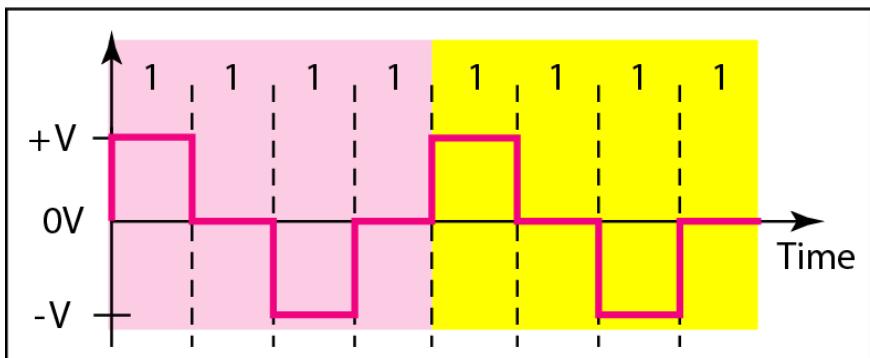
Os métodos NRZ-I e Manchester diferencial são classificados como codificação diferencial, mas usam duas regras de transição para codificar dados binários (nenhuma inversão, inversão). Se tivermos um sinal com mais de dois níveis, podemos elaborar um método de codificação diferencial com mais de duas regras de transição. O MLT-3 é um deles. O método MLT-3 (transmissão multilinha de três níveis) usa três níveis ($+V$, 0 e $-V$) e três regras de transição para mudar de nível.

Se o próximo bit for 0 , sem transição. Se o próximo bit for 1 e o nível atual 1 , o nível seguinte será 0 . Se o próximo bit for 1 e o nível atual for 0 , o nível seguinte será o oposto do ultimo nível não-zero.

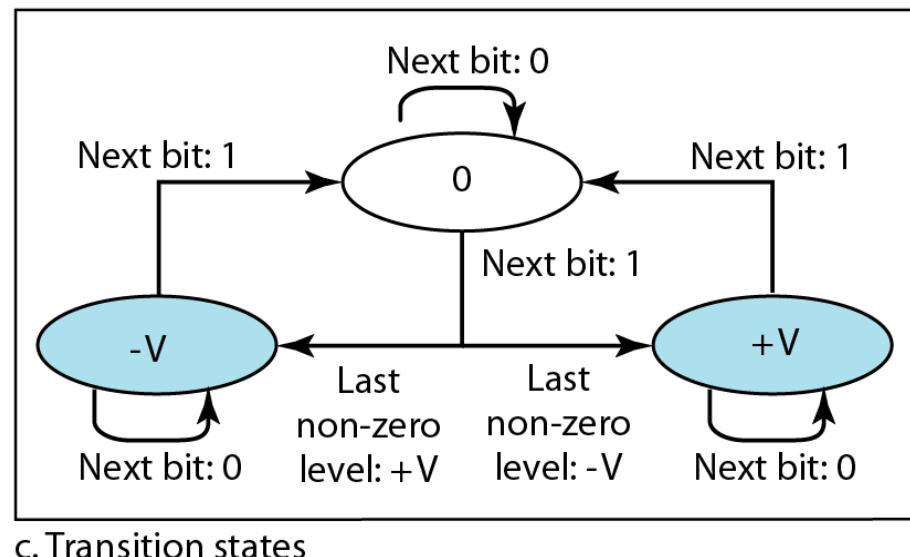
Transmissão Multilinha MLT-3



a. Typical case



b. Worse case



c. Transition states

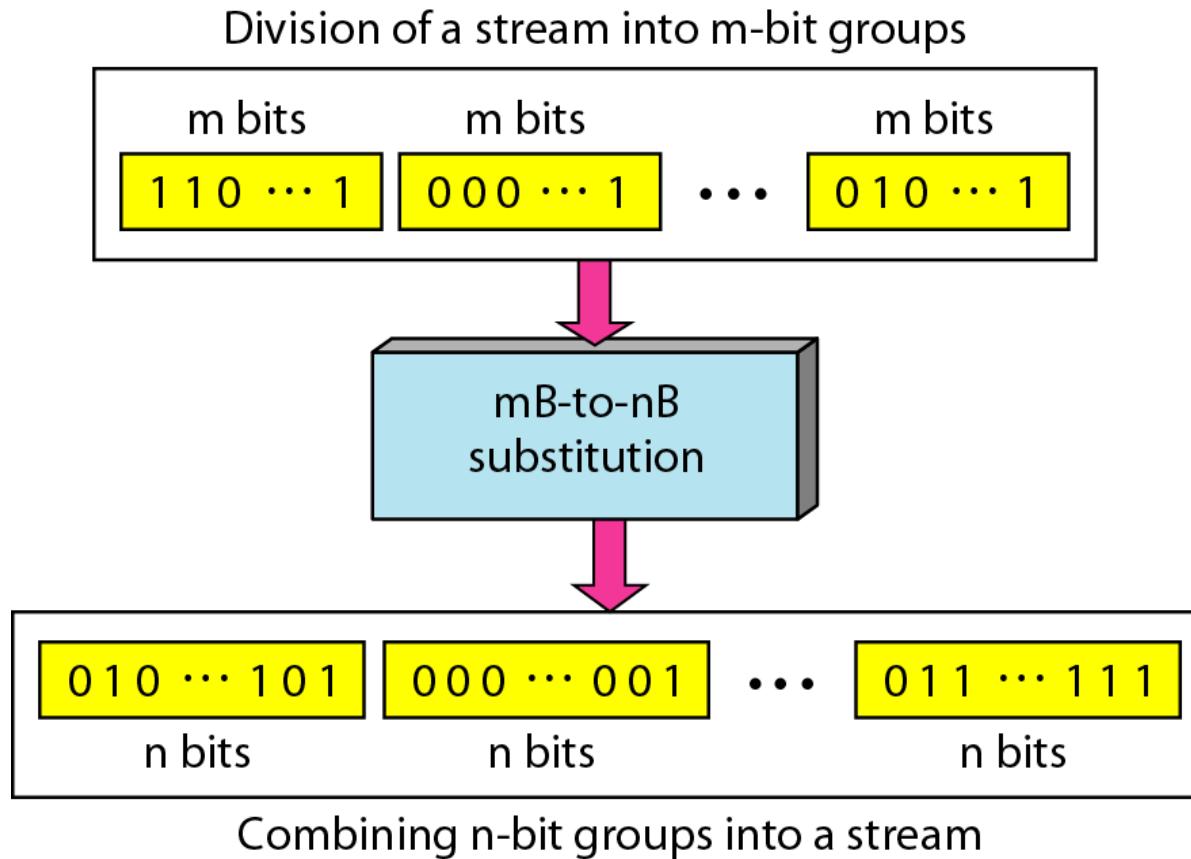
Resumo de Métodos de Codificação

<i>Category</i>	<i>Scheme</i>	<i>Bandwidth (average)</i>	<i>Characteristics</i>
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

Codificação de Blocos

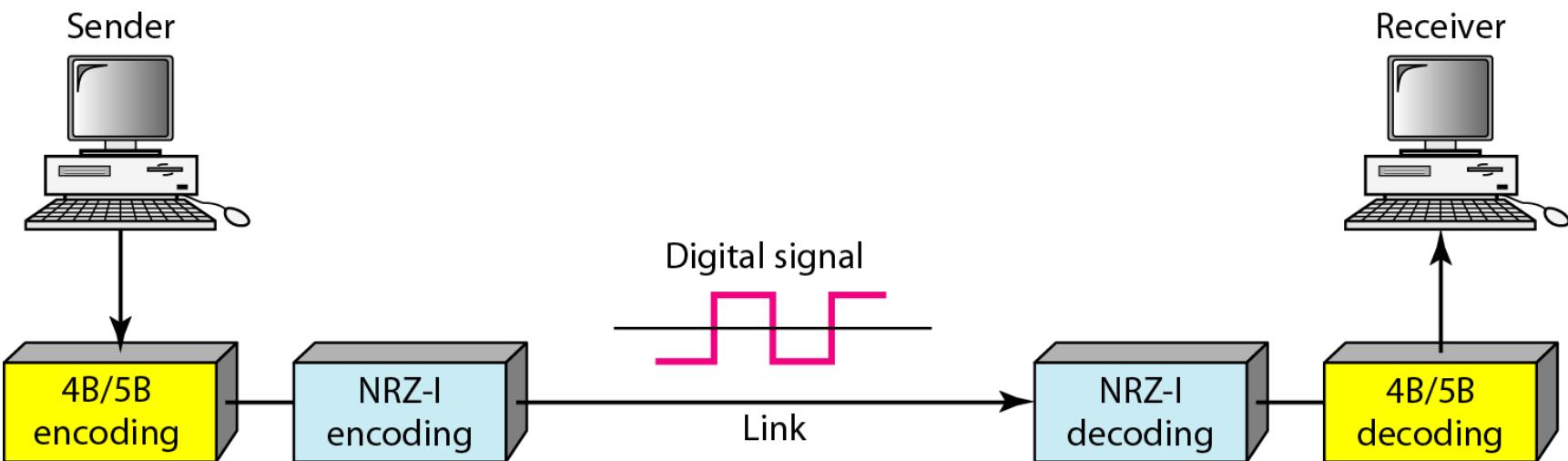
Precisamos de redundância para garantir sincronismo e para fornecer algum tipo de detecção de erros inerentes. A codificação de blocos pode fornecer essa redundância e melhorar o desempenho da codificação de linha. Em geral, a codificação de blocos muda um bloco de m bits em um bloco de n bits, em que n é maior que m . A codificação de blocos é conhecida como técnica de codificação mB/nB .

Codificação de Blocos



4B/5B com NRZ-I

O método de codificação quatro binário, cinco binário (4B/5B) foi desenvolvido para ser utilizado em combinação com o NRZ-I. O NRZ-I tem excelente taxa de sinal mas apresenta problemas de sincronização. Uma longa sequencia de 0s pode fazer que o *clock* do receptor perca o sincronismo.



4B/5B com NRZ-I

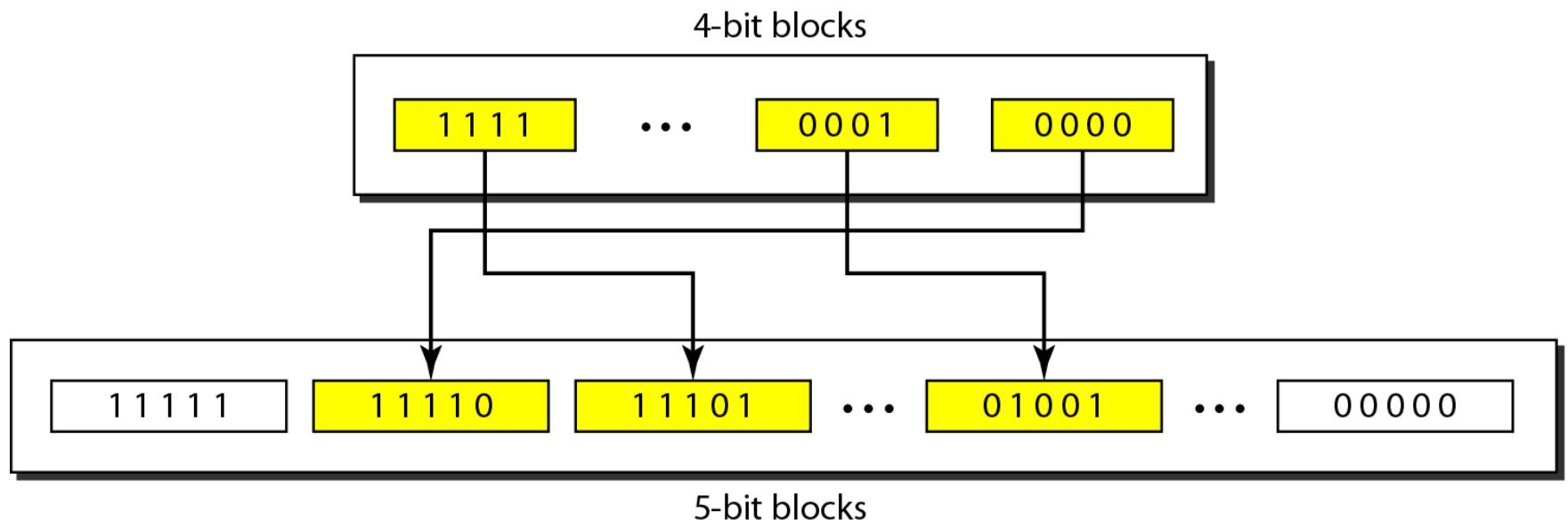
Uma solução é alterar o fluxo de bits, anterior à codificação com NRZ-I, de modo que ele não transmita um longo fluxo de 0s. O método 4B/5B atinge esse objetivo. O fluxo codificado em blocos não tem mais de três 0s consecutivos, como veremos a seguir.

Um grupo de 4 bits pode ter apenas 16 combinações diferentes ao passo que um grupo de 5 bits pode ter 32 combinações diferentes. Isso significa que existem 16 grupos que não são usados na codificação 4B/5B. Alguns desses grupos não utilizados são reaproveitados para fins de controle; os demais não são usados. Entretanto fornecem mecanismos de detecção de erros. Se chegar uma mensagem com um grupo de 5 bits que pertença à parte não utilizada da tabela, o receptor sabe que ocorreu um erro na transmissão.

mapa de códigos

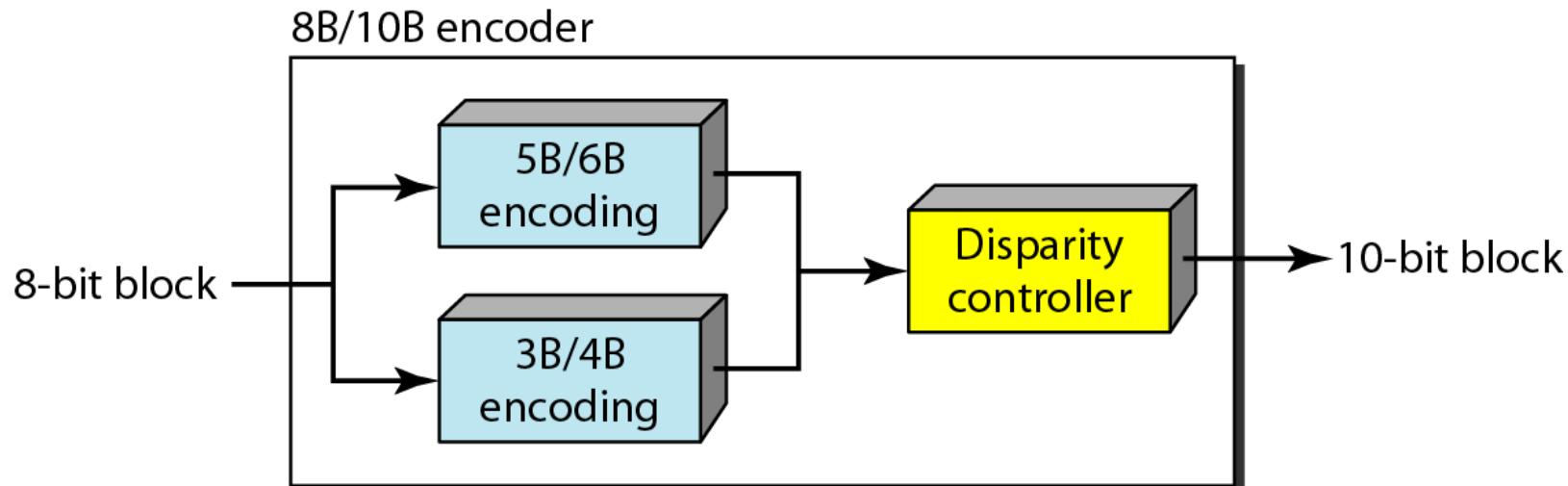
<i>Data Sequence</i>	<i>Encoded Sequence</i>	<i>Control Sequence</i>	<i>Encoded Sequence</i>
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

4B/5B com NRZ-I



8B/10B

A codificação oito binário, dez binário (8B/10B) é similar à codificação 4B/5B, exceto pelo fato de que agora um grupo de 8 bits de dados vai ser substituído por um código de 10 bits. Ela fornece maior capacidade de detecção de erros que a codificação 4B/5B. A codificação de blocos 8B/10B é, na verdade, uma combinação da codificação 5B/6B e 3B/4B, conforme mostrado na Figura abaixo.



B8ZS

A técnica B8ZS (bipolar com substituição de oito zeros) é usada comumente na América do Norte. Nessa técnica, oito voltagens consecutivas de nível zero são substituídas pela sequencia 000VB0VB.

O V nessa sequencia representa violação; trata-se de uma voltagem não-zero que quebra uma regra AMI de codificação (polaridade oposta em relação à anterior). O B na sequencia representa bipolar, o que significa um nível de voltagem não-zero de acordo com a regra AMI.

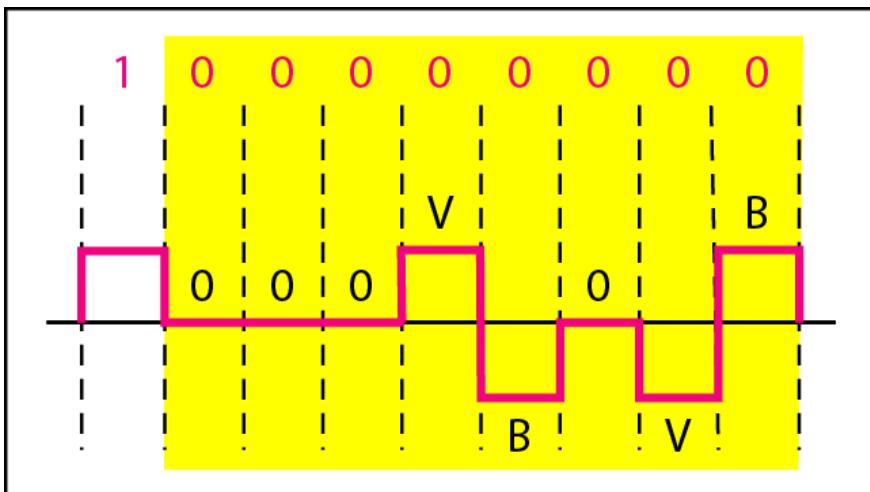
Observe que a mistura de sinais nesse caso não altera a taxa de bits. Da mesma forma, a técnica equilibra os níveis de voltagem positivo e negativo, equilíbrio DC é mantido.

B8ZS

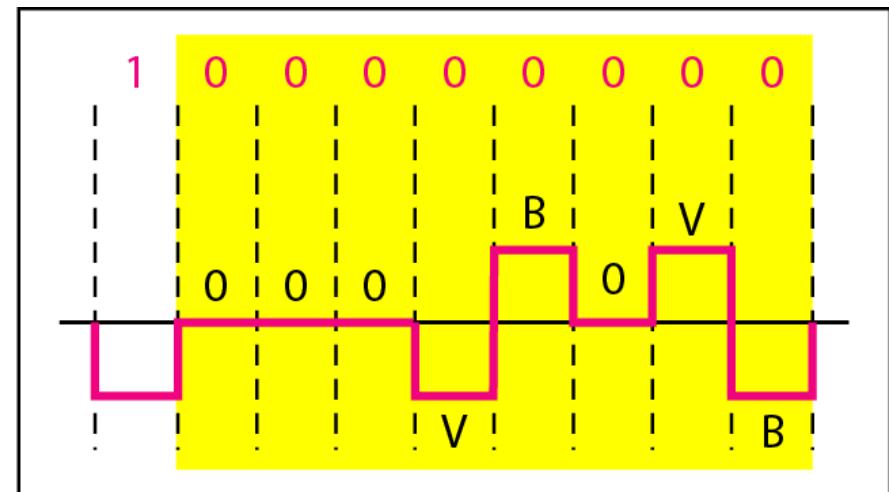
Existem dois casos, conforme mostrado na Figura abaixo.

O V significa a mesma polaridade da polaridade do pulso não-zero anterior;

B significa polaridade oposta à polaridade do pulso não-zero anterior.



a. Previous level is positive.



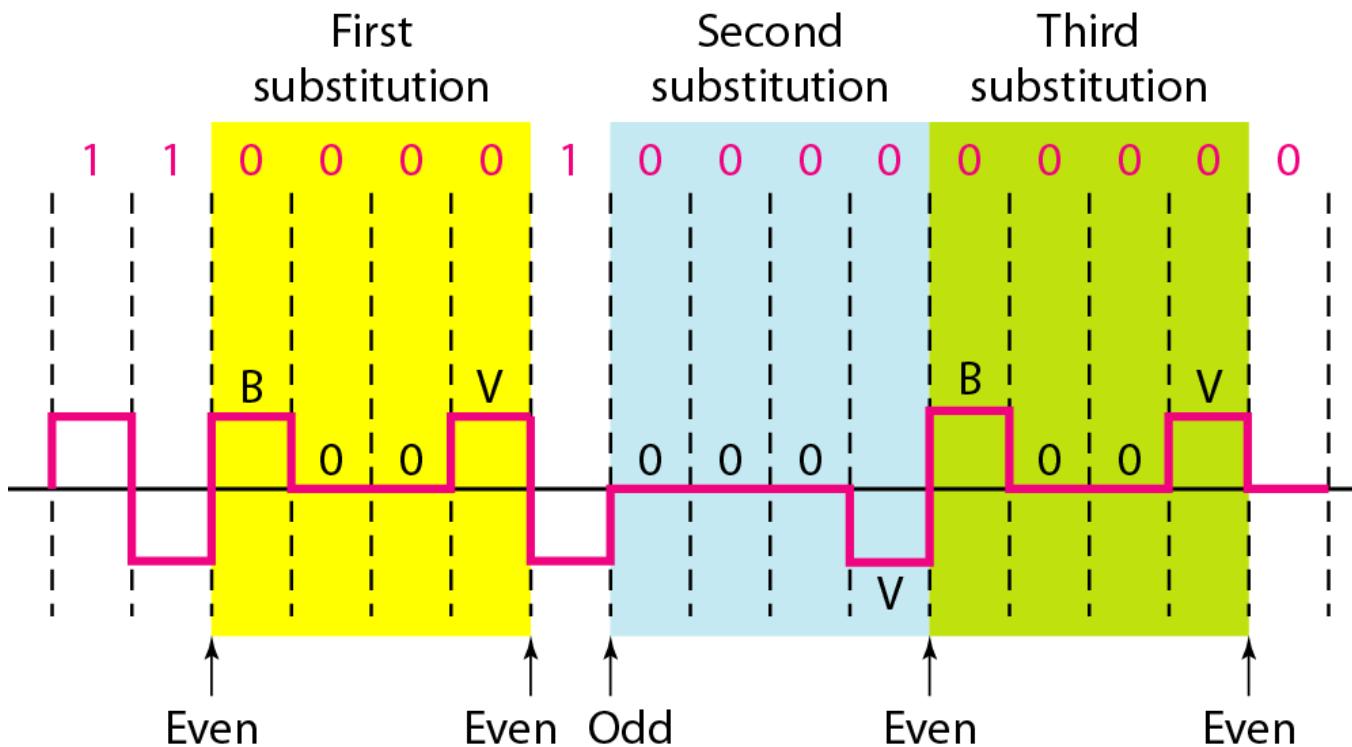
b. Previous level is negative.

HDB3

O método HDB3 (alta densidade bipolar com três zeros) é usado geralmente fora da América do Norte. Nessa técnica, que é mais conservadora que a B8ZS, quatro voltagens consecutivas de nível zero são substituídas por uma sequencia de 000V ou B00V. O motivo para duas substituições diferentes é manter o número par de pulsos não-zero após cada substituição. As duas regras podem ser assim declaradas:

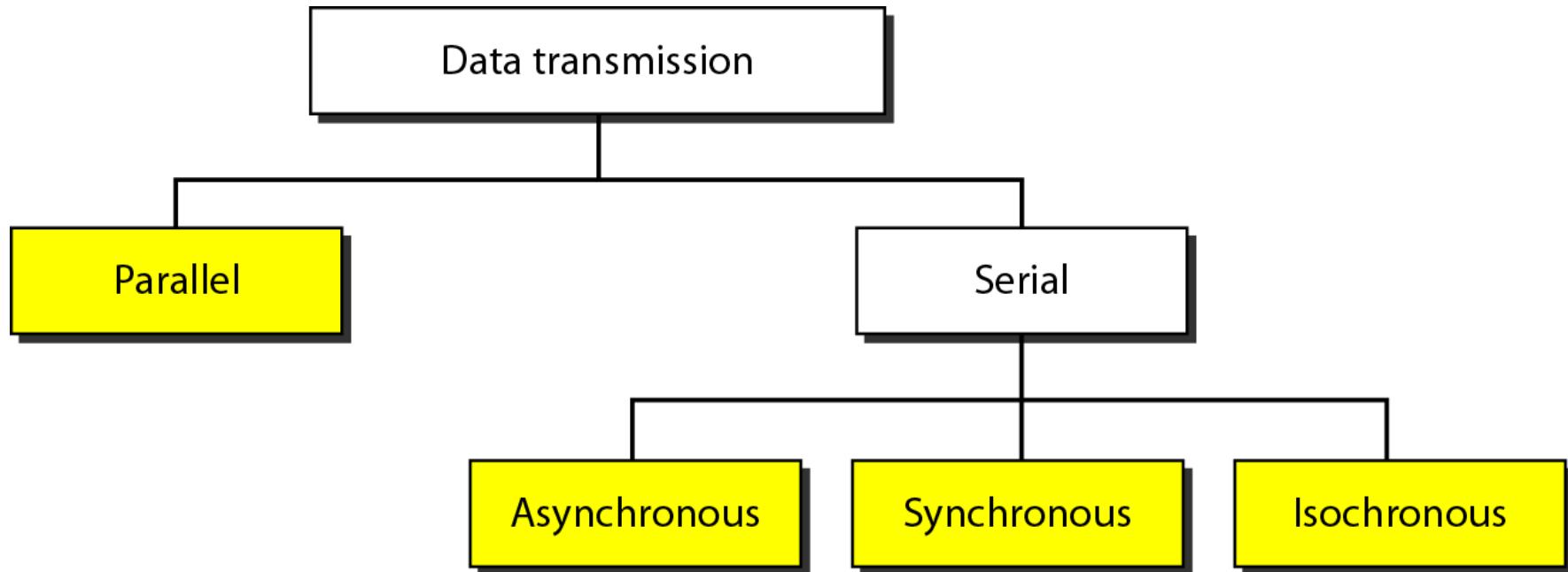
1. Se o número de pulsos não zero após a última substituição for ímpar, o padrão de substituição será 000V, o que torna o número total de pulsos não-zero par.
2. Se o número de pulsos não-zero após a última substituição for par, o padrão de substituição será B00V, o que torna o número total de pulsos não-zero par.

Figure 4.20 *Different situations in HDB3 scrambling technique*



Modos de transmissão

A transmissão de dados binários por um enlace pode ser realizada em modo paralelo ou em modo serial. No modo paralelo, vários *bits* são enviados a cada pulso de *clock*. No modo serial, somente 1 bit é enviado a cada pulso de *clock*. Embora haja apenas uma maneira de enviar dados paralelos, existem três subclasses de transmissão serial: assíncrona, síncrona e isócrona



Transmissão Paralela

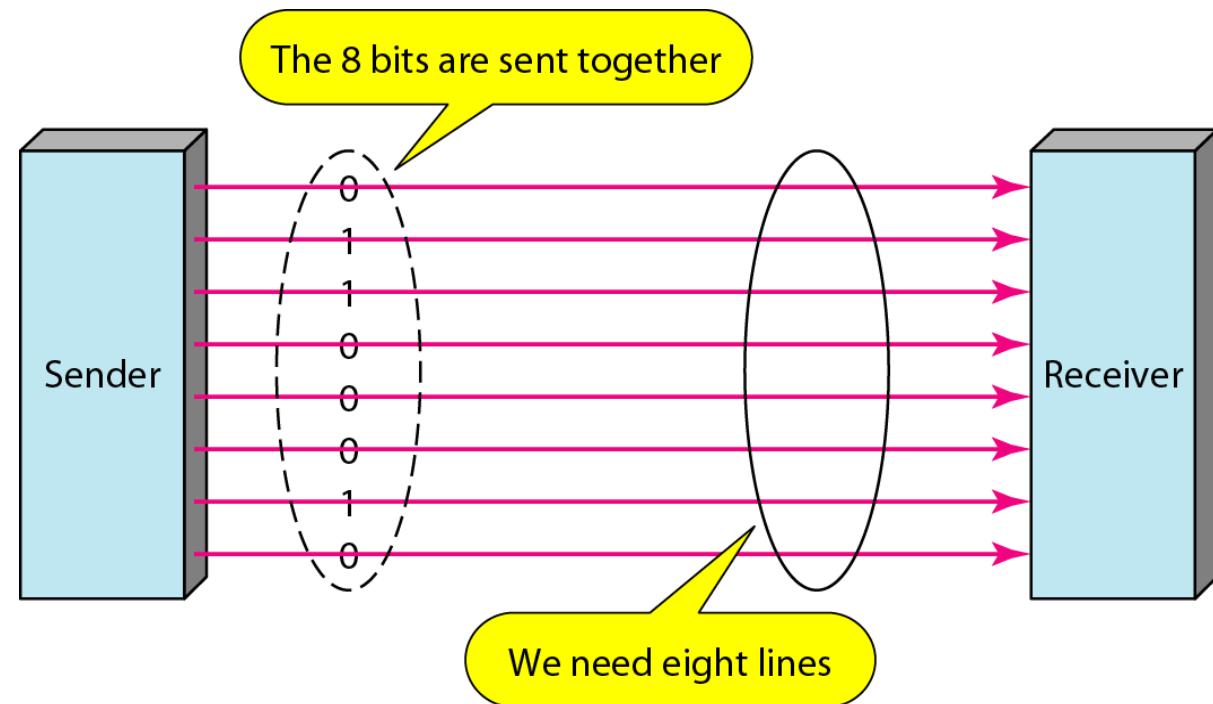
Dados binários, formados de bits 1 e 0, podem ser organizados em grupos de n bits cada. Por meio do agrupamento, podemos enviar dados em blocos de n bits por vez, em vez, de 1 único bit. Isso é chamado transmissão paralela.

O mecanismo para a transmissão paralela é conceitualmente simples: Use n fios para enviar n bits por vez. Dessa maneira, cada bit tem seu próprio fio e todos os n bits de um grupo podem ser transmitidos a cada pulso de *clock* de um dispositivo a outro. mostra como a transmissão paralela funciona quando n = 8. Normalmente, os oito fios são agrupados em um cabo com um conector em cada ponta.

Transmissão Paralela

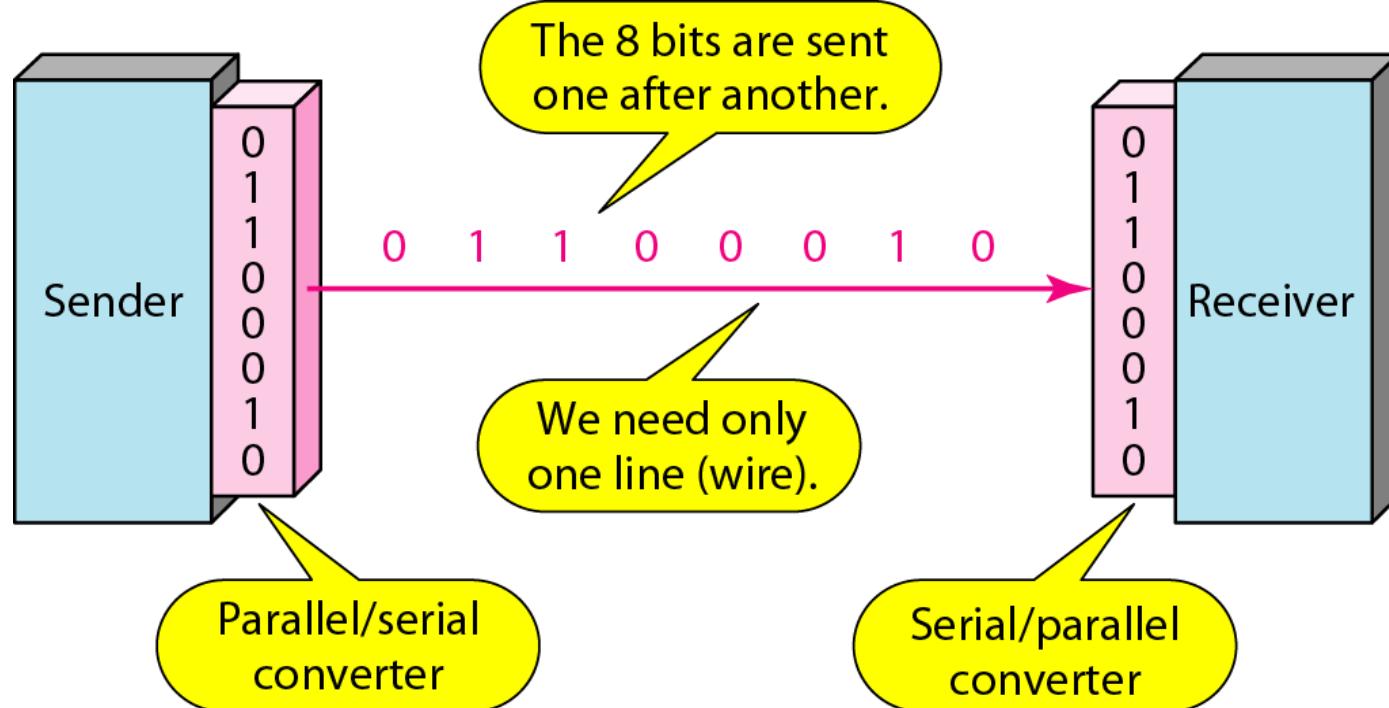
A vantagem da transmissão paralela é a velocidade de transmissão.

Com todo o restante igual, a transmissão paralela pode aumentar a velocidade de transferência por um fator de n em relação à transmissão serial



Transmissão Serial

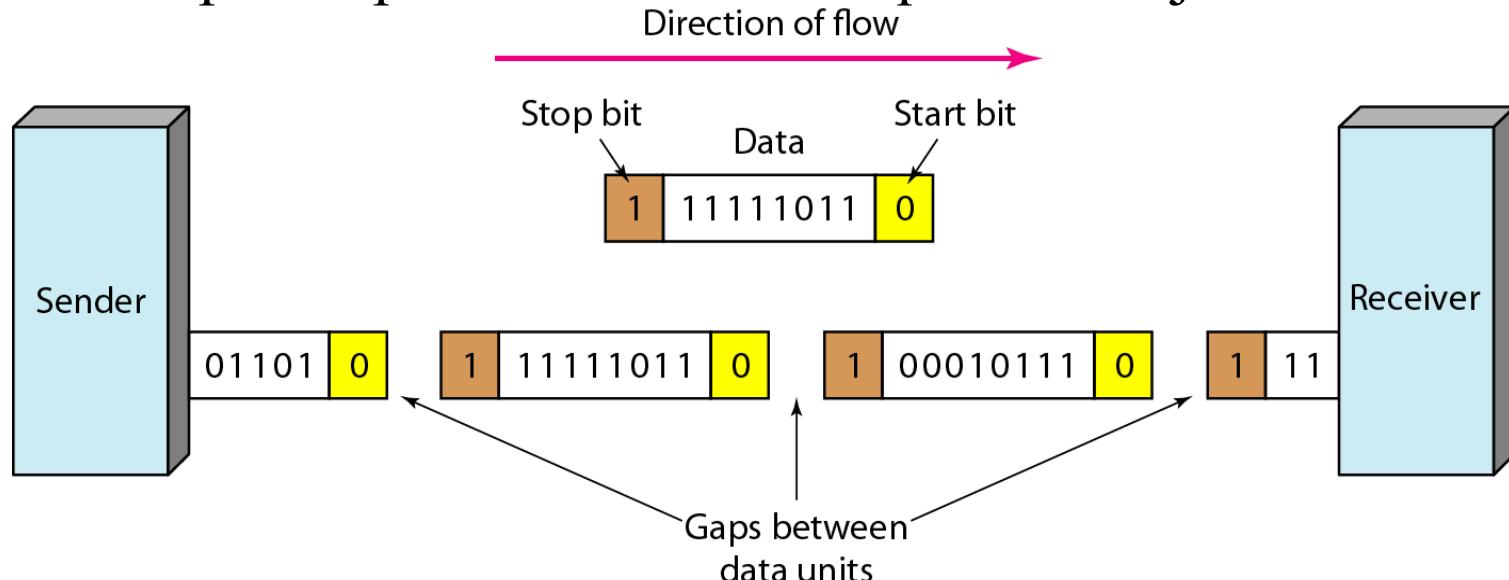
Na transmissão serial, um bit segue o outro; portanto, precisamos de apenas um canal de comunicação em vez de n canais para transmitir dados entre dois dispositivos comunicantes (ver Figura abaixo).



Transmissão Assíncrona

A transmissão assíncrona é assim chamada porque o intervalo de tempo entre mensagens não é importante. Em vez disso, as informações são recebidas e convertidas em padrões estabelecidos. Desde que esses padrões sejam seguidos, o dispositivo receptor pode recuperar as informações sem se preocupar com o ritmo no qual elas sejam enviadas

1 bit de start
(0) no inicio
e 1 ou mais
bits de stop
(1s) no final
intervalo
entre bytes.

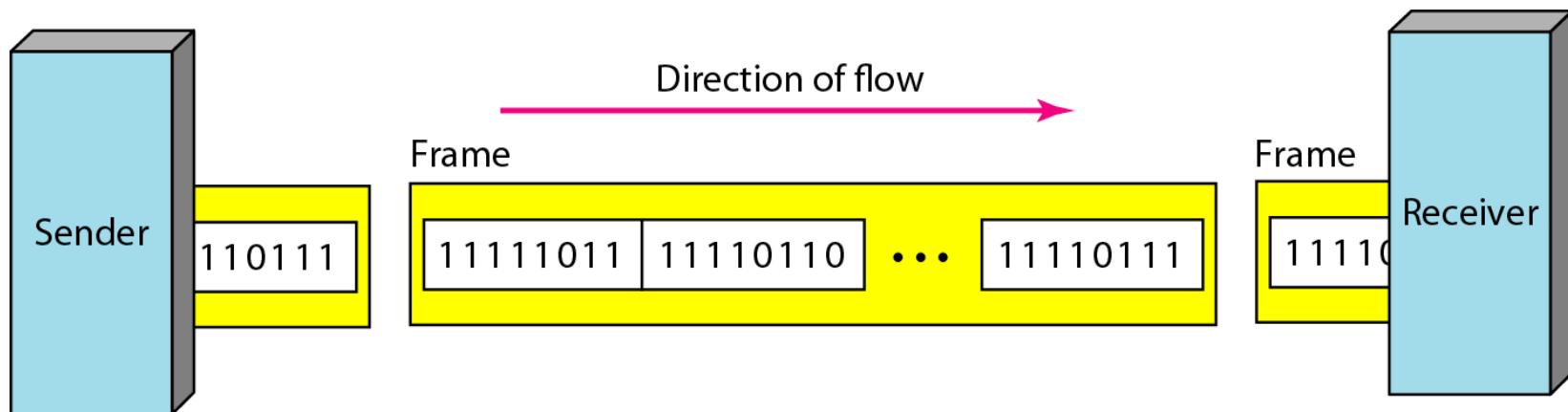


Transmissão Síncrona

O fluxo de bits é combinado em “frames” mais longos, que podem conter vários bytes. Cada byte, entretanto, é introduzido no enlace de transmissão sem um intervalo entre ele e o próximo. Fica a cargo do receptor separar o fluxo de bits em bytes para fins de decodificação. Em outras palavras, os dados são transmitidos na forma de uma *string* ininterrupta de 1s e 0s e o receptor separa essa *string* em bytes, ou caracteres, que ela precise para reconstruir as informações.

Transmissão Síncrona

Na transmissão síncrona enviamos bits um após o outro sem bits de start ou de stop ou intervalos. É de responsabilidade do receptor agrupar os bits.



Transmissão Isócrona

Em áudio e vídeo em tempo real, no qual retardos desiguais entre frames não são aceitáveis, a transmissão síncrona falha. Por exemplo, imagens de TV são transmitidas a uma taxa de 30 imagens por segundo; elas tem de ser visualizadas na mesma taxa. Se cada imagem for enviada usando um ou mais frames, não podem existir atrasos entre os frames. Para esse tipo de aplicação, somente a sincronização entre caracteres não é suficiente; o fluxo inteiro de bits deve ser sincronizado. A transmissão isócrona garante que os dados cheguem a uma taxa fixa.

Exercícios