

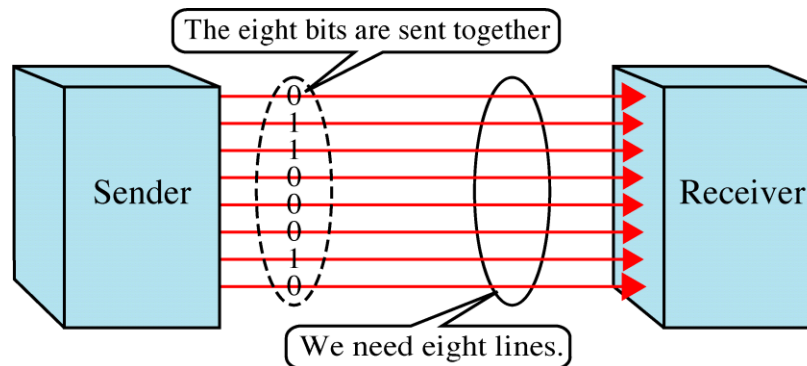
# CHƯƠNG 4

## KỸ THUẬT TRUYỀN DỮ LIỆU SỐ

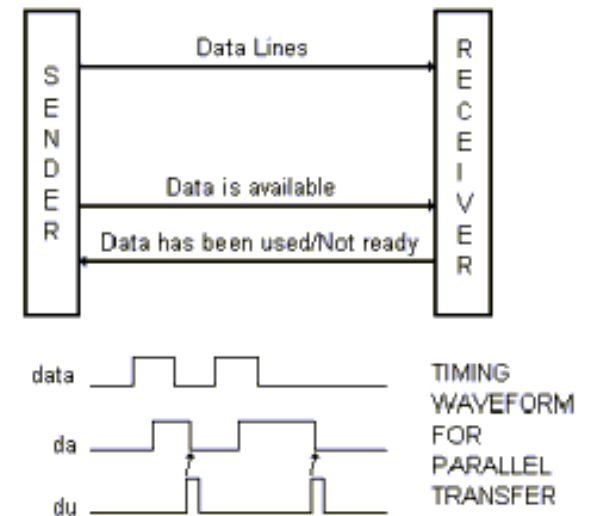
# Cấu trúc kênh truyền

## ■ Song song (Parallel)

- ❑ Mỗi bit dùng một đường truyền riêng. Nếu có 8 bits được truyền đồng thời sẽ yêu cầu 8 đường truyền độc lập
- ❑ Để truyền dữ liệu trên một đường truyền song song, một kênh truyền riêng được dùng để thông báo cho bên nhận biết khi nào dữ liệu có sẵn (clock signal)
- ❑ Cần thêm một kênh truyền khác để bên nhận báo cho bên gửi biết là đã sẵn sàng để nhận dữ liệu kế tiếp



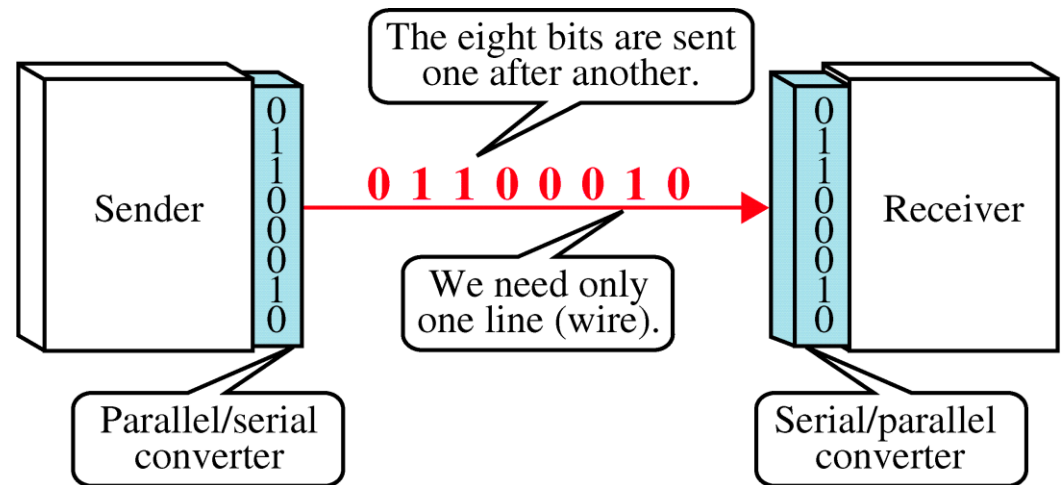
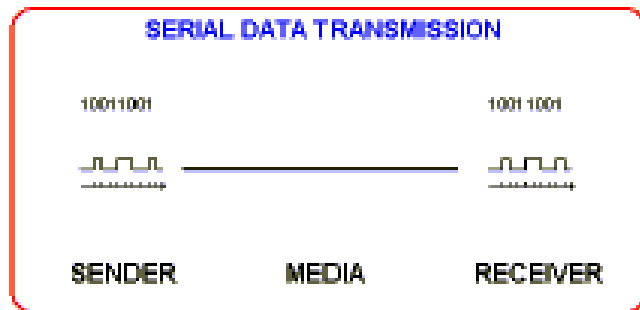
Parallel Transmission using Clocks



# Cấu trúc kênh truyền

## ■ Tuần tự (Serial)

- ❑ Tất cả các bit đều được truyền trên cùng một đường truyền, bit này tiếp theo sau bit kia
- ❑ Không cần các đường truyền riêng cho tín hiệu đồng bộ và tín hiệu bắt tay (các tín hiệu này được mã hóa vào dữ liệu truyền đi)



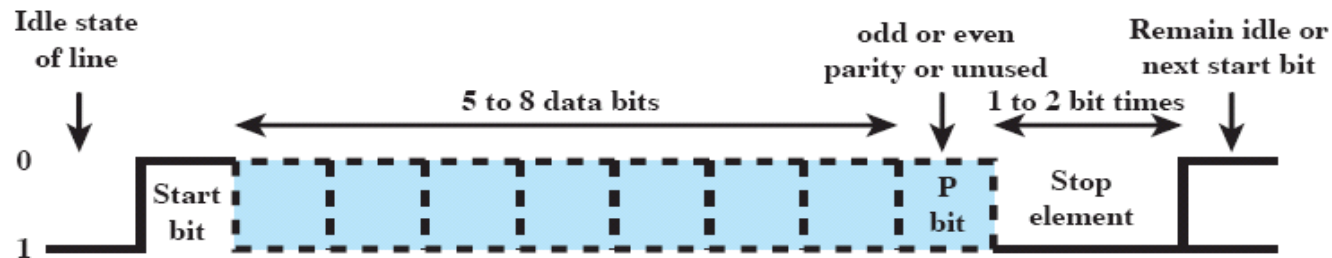
# Truyền bất đồng bộ và đồng bộ

- Những yêu cầu định vị thời gian (timing) đòi hỏi một cơ chế đồng bộ giữa máy gửi và máy nhận
- Có 2 giải pháp
  - Bất đồng bộ: mỗi ký tự được đồng bộ bởi start và stop bit
  - Đồng bộ: mỗi khối ký tự được đồng bộ dùng cờ

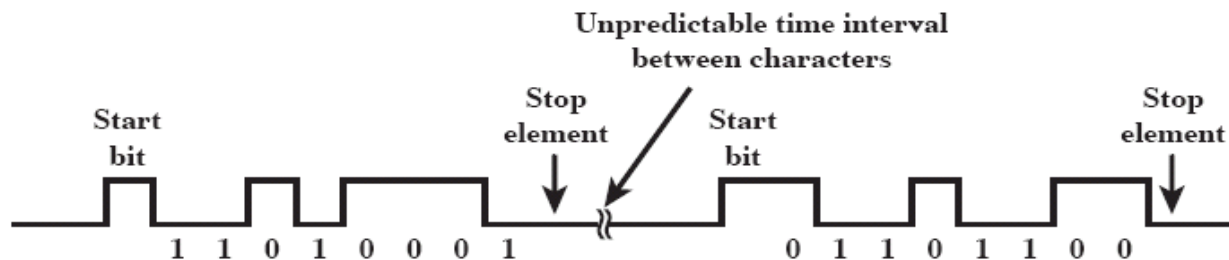
# Truyền bất đồng bộ

- Dữ liệu được truyền theo ký tự (5 → 8 bits)
- Định vị thời gian chỉ cần giữ trong mỗi một ký tự
- Thời gian sẽ được tái đồng bộ cho mỗi ký tự mới

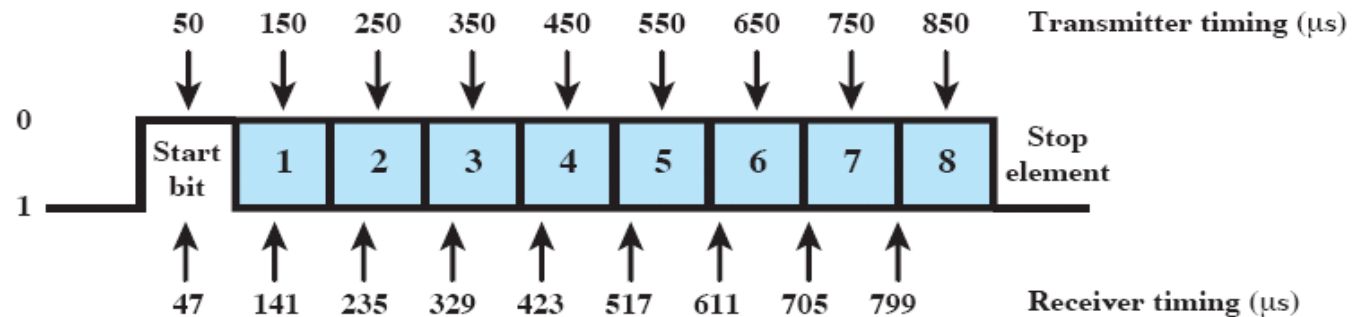
# Truyền bất đồng bộ



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

# Cơ chế hoạt động của truyền bất đồng bộ

- Đối với dòng dữ liệu đều, khoảng cách giữa các ký tự là đồng nhất (chiều dài của phần tử stop)
- Ở trạng thái không truyền, bộ thu tìm - xác định sự chuyển  $1 \rightarrow 0$
- Lấy mẫu 7 khoảng kế tiếp (chiều dài ký tự)
- Đợi việc chuyển  $1 \rightarrow 0$  cho ký tự kế tiếp
- Đơn giản
- Rẻ
- Phí tổn 2 hoặc 3 bit cho một ký tự (~20%)
- Thích hợp cho dữ liệu với khoảng trống giữa các ký tự lớn (dữ liệu nhập từ bàn phím)

# Truyền đồng bộ - mức bit

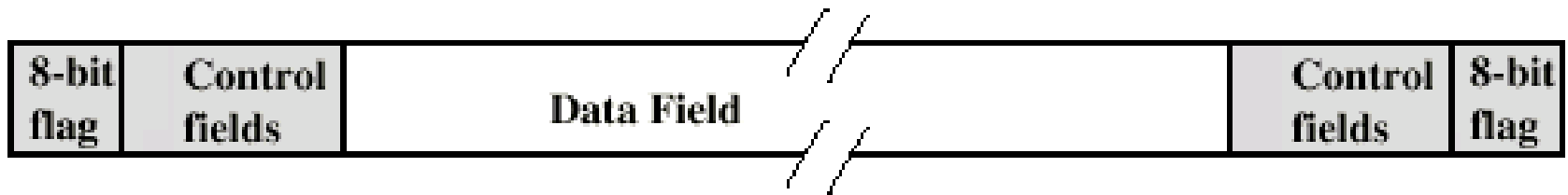
- Truyền các khối dữ liệu không cần start/stop bits
- Các đồng hồ tại các máy truyền và nhận cần đồng bộ
- Dùng một đường tín hiệu đồng bộ riêng biệt
  - Sử dụng một nguồn clock ổn định được giữ đồng bộ với dữ liệu đến tại nơi nhận
  - Tốt với khoảng cách ngắn
  - Tín hiệu đồng bộ dễ bị suy giảm trên đường truyền
- Tích hợp thông tin đồng bộ (clock) vào trong dữ liệu truyền
  - Máy nhận sẽ tách thông tin đồng bộ dựa vào dữ liệu nhận được
  - Dùng các phương pháp mã như Manchester, differential Manchester



# Truyền đồng bộ - mức nhóm (block)

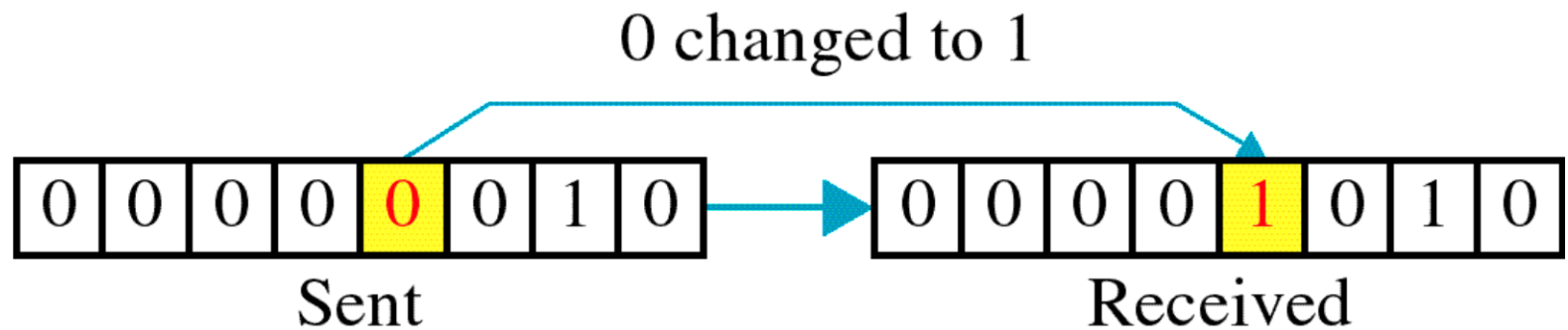
- Cần xác định đâu là bắt đầu và kết thúc của một nhóm
- Sử dụng ký tự đầu và kết thúc
  - Ví dụ chuỗi ký tự SYN (hex 16)
  - Một nhóm 111111111 kết thúc với 11111110
- Hiệu quả (phí tổn thấp) hơn so với truyền bất đồng bộ

# Truyền đồng bộ



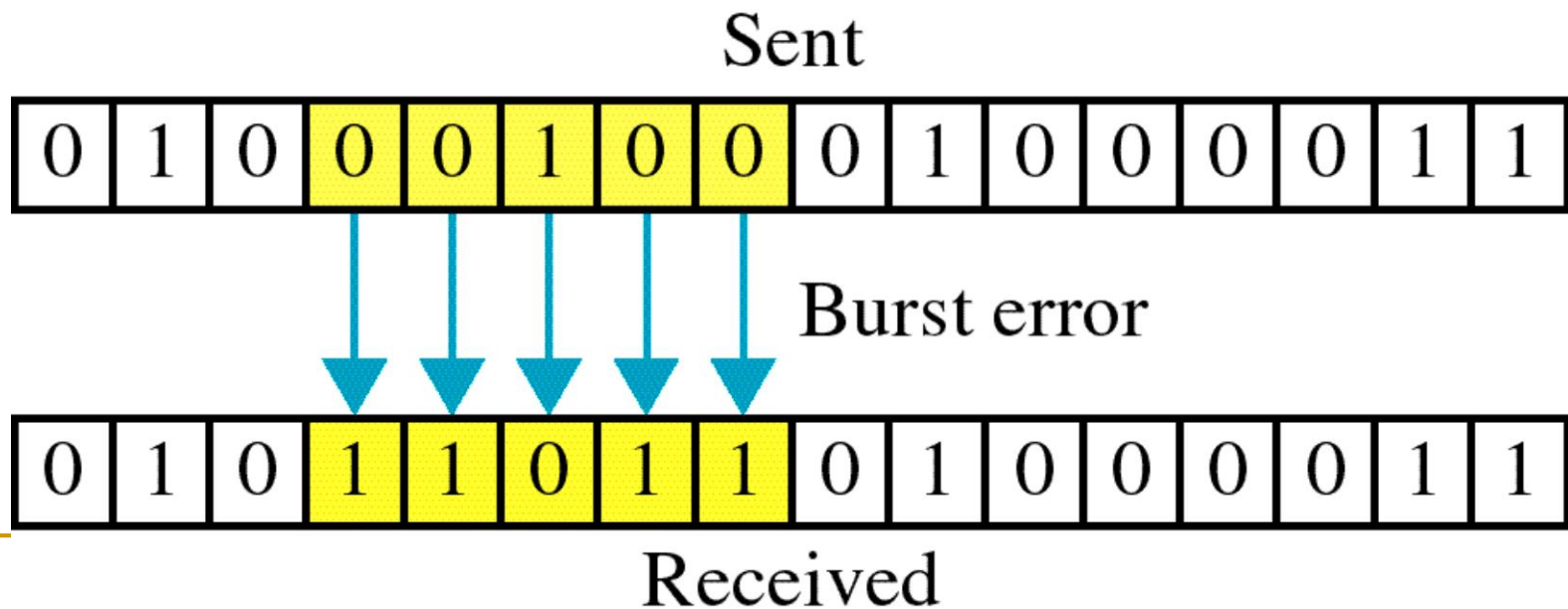
# Lỗi

- Một lỗi xuất hiện khi có một bit bị thay đổi giữa truyền và nhận
- Các lỗi bit đơn
  - Một bit bị thay đổi
  - Các bit bên cạnh không đổi
  - Trong trường hợp do nhiễu trắng

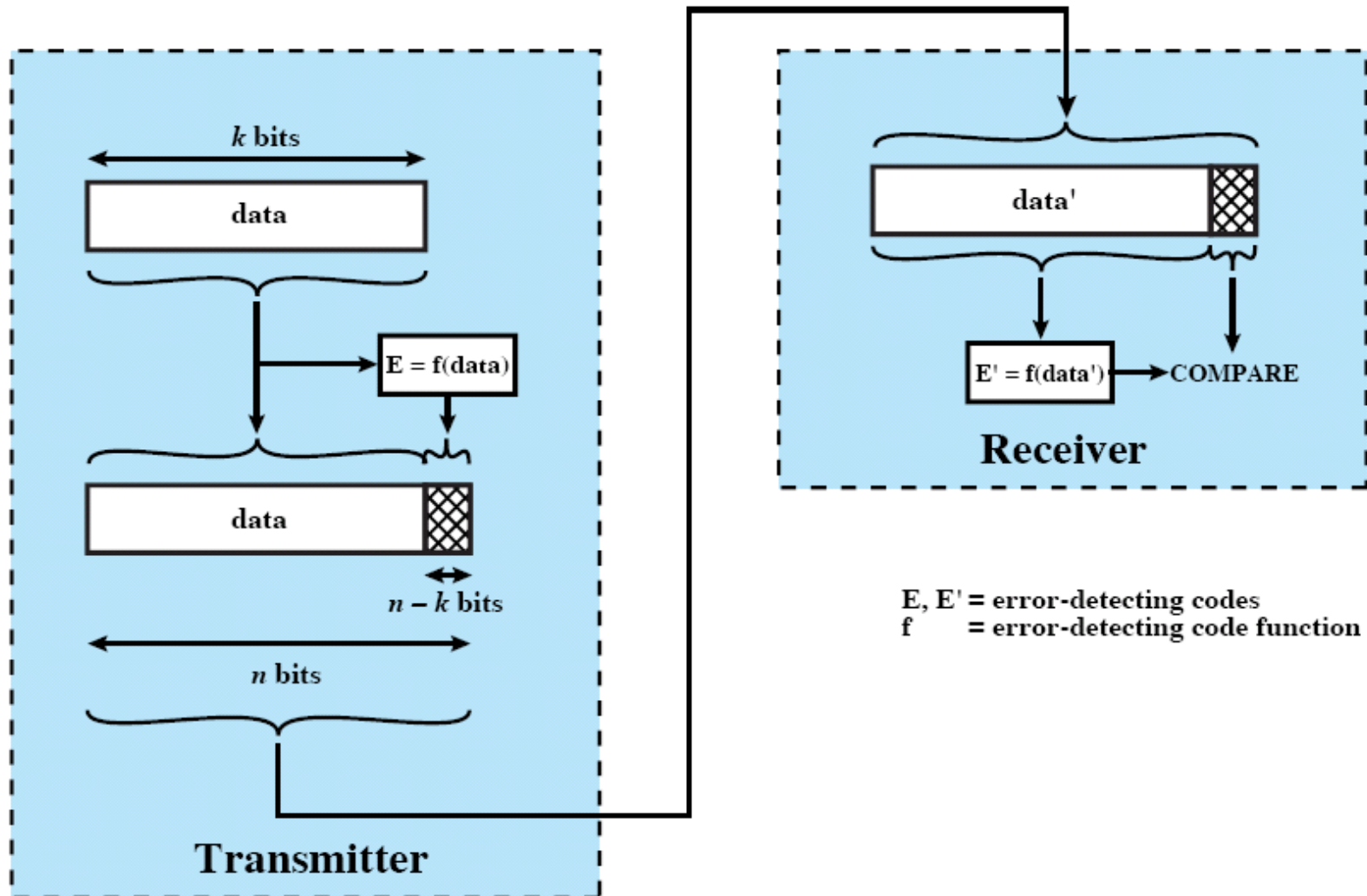


# Lỗi sai nhóm

- Chiều dài B
- Một chuỗi B bits trong đó bit đầu, cuối và nhiều bit liên tiếp trong đó bị sai
- Do nhiễu xung
- Làm suy giảm trong truyền không dây
- Tác động càng lớn khi đường truyền càng cao



# Quá trình phát hiện sai

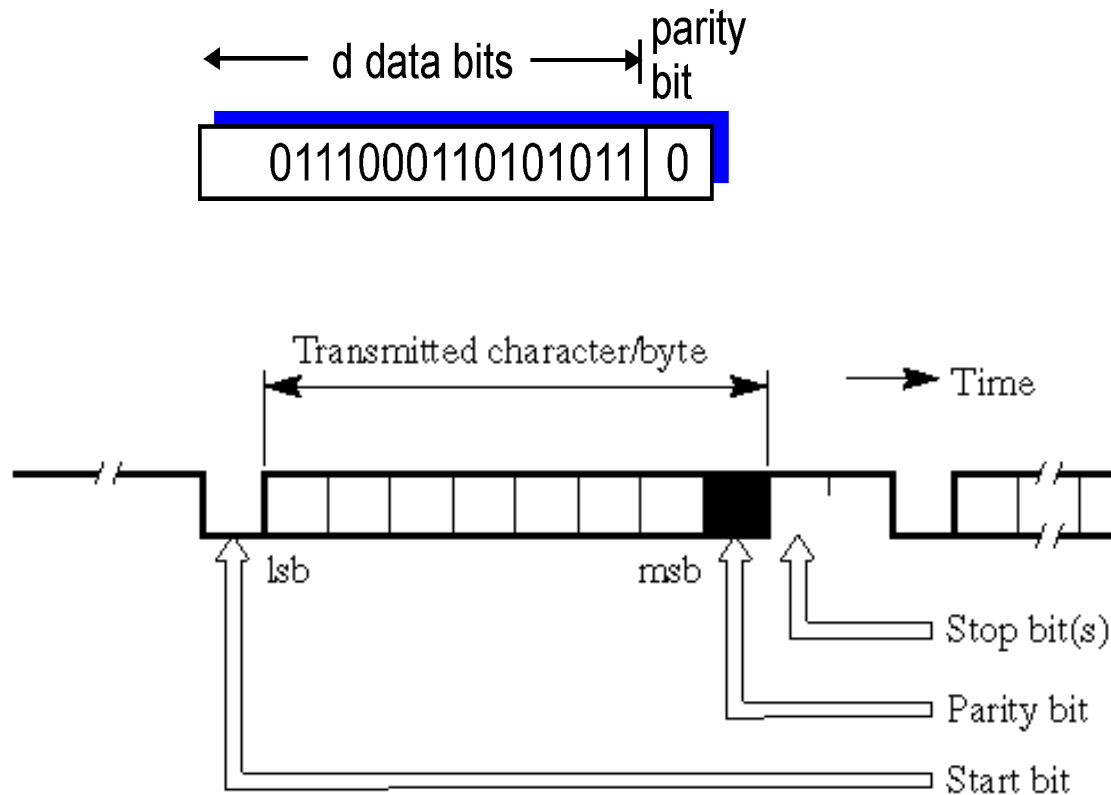


# Phát hiện lỗi

- Thêm các bits bổ xung bởi máy gửi cho mã xác định lỗi
- Parity
  - Giá trị của bit parity cho vào sau ký tự 7 bits là 0 nếu có chẵn bit 1 và 1 nếu có lẻ bit 1.
  - Nếu có chẵn số bit lỗi thì không phát hiện ra

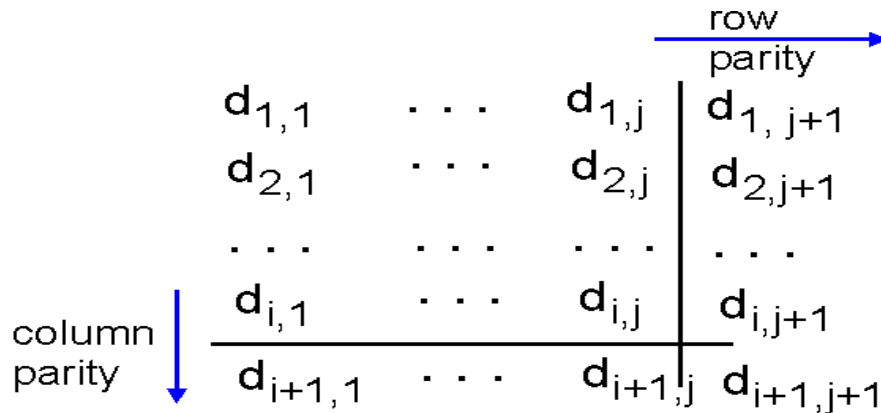
# Kiểm tra bằng 1 bit parity

Xác định được các lỗi 1 bit



# Bit Parity hai chiều

Xác định và sửa các lỗi bit đơn



1	0	1	0	1		1
1	1	1	1	0		0
0	1	1	1	0		1
<hr/>						
1	0	1	0	1		0

*no errors*

1	0	1	0	1		1
1	0	1	1	0		0
0	1	1	1	0		1
<hr/>						
1	0	1	0	1		0

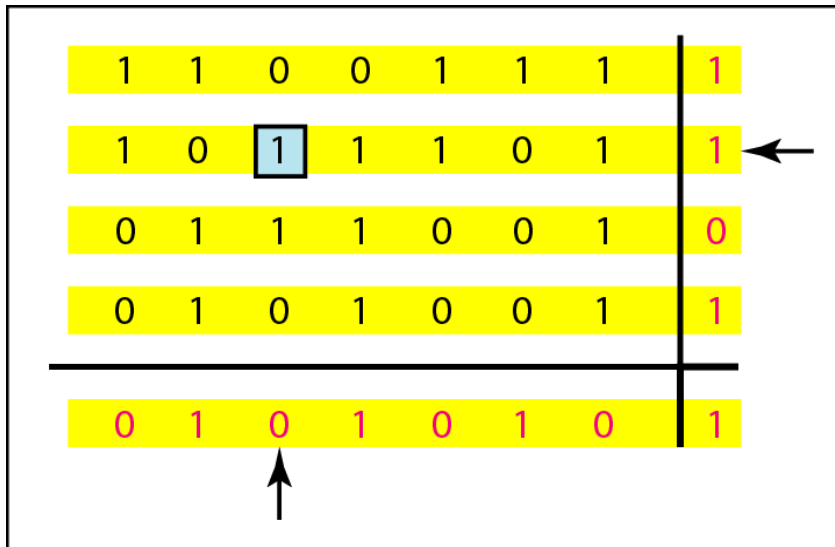
parity error

parity error

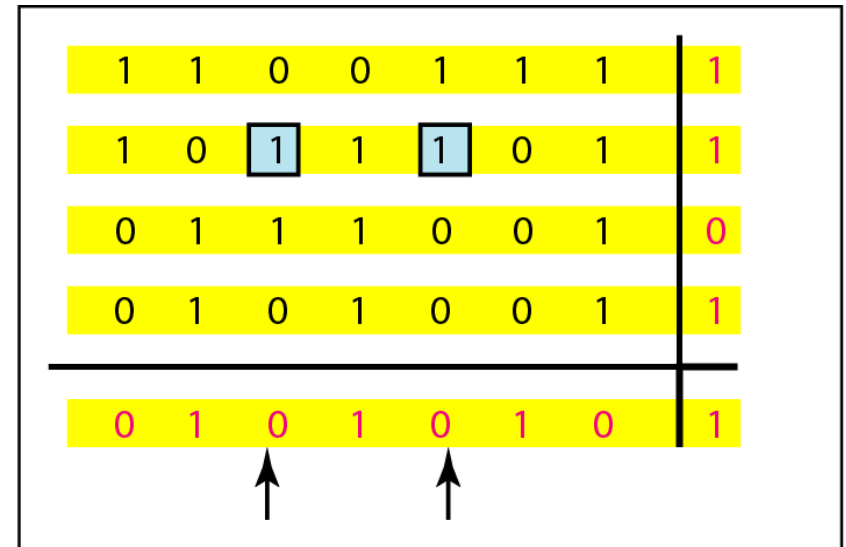
*correctable  
single bit error*



# Bit Parity hai chiều

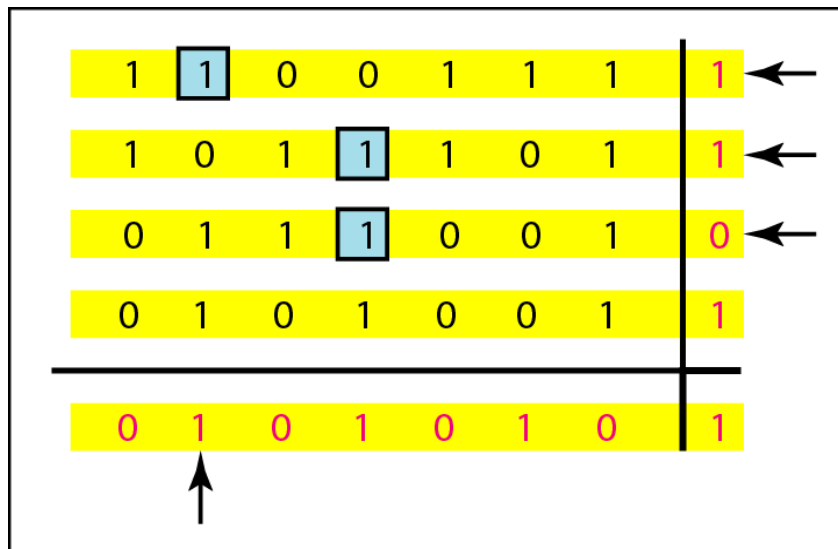


b. One error affects two parities

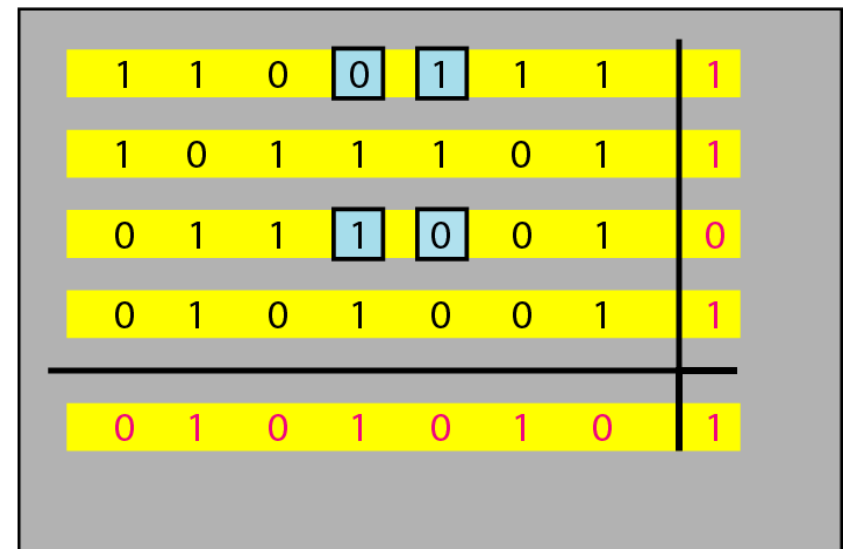


c. Two errors affect two parities

# Bit Parity hai chiều



d. Three errors affect four parities

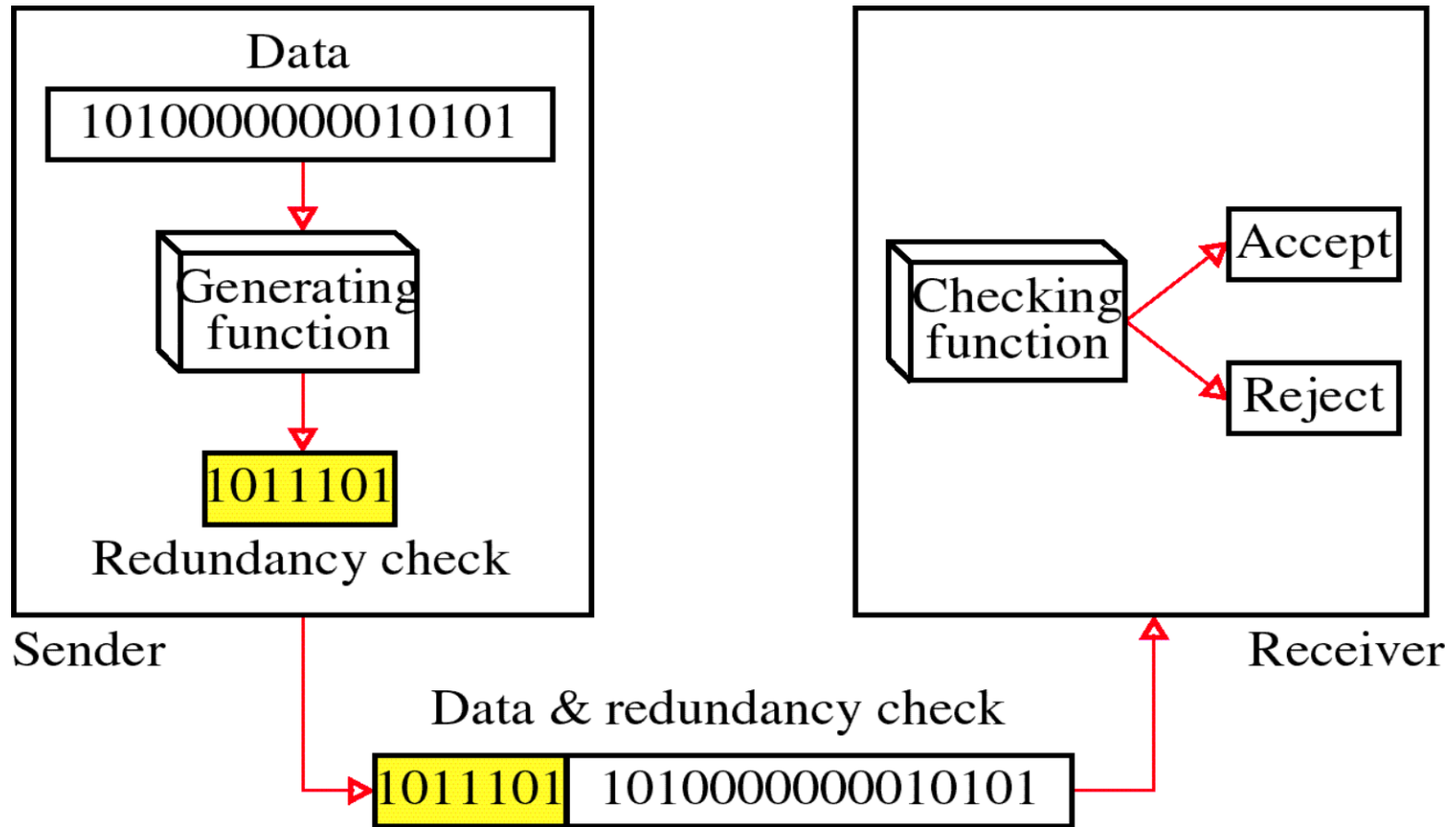


e. Four errors cannot be detected

# Cyclic Redundancy Check (CRC)

- Với  $k$ -bit phát, máy phát tạo ra chuỗi  $n$  bit kiểm tra FCS (Frame Check Sequence)
- Gửi  $k+n$  bit chia hết cho số kiểm tra  $P$  ( $n+1$ ) bit xác định trước
- Máy thu chia (modulo 2) frame nhận được cho cùng số kiểm tra  $P$  nếu không có phần dư thì có khả năng không có lỗi

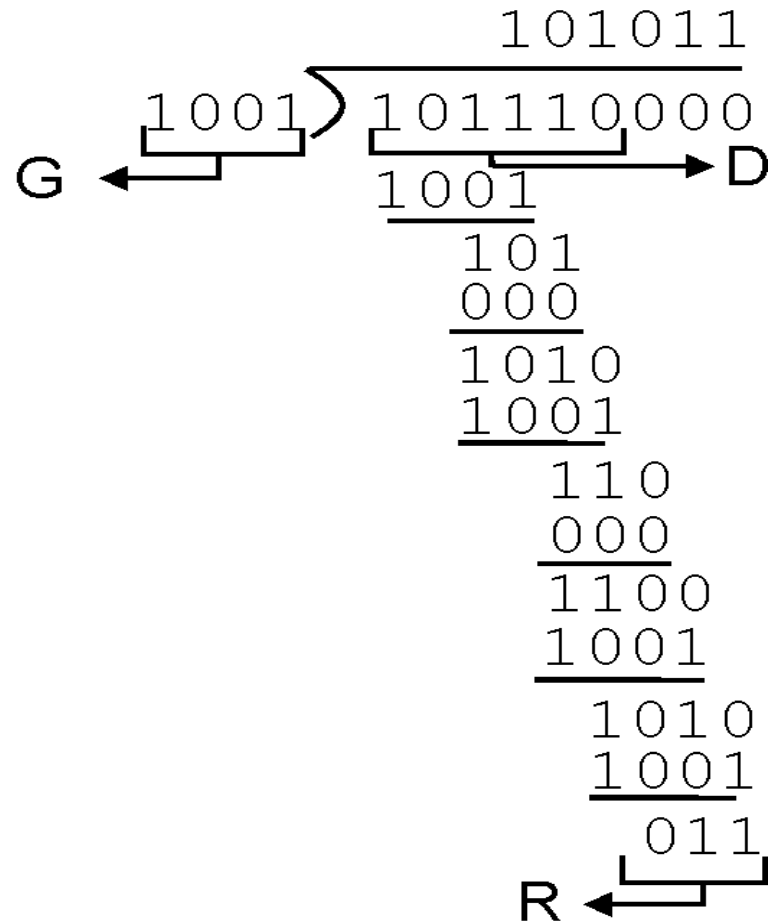
# CRC



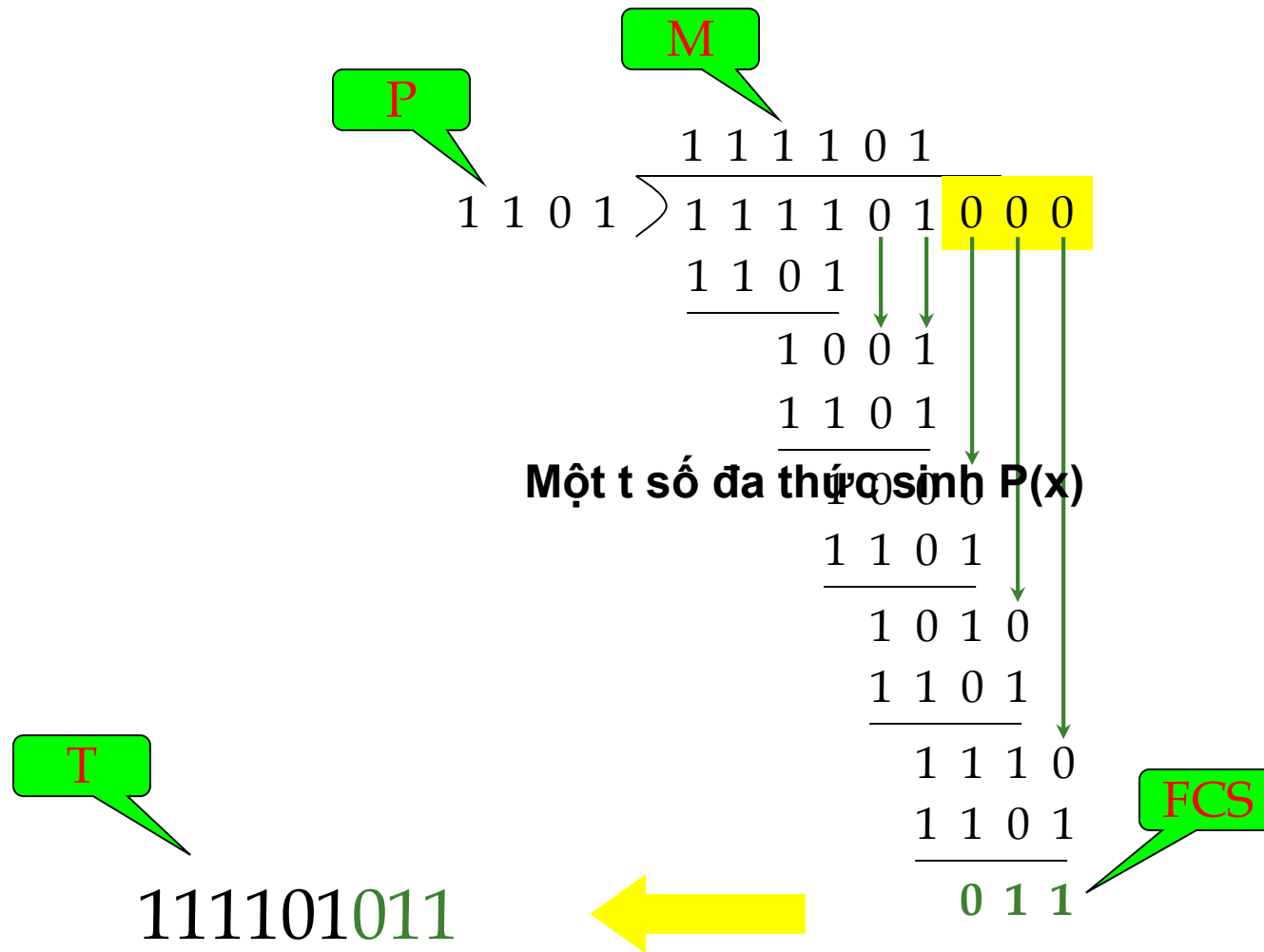
# CRC Example

- Chúng ta muốn:
  - $D \cdot 2^r \text{ XOR } R = nG$
- *Tương đương:*
  - $D \cdot 2^r = nG \text{ XOR } R$
- *Tương đương:*
  - Nếu chúng ta chia  $D \cdot 2^r$  với  $G$ , giá trị còn lại  $R$

$$R = \text{remainder} \left[ \frac{D \cdot 2^r}{G} \right]$$



# Cyclic Redundancy Check



# Đa thức sinh $P(x)$

- Các chuỗi  $P$  thường biểu diễn bằng 1 đa thức theo biến  $x \rightarrow P(x)$  gọi là đa thức sinh
- Bậc của  $x$  chỉ trọng số, và hệ số là các số nhị phân
- Ví dụ:  
chuỗi 1101 được biểu diễn là:  $x^3 + x^2 + 1$

# Cyclic Redundancy Check

- Cách xác định FCS bằng chia đa thức
  - $M = 111101 \rightarrow M(x) = X^5 + X^4 + X^3 + X^2 + 1$
  - $P = 1101 \rightarrow P = X^3 + X^2 + 1$
  - $\Rightarrow$  FCS có 3 bits ( $n = 3$ )
  - Dữ liệu dịch trái  $n$  bits:  $2^n M(x) = X^8 + X^7 + X^6 + X^5 + X^3$

$$\begin{array}{r}
 X^3 + X^2 + 1 \overline{) X^8 + X^7 + X^6 + X^5 + X^3} \\
 \underline{X^8 + X^7 \phantom{+ X^6} + X^5} \phantom{+ X^3} \\
 X^6 \phantom{+ X^5} + X^3 \\
 \underline{X^6 + X^5 + X^3} \\
 X^5 \\
 \underline{X^5 + X^4 + X^2} \\
 X^4 + X^2 \\
 \underline{X^4 + X^3 + X} \\
 X^3 + X^2 + X \\
 \underline{X^3 + X^2 + 1} \\
 X + 1
 \end{array}$$

$T = 111101011$



FCS = 011  $\Leftarrow$

$X + 1$



# Các CRCs thông dụng

CRC-1	$x + 1$ (most hardware; also known as <u>parity bit</u> )
CRC-4-ITU	$x^4 + x + 1$ ( <u>ITU G.704</u> , p. 12)
CRC-5-ITU	$x^5 + x^4 + x^2 + 1$ ( <u>ITU G.704</u> , p. 9)
CRC-5-USB	$x^5 + x^2 + 1$ ( <u>USB</u> token packets)
CRC-6-ITU	$x^6 + x + 1$ ( <u>ITU G.704</u> , p. 3)
CRC-7	$x^7 + x^3 + 1$ (telecom systems, <u>MMC</u> )
CRC-8- <u>ATM</u>	$x^8 + x^2 + x + 1$ (ATM <u>HEC</u> )
CRC-8- <u>CCITT</u>	$x^8 + x^7 + x^3 + x^2 + 1$ ( <u>1-Wire bus</u> )
CRC-8- <u>Dallas/Maxim</u>	$x^8 + x^5 + x^4 + 1$ ( <u>1-Wire bus</u> )
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
CRC-8-SAE J1850	$x^8 + x^4 + x^3 + x^2 + 1$
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$
CRC-11	$x^{11} + x^9 + x^8 + x^7 + x^2 + 1$ ( <u>FlexRay</u> )
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ (telecom systems)
CRC-15- <u>CAN</u>	$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$

# Các CRCs thông dụng

CRC-16-CCITT

$x^{16} + x^{12} + x^5 + 1$  ([X.25](#), [V.41](#), [CDMA](#), [Bluetooth](#), [PPP](#), [IrDA](#), [BACnet](#); known as *CRC-CCITT*)

CRC-16-[IBM](#)

$x^{16} + x^{15} + x^2 + 1$  ([SDLC](#), [XMODEM](#), [USB](#), also known as *CRC-16*)

CRC-24-[Radix-64](#)

$x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$  ([FlexRay](#))

CRC-30

$x^{30} + x^{29} + x^{21} + x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + x + 1$  ([CDMA](#))

CRC-32-[IEEE 802.3](#)

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$  ([V.42](#), [MPEG-2](#))

CRC-32K (Koopman)

$x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^7 + x^6 + x^4 + x^2 + x + 1$

CRC-64-ISO

$x^{64} + x^4 + x^3 + x + 1$  ([HDLC — ISO 3309](#))

CRC-64-[ECMA](#)-182

$x^{64} + x^{62} + x^{57} + x^{55} + x^{54} + x^{53} + x^{52} + x^{47} + x^{46} + x^{45} + x^{40} + x^{39} + x^{38} + x^{37} + x^{35} + x^{33} + x^{32} + x^{31} + x^{29} + x^{27} + x^{24} + x^{23} + x^{22} + x^{21} + x^{19} + x^{17} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x + 1$  (as described in [ECMA-182](#) p.63)

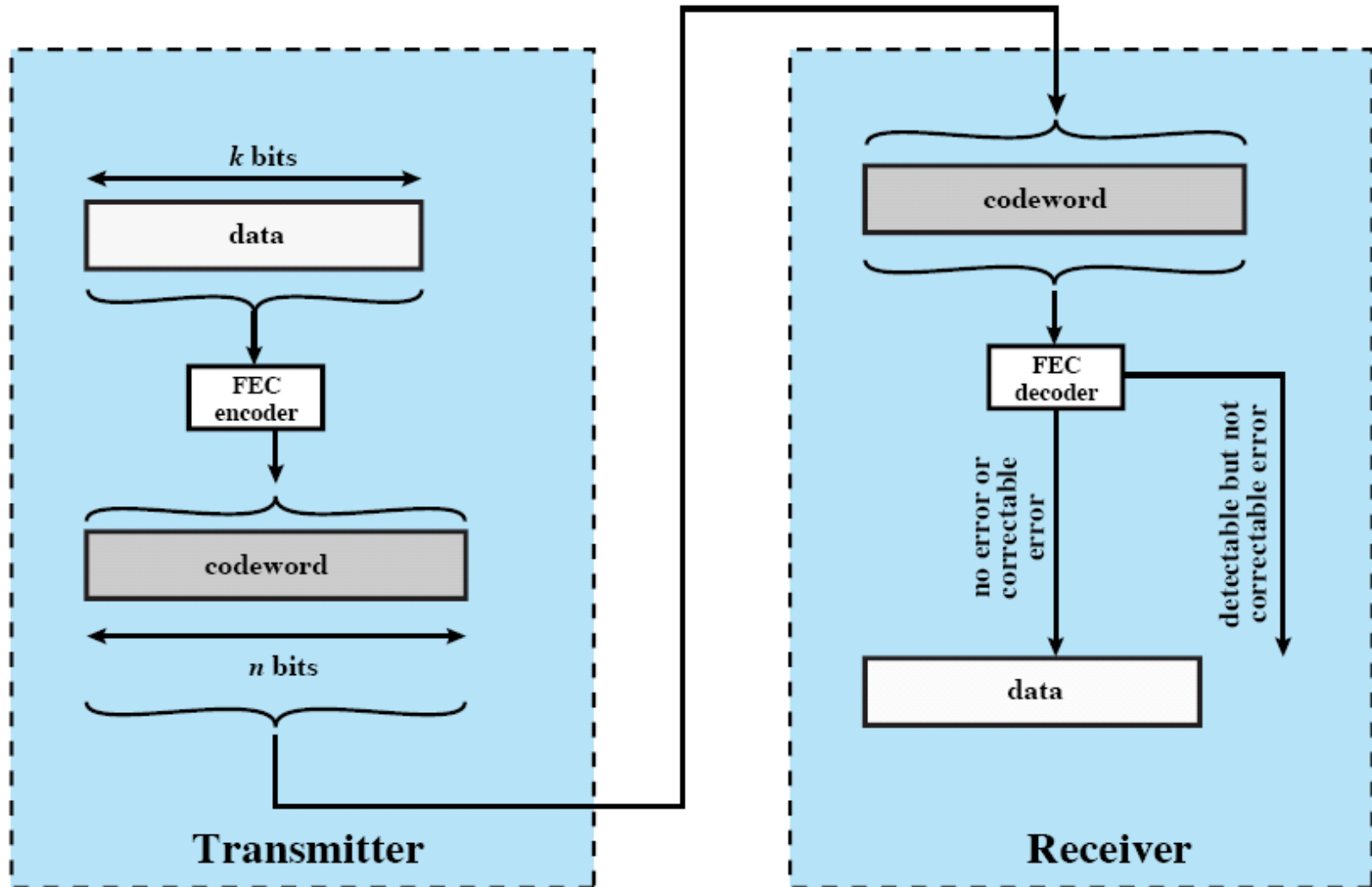
# Các lỗi được phát hiện

- Tất cả các lỗi bit đơn
- Tất cả các lỗi kép nếu  $P$  có ít nhất 3 toán hạng
- Một số lẻ lỗi bất kỳ nếu  $P$  chứa 1 thừa số  $(X+1)$
- Bất kỳ lỗi chùm nào mà chiều dài của chùm nhỏ hơn chiều dài FCS
- Hầu hết các lỗi chùm lớn hơn

# Sửa lỗi

- Sửa các lỗi được phát hiện thông thường yêu cầu truyền lại khối dữ liệu
- Không thích hợp cho các ứng dụng trao đổi dữ liệu không dây
  - BER cao
    - Truyền lại nhiều
  - Thời gian trễ truyền lớn hơn nhiều so với thời gian truyền dữ liệu (vd truyền vệ tinh)
    - Khối dữ liệu được truyền lại bị lỗi và nhiều khối dữ liệu khác tiếp theo
- Cần thiết phải sửa lỗi dựa vào các dữ liệu nhận được

# Quá trình sửa lỗi



# Quá trình sửa lỗi

- Mỗi khối dữ liệu  $k$  bit được ánh xạ vào khối  $n$  bit ( $n > k$ )
  - Từ mã – Codeword
  - Forward error correction (FEC) encoder
- Codeword được truyền đi
- Chuỗi bit nhận được tương tự như chuỗi được truyền đi, nhưng có chứa lỗi
- Codeword được gửi tới bộ giải mã FEC
  - Nếu không có lỗi, trích xuất khối dữ liệu ban đầu
  - Một vài mẫu lỗi có thể được phát hiện và sửa lỗi
  - Một vài mẫu lỗi có thể được phát hiện nhưng không sửa được
  - Một vài mẫu lỗi có thể không được phát hiện (ít xảy ra)
    - FEC trích xuất khối dữ liệu sai

# Block Code Principles

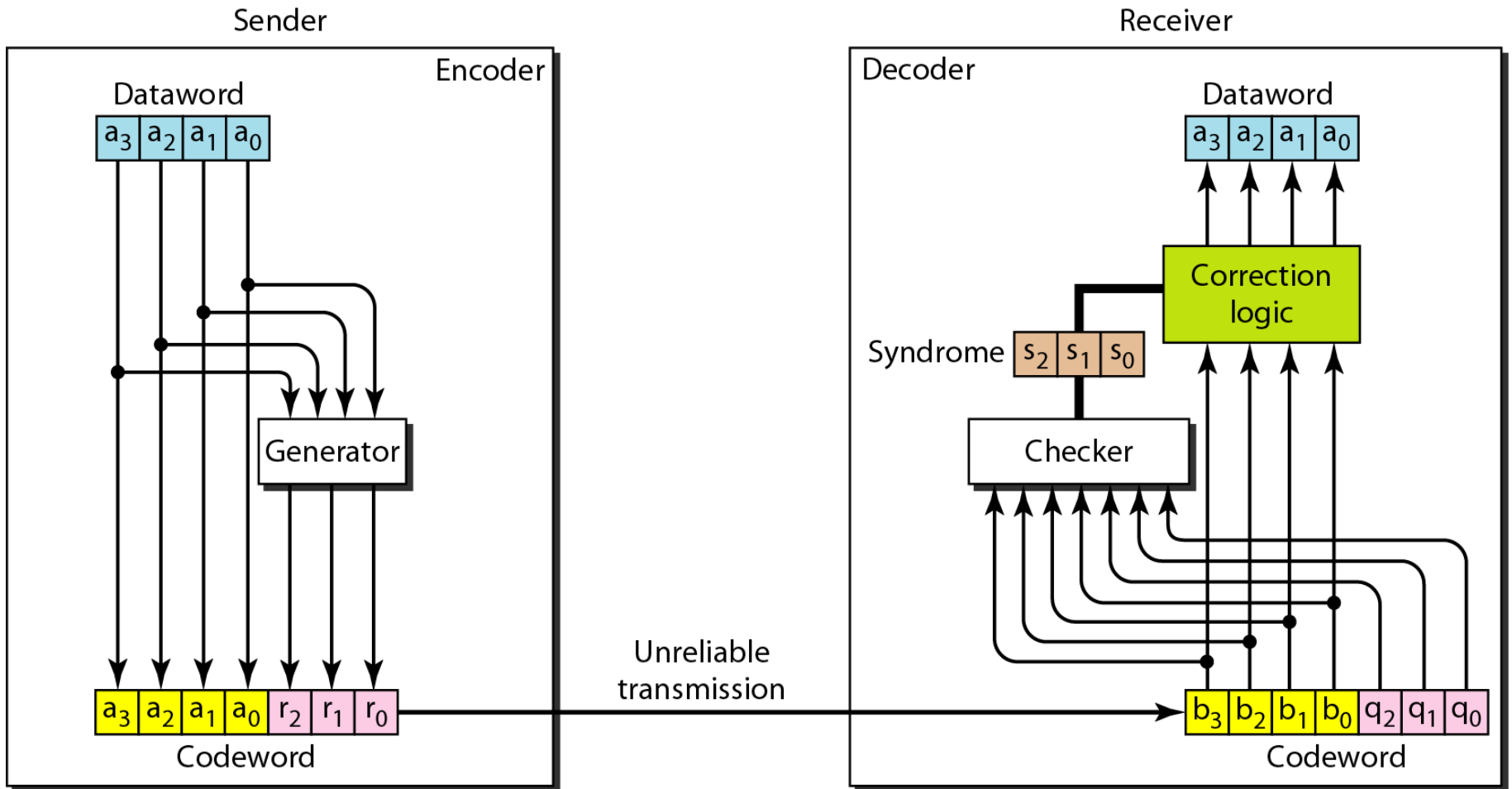
- Hamming distance = số lượng Bit khác nhau,
- $p = 011011$ ,  $q = 110001$ ,  $d(p,q) = ?$
- | <u>Data</u> | <u>Code</u> |
|-------------|-------------|
| 00          | 00000       |
| 01          | 00111       |
| 10          | 11001       |
| 11          | 11110       |
- Nhận được 00100, đúng? Làm sao có thể sửa?

# Block Code Principles

<i>Datawords</i>	<i>Codewords</i>	<i>Datawords</i>	<i>Codewords</i>
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

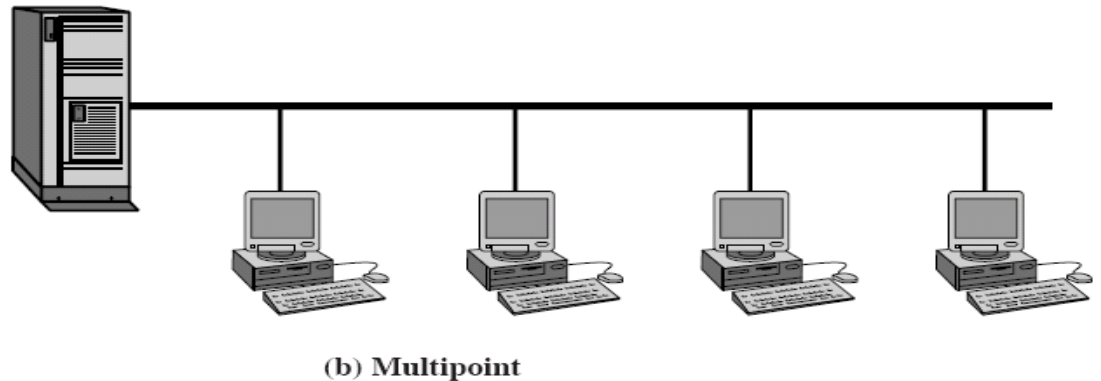
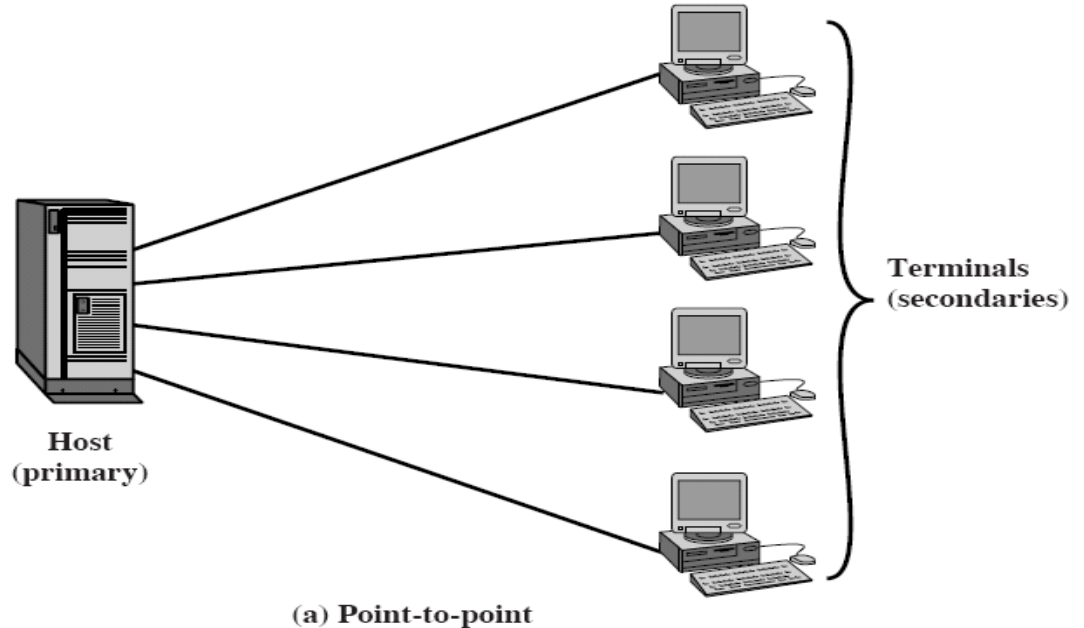


# Block Code



# Cấu hình đường truyền

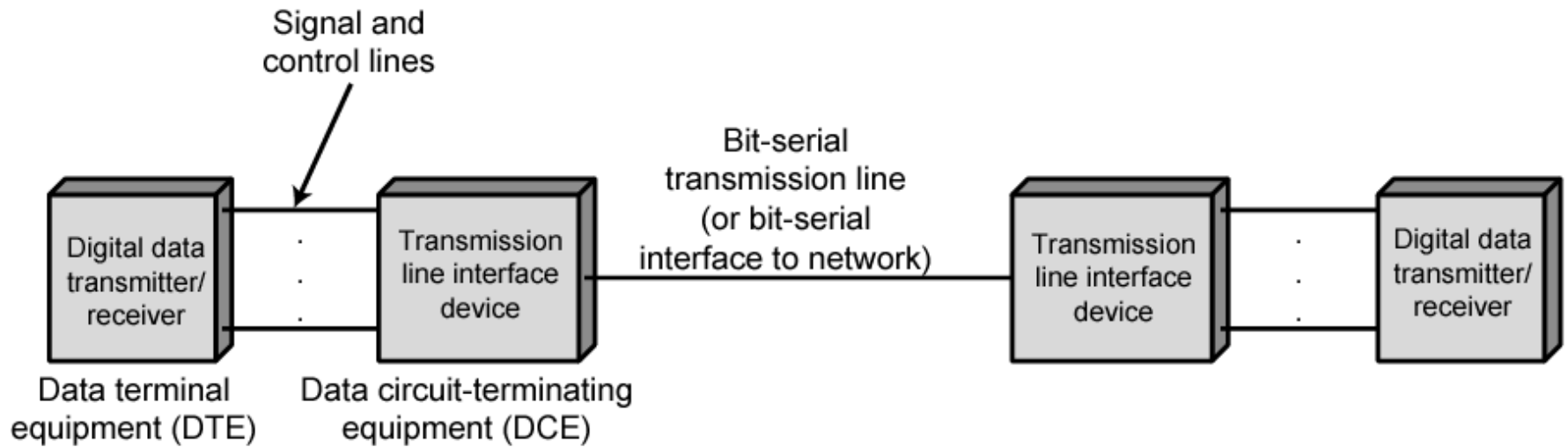
- Cấu hình: sắp xếp vật lý các trạm trên môi trường
- Cấu hình truyền thống



# Giao tiếp

- Thiết bị xử lý dữ liệu (DTE) thường không có các phương tiện phát dữ liệu
- Cần một thiết bị giao tiếp (DCE) – ví dụ: modem, NIC, ...
- DCE phát các bit dữ liệu trên môi trường truyền dẫn, DCE trao đổi dữ liệu và thông tin điều khiển với DTE
  - Được thực hiện thông qua mạch trao đổi
  - Cần một chuẩn giao tiếp rõ ràng

# Mô hình giao tiếp



(a) Generic interface to transmission medium



(b) Typical configuration

# Các đặc tả của giao tiếp

- Cơ học

- Các đầu nối

- Điện

- Điện áp, thời gian, mã

- Hoạt động

- Dữ liệu, điều khiển, định thời gian, tiếp đất

- Thủ tục

- Chuỗi liên tiếp các sự kiện

# Chuẩn V.24/EIA-232-F

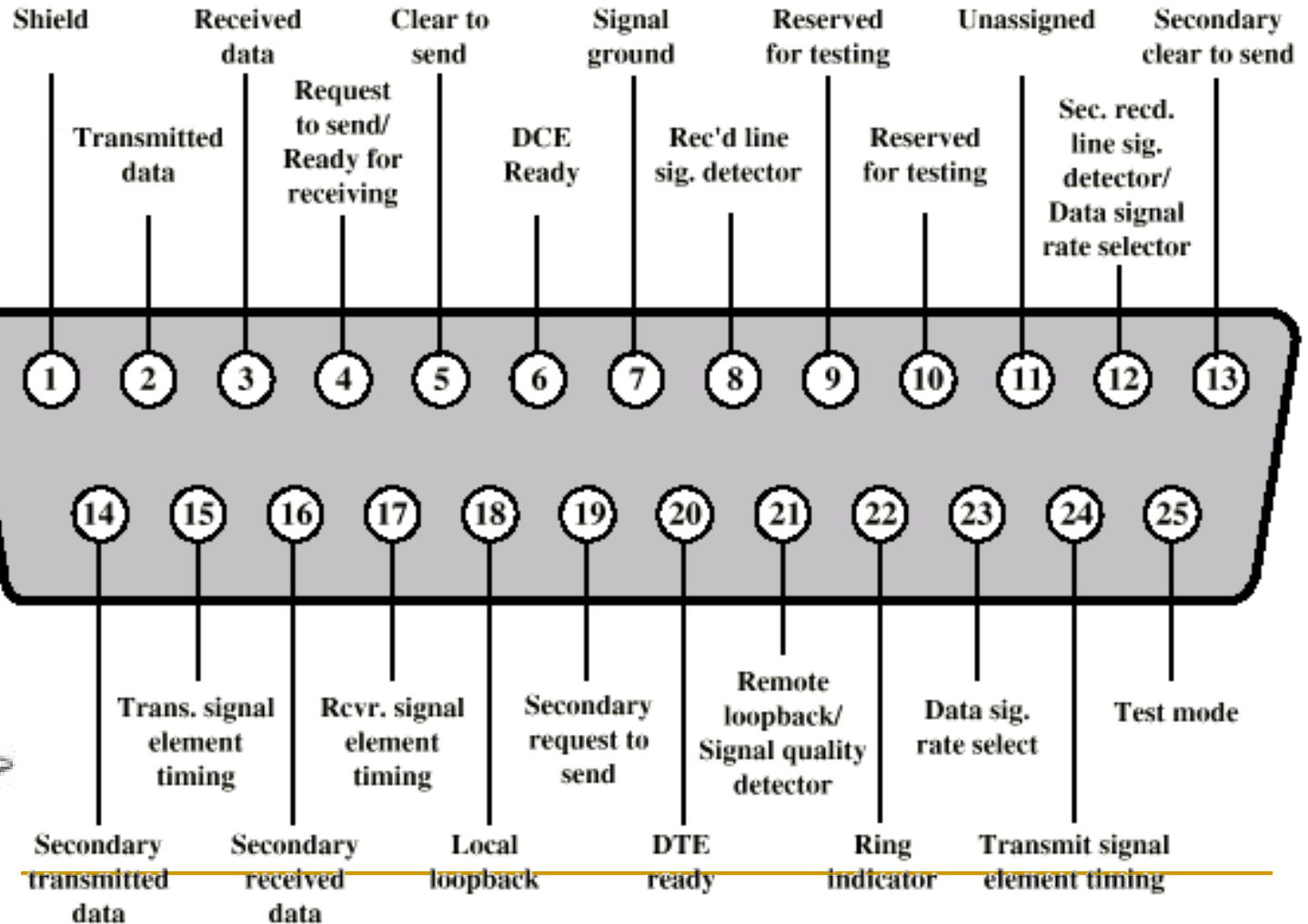
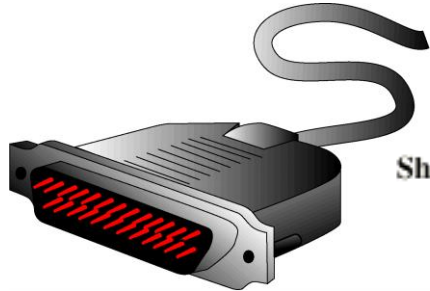
## ■ ITU-T v.24

- ❑ Chỉ đặc tả chức năng và thủ tục
- ❑ Tham khảo các chuẩn khác cho các đặc tính cơ khí và đặc tính điện

## ■ EIA-232-F (USA)

- ❑ RS-232
- ❑ Đặc tính cơ khí: ISO 2110
- ❑ Đặc tính điện: v.28
- ❑ Chức năng: v.24
- ❑ Thủ tục: v.24

# Kết nối V.24/EIA-232 (DTE)

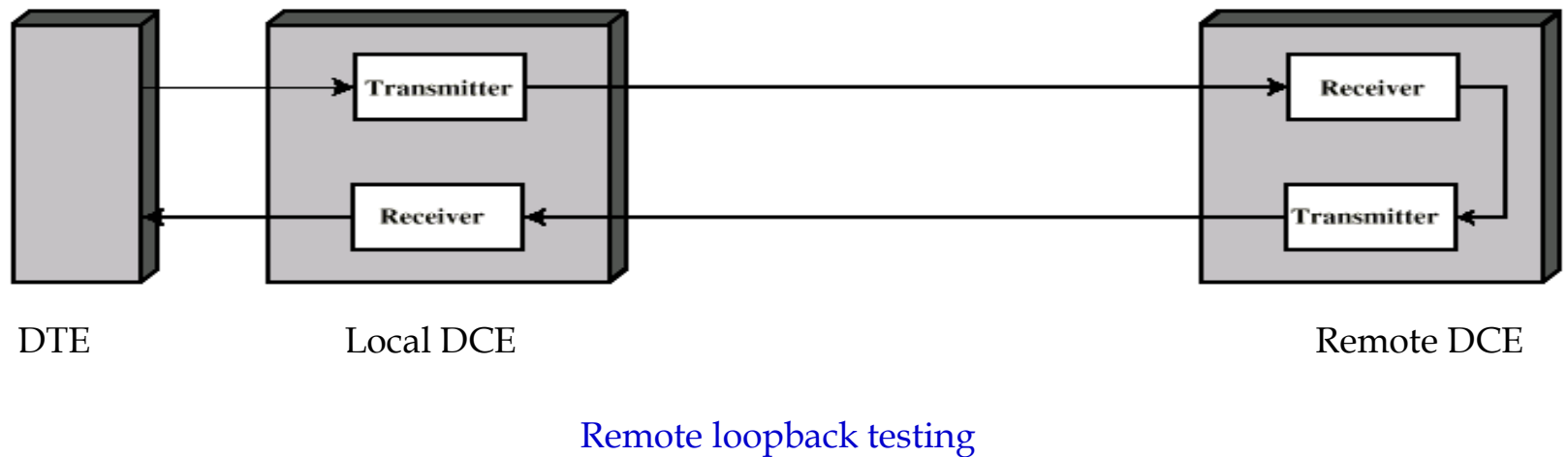
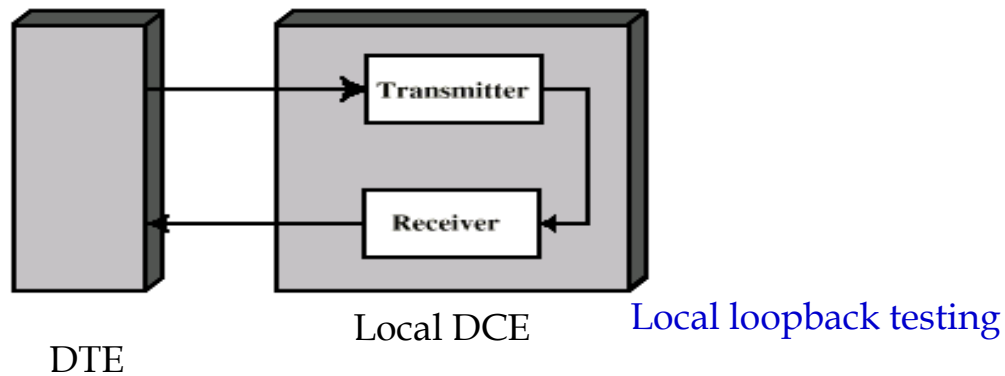


# V.24

V.24	EIA-232	Name	Direction to	Function
Data signals				
103	BA	Transmitted data	DCE	Transmitted by DTE
104	BB	Received data	DTE	Received by DTE
Control signals				
105	CA	Request to send	DCE	DTE wishes to transmit
106	CB	Clear to send	DTE	DCE is ready to receive, response to RTS
107	CC	DCE ready	DTE	DCE is ready to operate
108.2	CD	DTE ready	DCE	DTE is ready to operate
125	CE	Ring indicator	DTE	DCE is receiving a ringing signal on the channel line
109	CF	Received line signal detector	DTE	DCE is receiving a signal within appropriate limits on the channel line
Timing signals				
113	DA	Transmitter sig. elm. timing	DCE	Clocking signal
114	DB	Transmitter sig. elm. timing	DTE	Clocking signal;
115	DD	Receiver sig. elm. timing	DTE	Clocking signal for circuit 104
Ground				

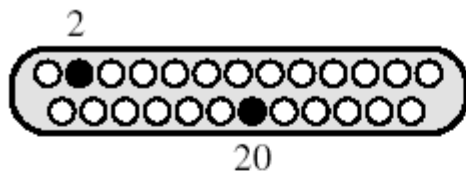


# Local/Remote loopback testing

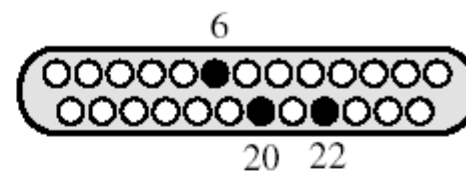
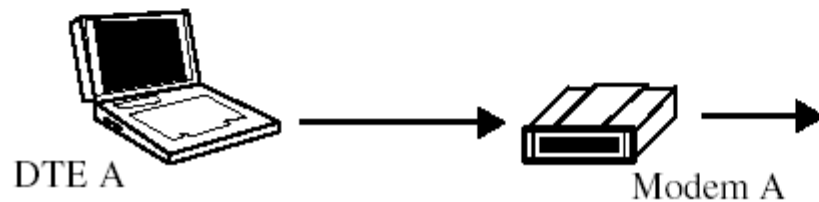


# Nghi thức

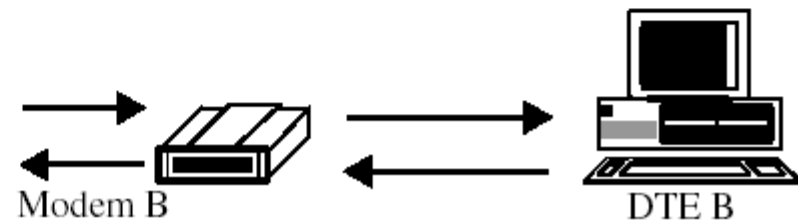
- Ví dụ modem riêng bất đồng bộ
- Khi modem được bật lên và sẵn sàng, nó (DCE) bật tín hiệu “DCE ready”
- Khi DTE sẵn sàng gửi dữ liệu, nó bật tín hiệu “Request To Send”
  - Cấm chế độ nhận dữ liệu (nếu trong chế độ truyền half-duplex)
- Modem đáp lại sẵn sàng bằng tín hiệu “Clear To Send”
- DTE gửi dữ liệu
- Khi dữ liệu đến, modem gắn vào DTE sẽ bật tín hiệu “Line Signal Detector” và gửi dữ liệu cho DTE



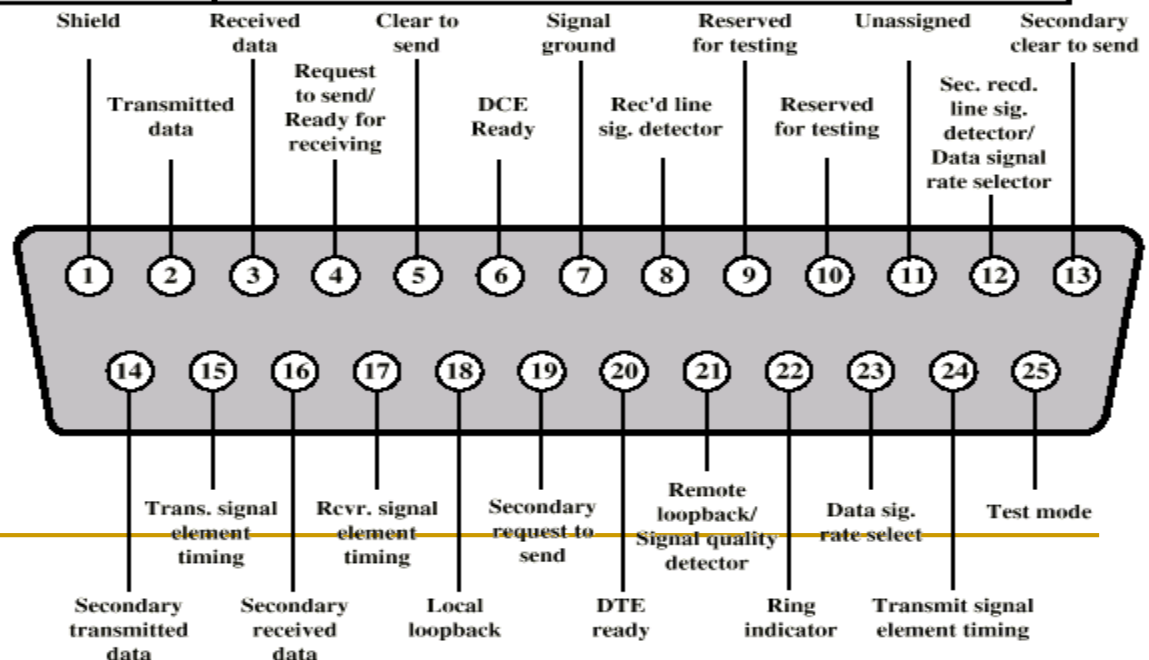
1. DTE A turns on the DTE ready pin (20) to tell its modem it wants to begin a data exchange. While this signal remains asserted, DTE A transmits a phone number via Transmitted Data (pin 2) for modem A to dial.

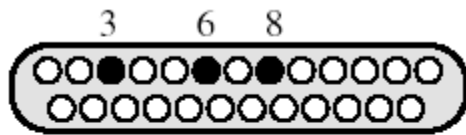


2. When modem B alerts its DTE to the incoming call via the Ring Indicator pin (22), DTE B turns on its DTE Ready pin (20). Modem B then generates a carrier signal, to be used in the exchange, and turns on pin 6, to show its readiness to receive data.

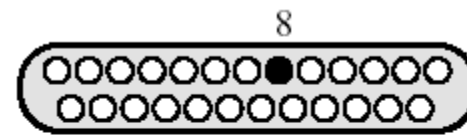


# Hoạt động quay số (1)

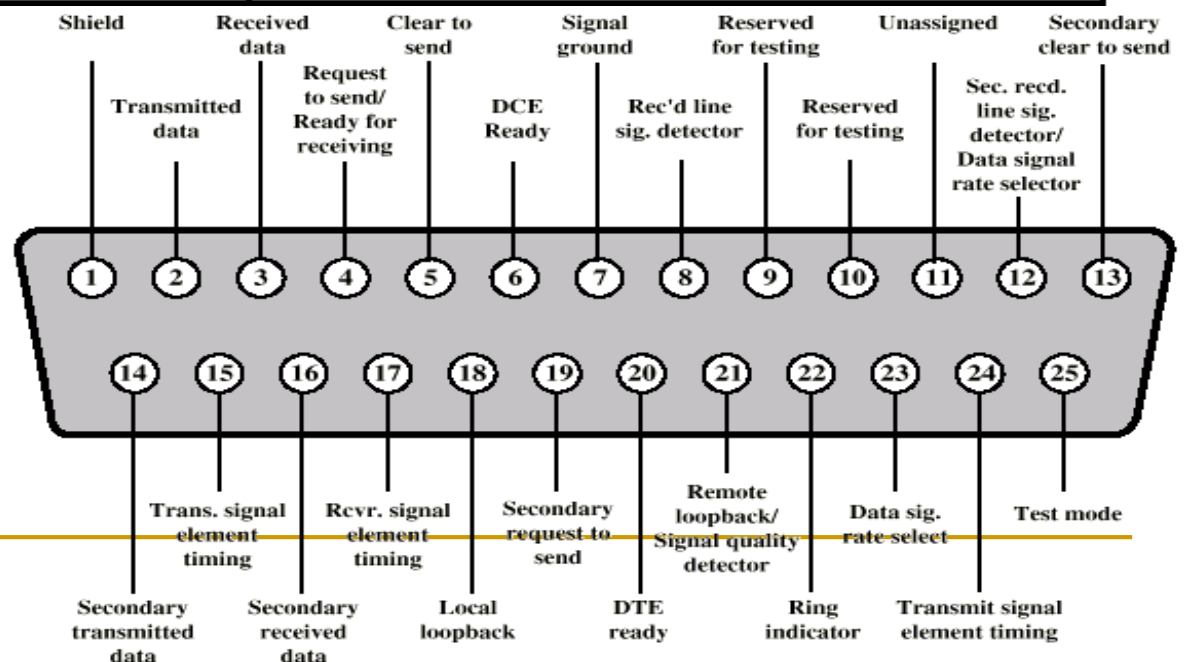
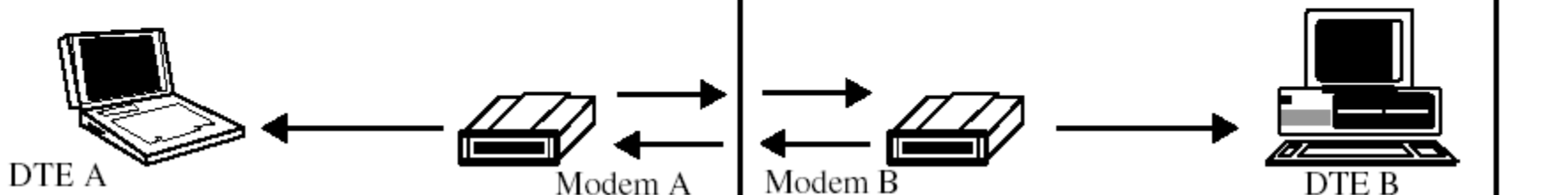




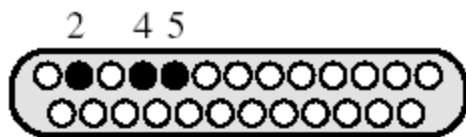
3. When modem A detects a carrier signal, it alerts DTE A via pin 8. The modem also tells the DTE that a circuit has been established (pin 6). If the modem has been so programmed, it will also send an "on line" message to the DTE's screen via the Received Data pin (3).



4. Modem A then generates its own carrier signal to modem B, which reports it via pin 8.



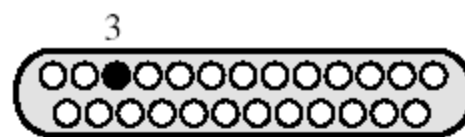
Hoạt động  
quay số (2)



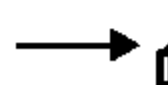
5. When it wishes to send data, DTE A activates Request to Send (pin 4). Modem A responds with Clear to Send (pin 5). DTE A sends data (pulses representing 1s and 0s) to modem A via the Transmitted Data pin (2). Modem A modulates the pulses to send the data over its analog carrier signal.



Modem A



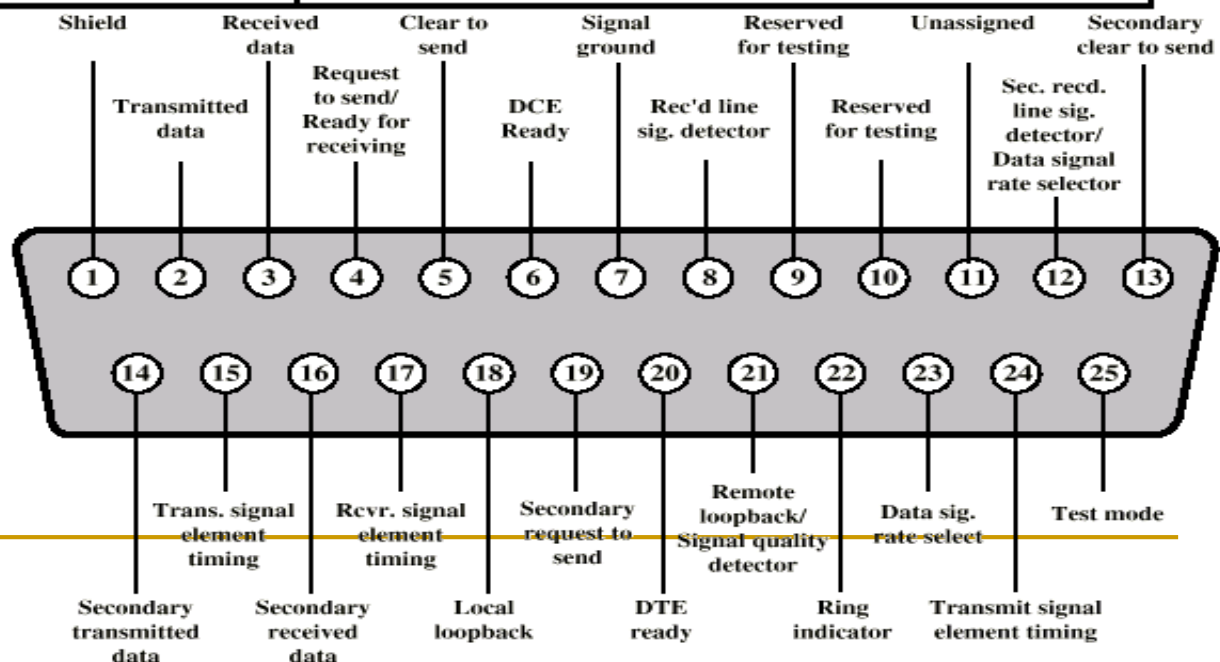
6. Modem B reconverts the signal to digital form and sends it to DTE B via the Received Data pin (3).



Modem B

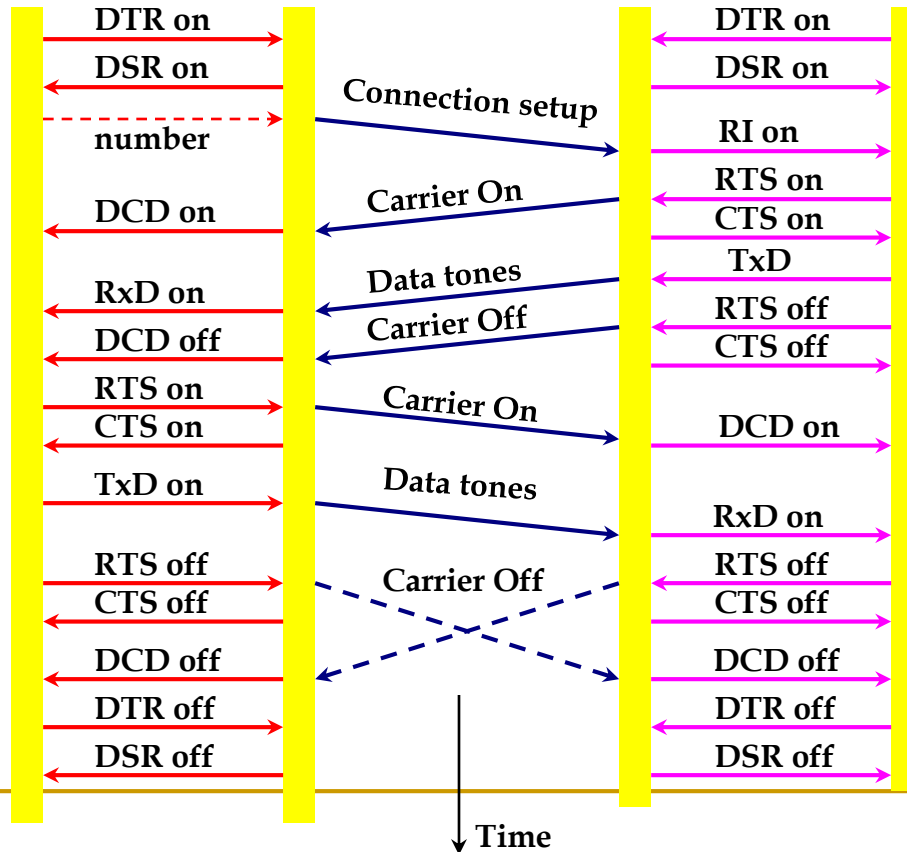
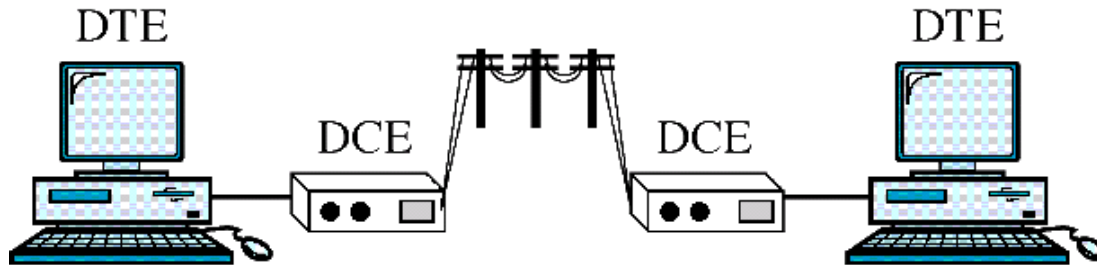


DTE B



Hoạt động  
quay số (3)

# Chuẩn giao tiếp EIA RS-232C



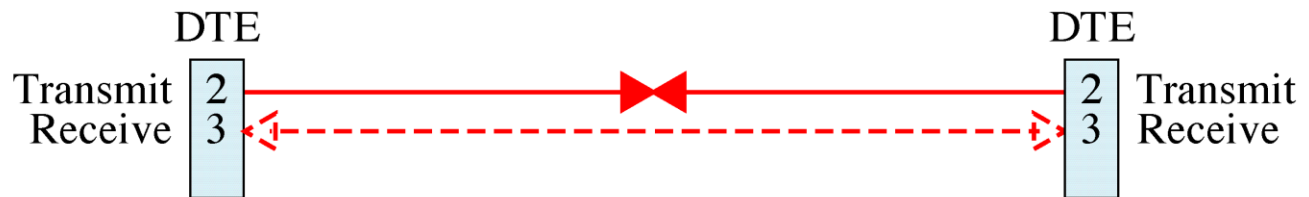
<b>DTR</b>	Data Terminal Ready
<b>DSR</b>	Data Set Ready
<b>DCD</b>	Data Carrier Detect
<b>RI</b>	Ring Indicator
<b>RTS</b>	Request To Send
<b>CTS</b>	Clear To Send
<b>TxD</b>	Transmitted Data
<b>RxD</b>	Received Data

# Trao đổi thông tin giữa DCE và DTE

- Trao đổi thông tin giữa DTE và DCE
  - Truyền dữ liệu (DTE→DCE)
    - Bật DTR và RTS
    - Đợi DSR
    - Đợi CTS
    - Truyền dữ liệu
  - Nhận dữ liệu (DCE→DTE)
    - Bật DTR
    - Đợi DSR
    - Nhận dữ liệu
- Trao đổi thông tin giữa 2 DTE
  - Không cần DCE
  - Null modem cable

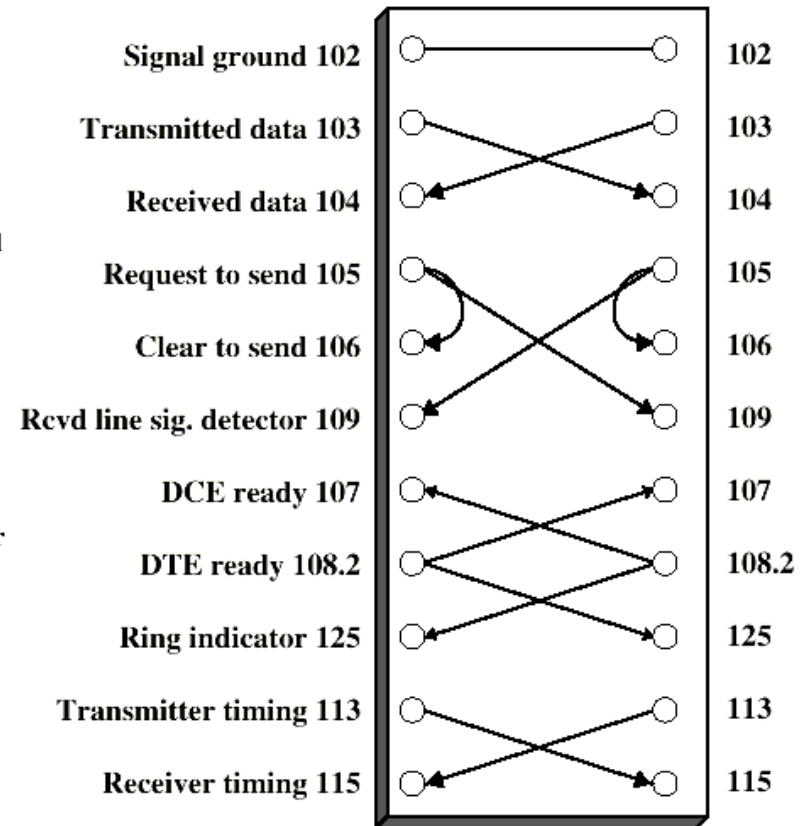
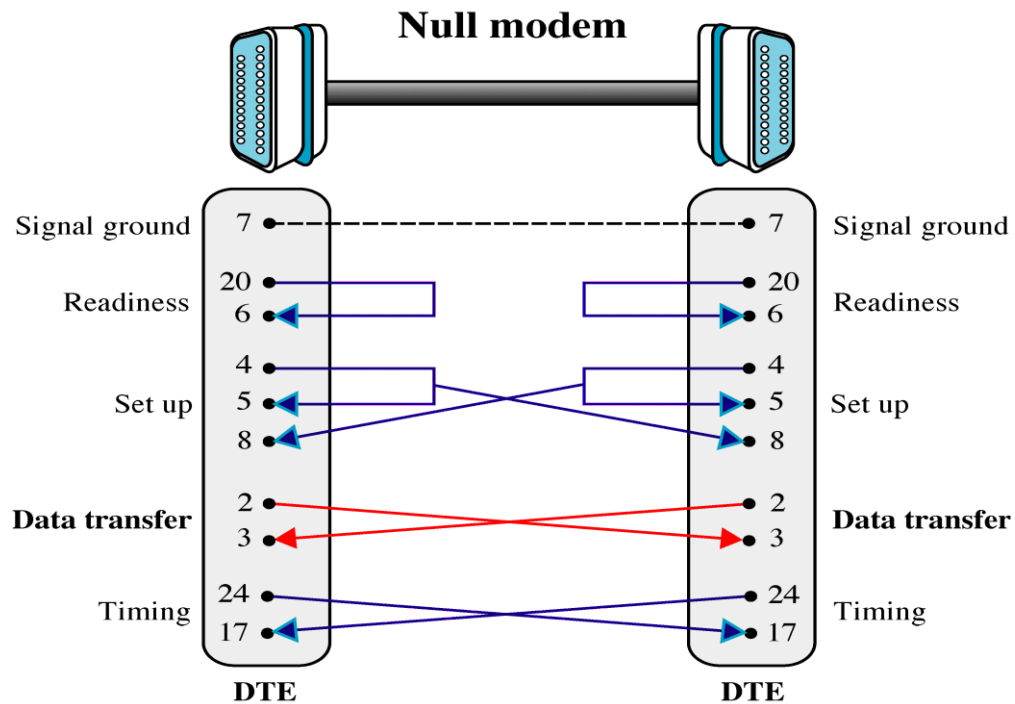


a. DTEs connected through DCEs



b. DTEs connected directly

# Cấu hình dây dẫn kết nối DTE ↔ DTE

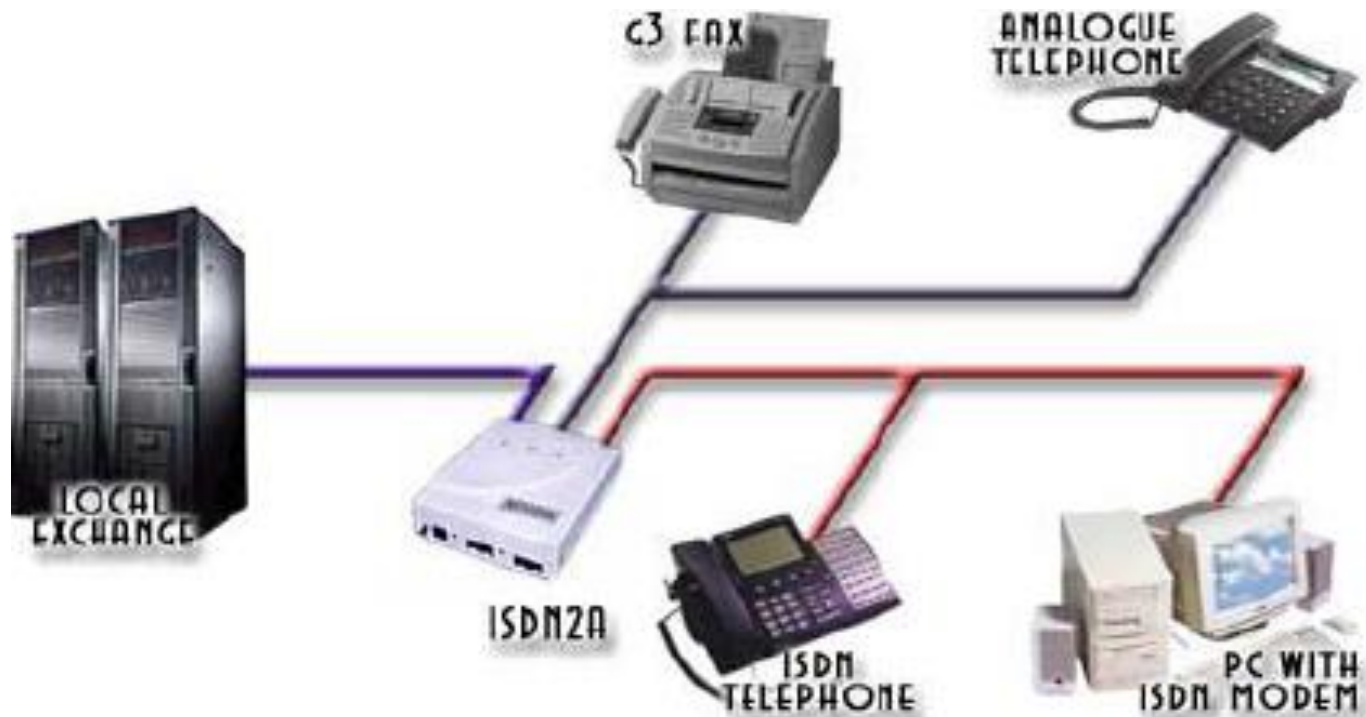




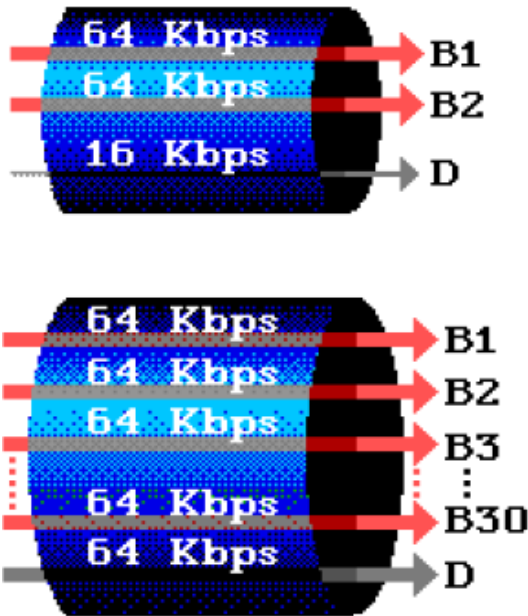
# Mạng ISDN

- Mạng số tích hợp dịch vụ ISDN (Integrated Services Digital Network) được định nghĩa như mạng thông tin có thể đầu nối theo công nghệ số (digital) từ thuê bao chủ gọi và xử lý tất cả các loại dịch vụ thoại và phi thoại.
- Mạng ISDN cho phép tất cả các thông tin thoại (phone), số liệu (data) và hình ảnh (video) có thể truyền qua một đường dây thuê bao (subscriber line) với tốc độ cao và chất lượng tốt.
- ISDN truyền bằng tín hiệu điện tử (digital) qua dây đồng (dây điện thoại) và một số loại vật liệu khác với tốc độ lên tới 128Kbps.

# Cấu hình mạng ISDN

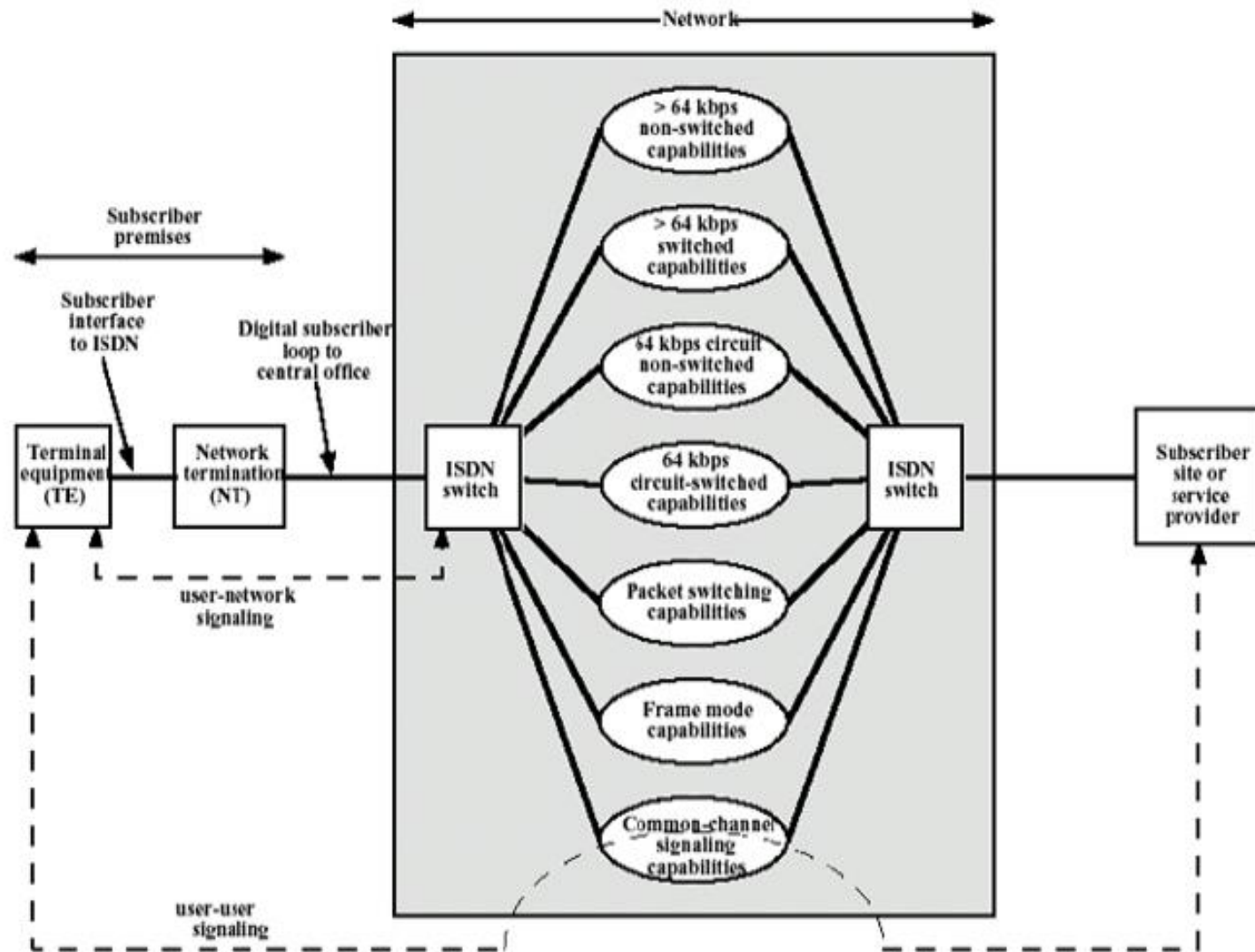


# Kênh trong ISDN

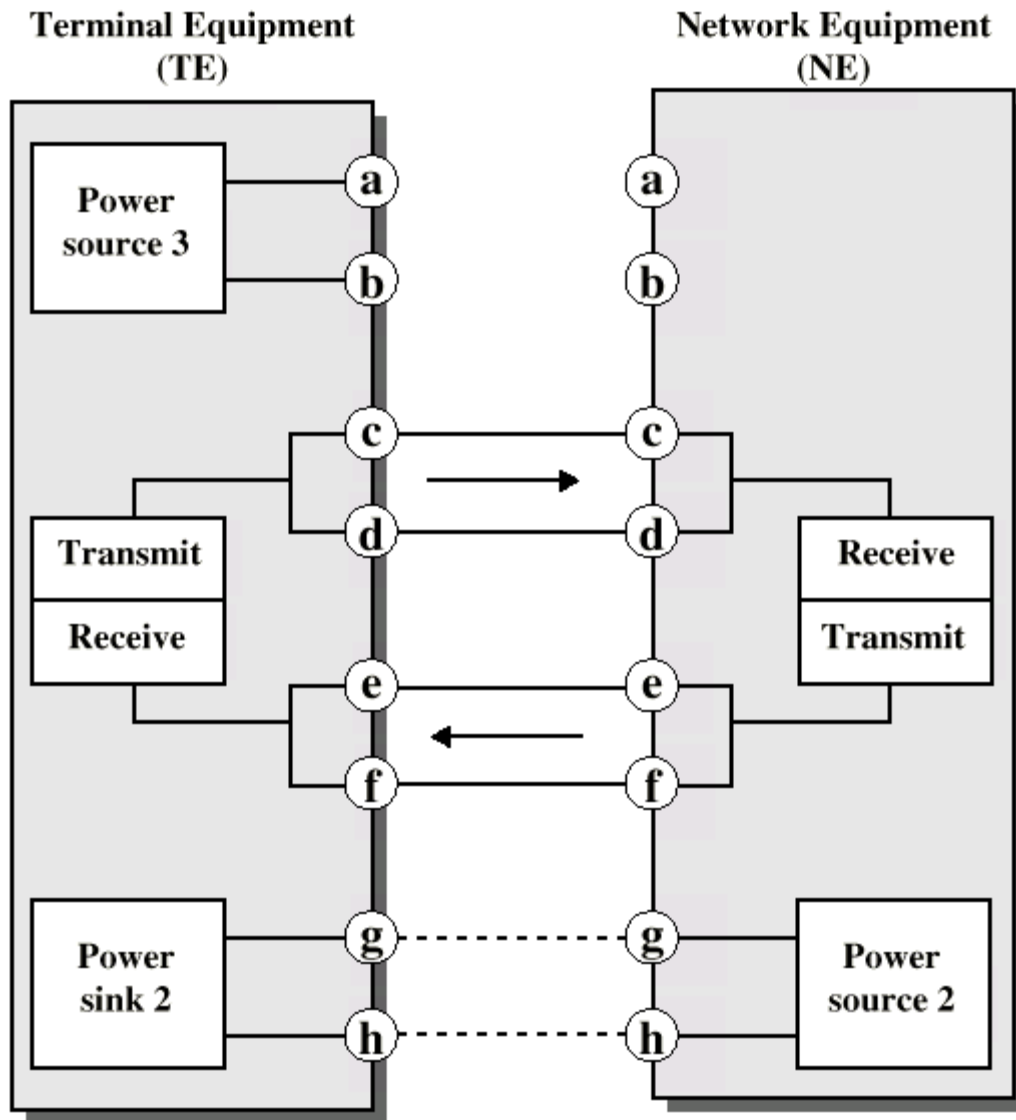


	BRI	T1 PRI	E1 PRI
B-Channels	2x64 KBPS	23x64 KBPS	30x64 KBPS
D-Channels	1x16 KBPS	1x64 KBPS	1x64 KBPS
Synchronization	16 KBPS	8 KBPS	64 KBPS
Total Data Rate	160 KBPS	1.544 MBPS	2.048 MBPS
Line Coding	2B1Q / 4B3T	AMI / B8ZS	HDB3

# Kiến trúc của ISDN



# ISDN



---

HẾT CHƯƠNG 4

---