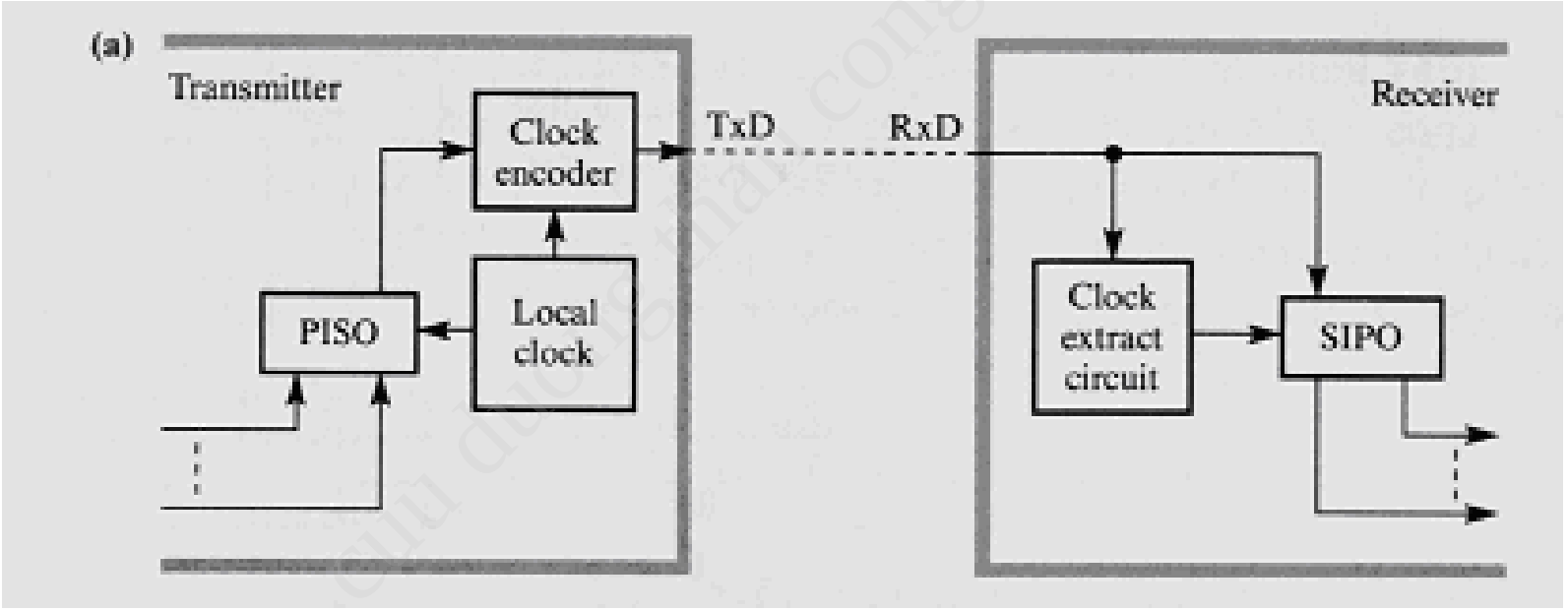
Truyền đồng bộ (Synchronous transmission)

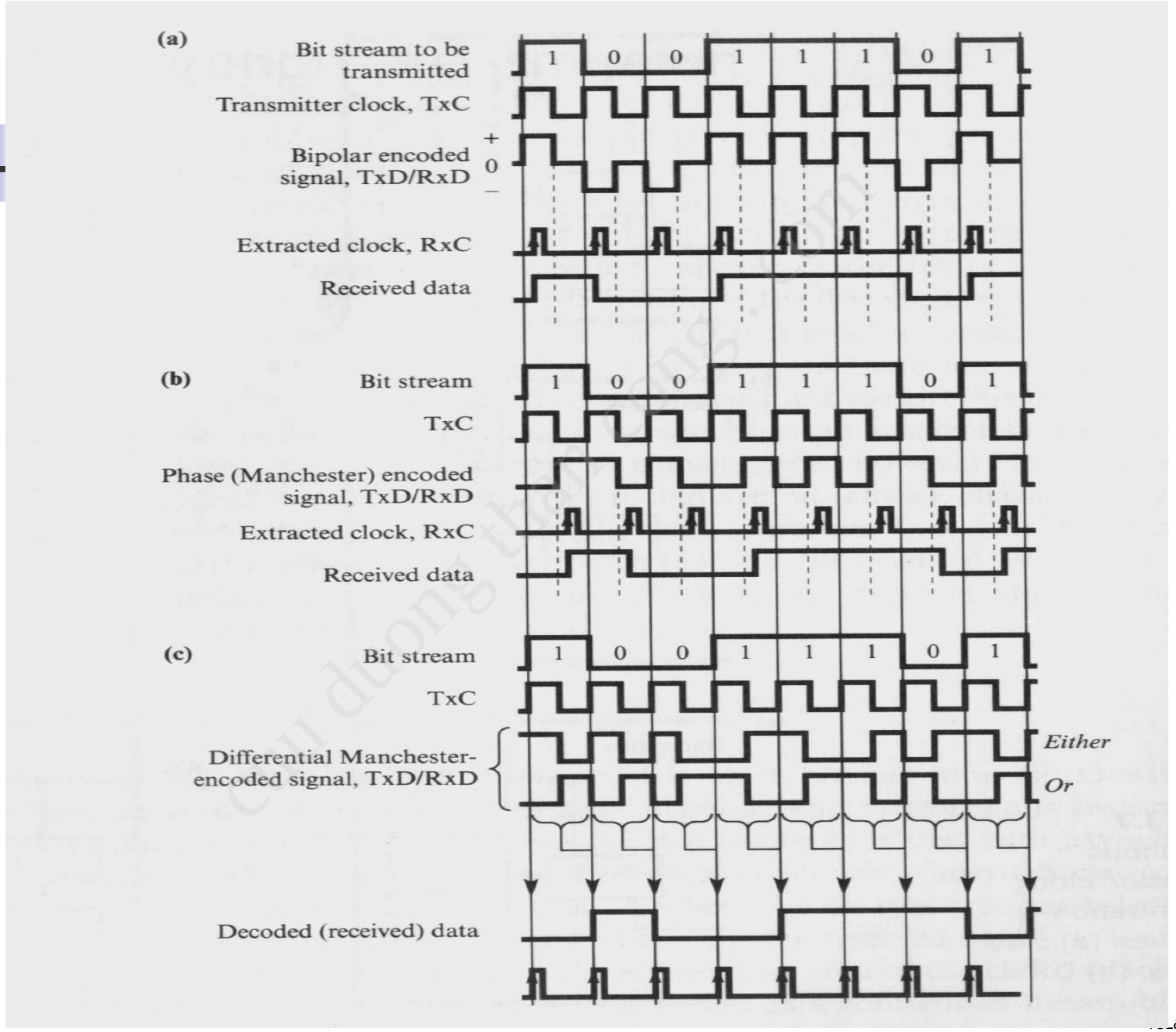
- Kỹ thuật đồng bộ trong kiểu truyền đồng bộ
  - Đồng bộ bit :
    - Clock encoding and extraction
    - Digital Phase-lock-loop (DPLL)
    - Hybrid
  - Đồng bộ khung :
    - Character-oriented
    - Bit-oriented

Truyền đồng bộ (Synchronous transmission)

- Đồng bộ bit :
  - Clock encoding and extraction
    - Phía phát gửi xung clock vào tín hiệu phát bằng cách mã hóa dữ liệu trước khi phát thông qua mạch Clocl Encoder. Phía thu sẽ trích tín hiệu clock từ tín hiệu nhận được nhờ mạch Clock Extract Circuit.
    - Mã đường dây : RZ, Manchester, differential Manchester

Truyền đồng bộ (Synchronous transmission)

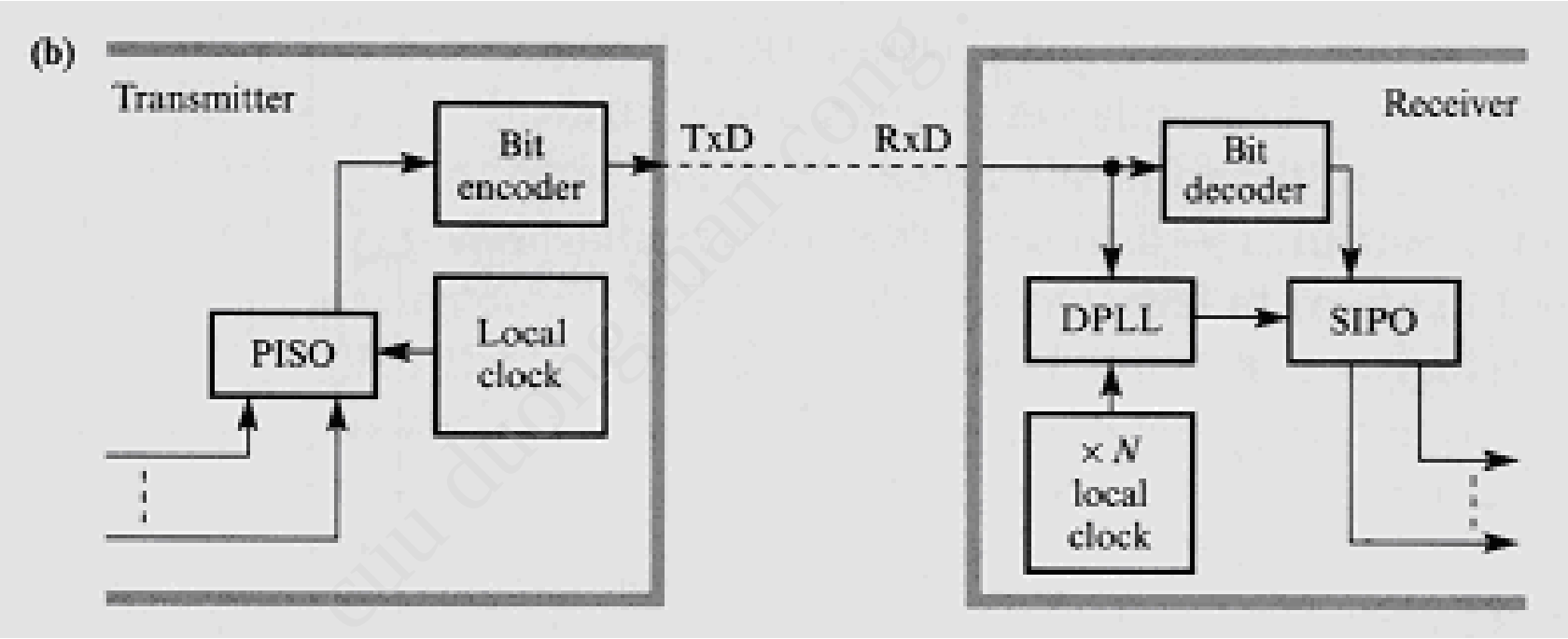


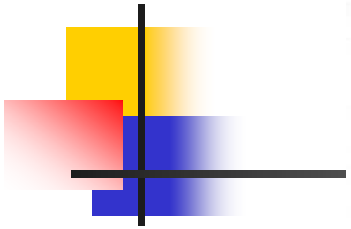


Truyền đồng bộ (Synchronous transmission)

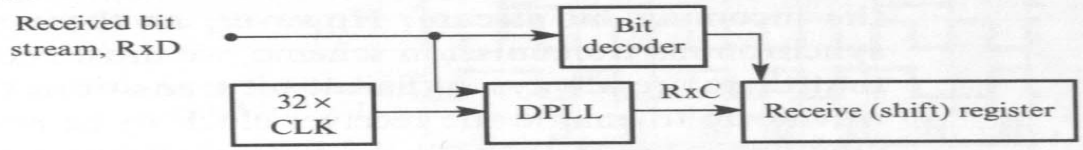
- Digital Phase-lock-loop (DPLL):
  - Bộ thu đồng bộ với bộ phát nhờ vào vòng khóa pha số. Phía thu sử dụng đồng hồ có tần số gấp N lần phía phát cấp cho PLL. PLL có nhiệm vụ tạo tín hiệu clock cho thanh ghi SIPO từ tín hiệu đồng hồ và tín hiệu nhận được sao cho đúng giữa chu kỳ bit.
  - Để clock thu duy trì được sự đồng bộ với clock phát thì chuỗi dữ liệu phát phải được mã hoá để có đủ sự thay đổi trạng thái ($1 \rightarrow 0$ hay $0 \rightarrow 1$).
  - Mã đường dây : NRZ, AMI, HDB3, B3ZS, B6ZS, B8ZS, 4B3T, 2B1Q

Truyền đồng bộ (Synchronous transmission)

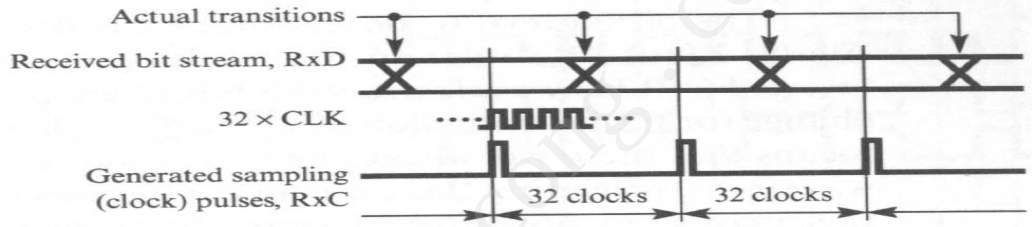




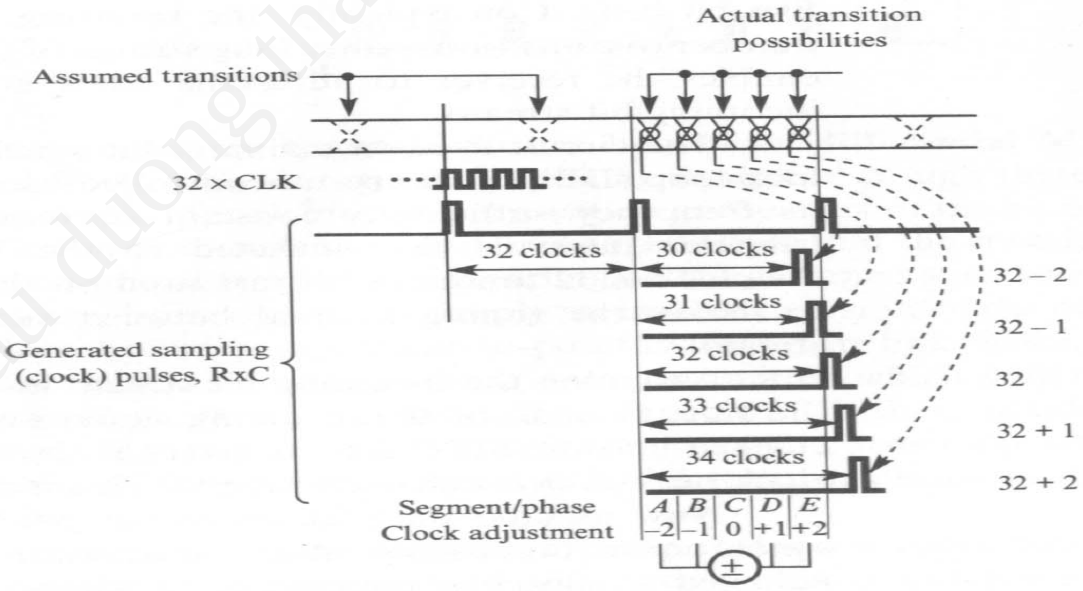
(b)



(c)



(d)



Truyền đồng bộ (Synchronous transmission)

- Thông thường clock thu có tần số gấp N=32 lần tần số clock phát. Bộ tạo dao động này được nối tới DPLL nhằm duy trì sự đồng bộ.
- DPLL là một bộ phận được sử dụng để duy trì sự đồng bộ bit giữa bộ tạo xung clock thu với chuỗi dữ liệu thu vào. Việc duy trì sự đồng bộ này được dựa trên sự thay đổi trạng thái trong chuỗi dữ liệu thu được.
- Trong trường hợp clock thu và chuỗi dữ liệu thu duy trì được sự đồng bộ với chuỗi dữ liệu thu vào (hình 3.3.3 c), bit dữ liệu thu sẽ được lấy mẫu ngay tại vị trí giữa chu kỳ bit sau mỗi 32 xung clock.

Truyền đồng bộ (Synchronous transmission)

- Trong trường hợp Clock thu và chuỗi dữ liệu thu không đồng bộ thì xung lấy mẫu sẽ được hiệu chỉnh trong vòng từ 30 đến 34 xung Clock
- Nếu trong một khoảng thời gian dài không có trạng thái chuyển đổi, DPLL sẽ phát ra một xung lấy mẫu sau mỗi 32 chu kỳ clock. Khi đó, phía thu có thể không duy trì được sự đồng bộ với chuỗi dữ liệu thu vào (hình 3.3.3 d). Khi phát hiện được trạng thái chuyển đổi trên đường dây, DPLL sẽ so sánh nó với thời điểm dịch chuyển giả sử, vàhiệu chỉnh xung lấy mẫu tương ứng với sự chênh lệch này. Qúa trình này được thực hiện như sau :

Truyền đồng bộ (Synchronous transmission)

- 1 chu kỳ bit chia thành 5 đoạn A, B,C,D,E nhu hình vẽ.
- Sự chênh lệch vị trí chuyển mức có thể xảy ra trong các đoạn A, B, C, D, E (hình 3.3.3 d).
- Nếu vị trí chuyển đổi xảy ra trong đoạn A, thì vị trí xung lấy mẫu cuối cùng trước đó rất gần với sự chuyển đổi trạng thái kế nó, nghĩa là vị trí lấy mẫu bị trễ (tốc độ lấy mẫu chậm). Do đó DPLL sẽ hiệu chỉnh bằng cách rút ngắn khoảng thời gian lấy mẫu xuống còn 32 – 2 = 30 clock.
- Ngược lại, nếu vị trí chuyển đổi xảy ra như trong trường hợp E, thì vị trí xung lấy mẫu cuối cùng trước đó rất xa với sự chuyển đổi trạng thái kế nó, nghĩa là vị trí lấy mẫu bị sớm (tốc độ lấy mẫu nhanh). Do đó DPLL sẽ hiệu chỉnh bằng cách kéo dài khoảng thời gian lấy mẫu lên 32+2 =34 clock.

Truyền đồng bộ (Synchronous transmission)

- Tương tự trong trường hợp B hoặc D, vị trí lấy mẫu trễ và sớm ít hơn so với A hoặc E, do đó DPLL sẽ hiệu chỉnh khoảng thời gian lấy mẫu tương ứng sẽ là 32-1=31 hoặc 32+1 = 33 clock.
- Trong trường hợp C, vị trí xung clock DPLL giả sử nó xảy ra trùng với vị trí chuyển đổi trạng thái thực, sự đồng bộ được duy trì, do đó không cần phải hiệu chỉnh. (khoảng lấy mẫu vẫn là 32 xung clock).

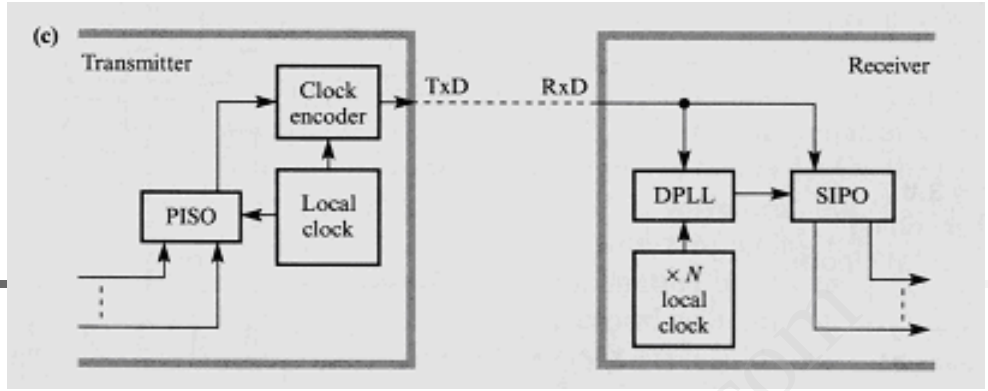
Truyền đồng bộ (Synchronous transmission)

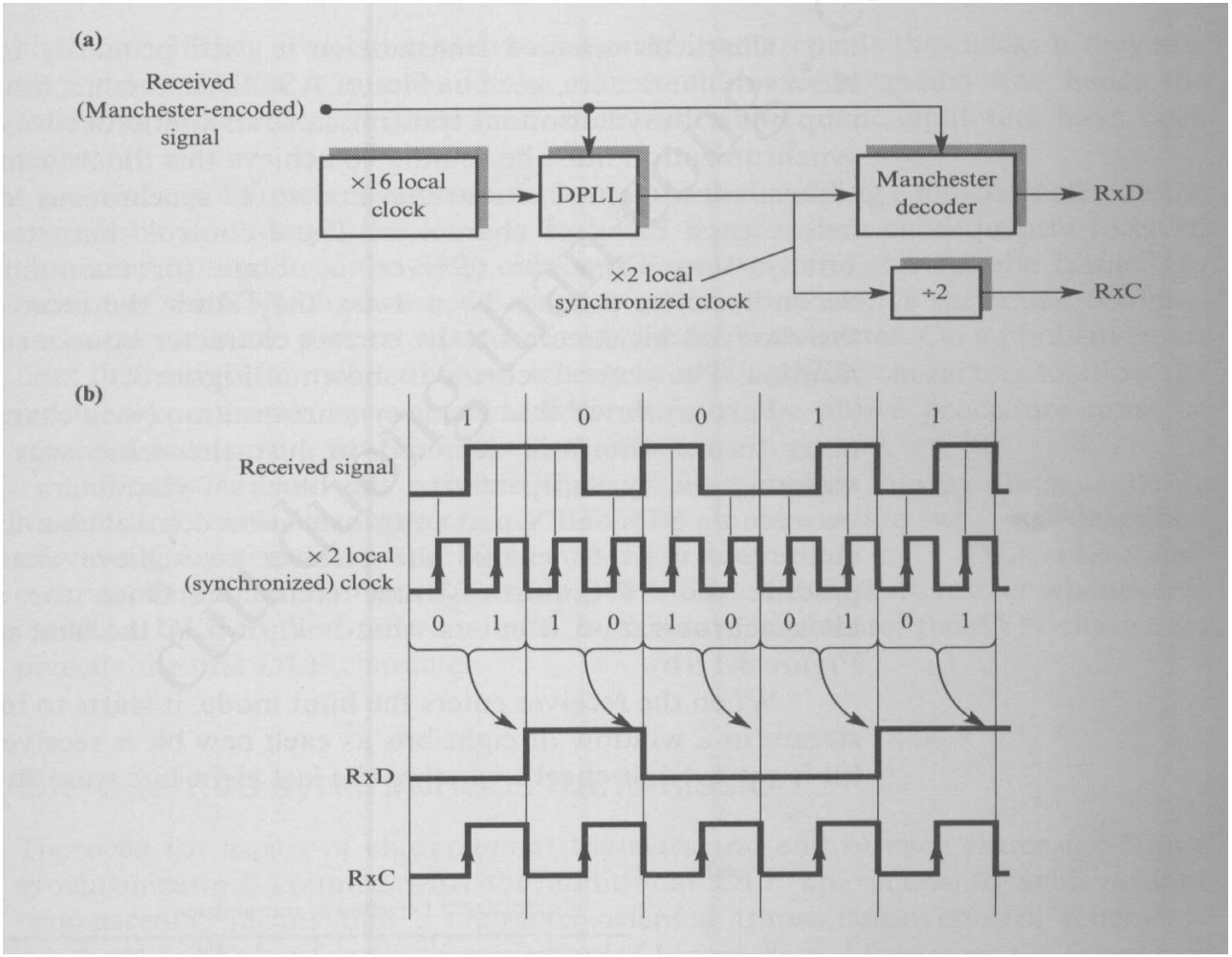
- Bằng cách hiệu chỉnh như trên, sẽ tạo ra xung lấy mẫu càng lúc càng gần trung tâm của bit dữ liệu.
- Như vậy độ rộng của 1 bit tương đương với 32 xung clock. Vùng A.B xa nhất nên gán A=E=10 clock, B=C=D= 4 clock.
- Số bit dữ liệu tối đa để xung lấy mẫu bộ thu đồng bộ với dữ liệu nhận được được tính như sau: Khảng A hoặc E mỗi lần hiệu chỉnh 2 nhịp clock (±2) nên để thoát ra khỏi vùng A hoặc E cần 5 bit dữ liệu cho sự hiệu chỉnh thô (vì đoạn này chiếm 10 xung clock) và tương tự để thoát ra vùng B hoặc D thì cần 4 bit dữ liệu, và cuối cùng để đảm bảo lấy chính xác tại trung tâm mỗi bít thì cần 1 bit dữ liệu nữa. Vậy tổng cộng cần 10 bit dữ liệu.

->Do đó thường trong kỹ thuật truyền đồng bộ thì các bít đồng bộ thường được phát trước khi tuyền dữ liệu thực sự, để đảm bảo đồng bộ bên phát và bên thu không ảnh hưởng đến thông tin cần truyền.

Truyền đồng bộ (Synchronous transmission)

- Hybrid :
  - Khi tốc độ bit tăng thì các phương pháp trên rất khó thực hiện đồng bộ. Để giải quyết vấn đề này ta sử dụng phương pháp Hybrid.
  - Đây là phương pháp kết hợp 2 phương pháp Clock encoder và DPLL. Clock encoder đảm bảo các bit khi nhận được có ít nhất 1 sự xáo trộn trong chu kỳ 1 bit, trong khi đó DPLL được dùng để giữ nhịp nội tại đồng bộ với dữ liệu nhận.
  - Khuyết điểm : Sử dụng băng thông lớn.
  - Mã đường dây : Manscheter



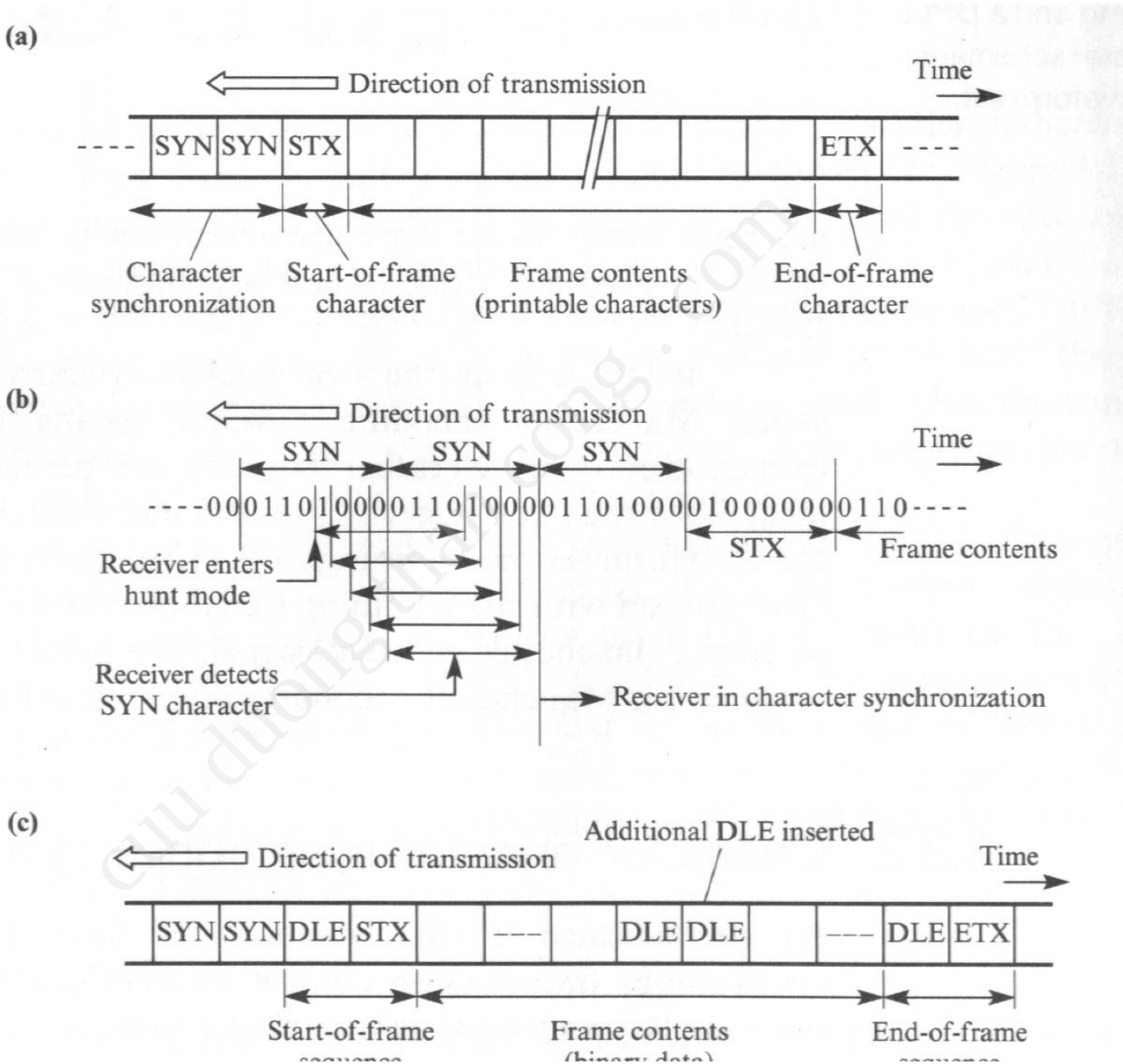


Truyền đồng bộ (Synchronous transmission)

- Đồng bộ khung :
  - Character-oriented :
    - Thường sử dụng khi truyền các khối ký tự.
    - Để thực hiện việc đồng bộ ký tự, bộ phát sẽ truyền trước ít nhất là 2 ký tự điều khiển (control characters) còn gọi là ký tự đồng bộ SYN trước khi truyền khối ký tự. Điều này sẽ thực hiện 2 chức năng:
      - Đồng bộ bit: tạo ra các trạng thái chuyển đổi mức tín hiệu trên đường truyền để DPLL thiết lập được sự đồng bộ.
      - Đồng bộ ký tự: cho phép phía thu xác định chính xác vị trí bắt đầu và kết thúc của mỗi ký tự.

Truyền đồng bộ (Synchronous transmission)

- Khi phía thu thực hiện được việc đồng bộ bit, nó sẽ bắt đầu chế độ dò tìm (hunt mode). Trong mode này phía thu sẽ thu và kiểm tra mỗi nhóm 8 bit xem có phải là ký tự đồng bộ (SYN) hay không. Nếu không phải là ký tự SYN, phía thu sẽ thu bit kế tiếp và kiểm tra. Ngược lại nếu đúng là SYN thì phía thu xem như đã thực hiện xong việc đồng bộ ký tự, và sau đó nhận vào 8 bit xem như 1 ký tự.
- Sau khi đã thực hiện xong vấn đề đồng bộ như trên Việc đồng bộ khung được thực hiện giống như kỹ thuật truyền bất đồng bộ.



Truyền đồng bộ (Synchronous transmission)

- Nhận xét :
  - Kiểu truyền định hướng ký tự này sử dụng nhiều ký tự điều khiển (STX, ETX, DLE), do đó hiệu suất truyền thấp.
  - Trong kiểu truyền này yêu cầu khối dữ liệu phát phải có chiều dài là bội số của 8 đảm bảo hệ thống xử lý theo từng ký tự (định hướng ký tự). Điều này có thể không được đảm bảo nếu khối ký tự phát là dữ liệu nhị phân bất kỳ.

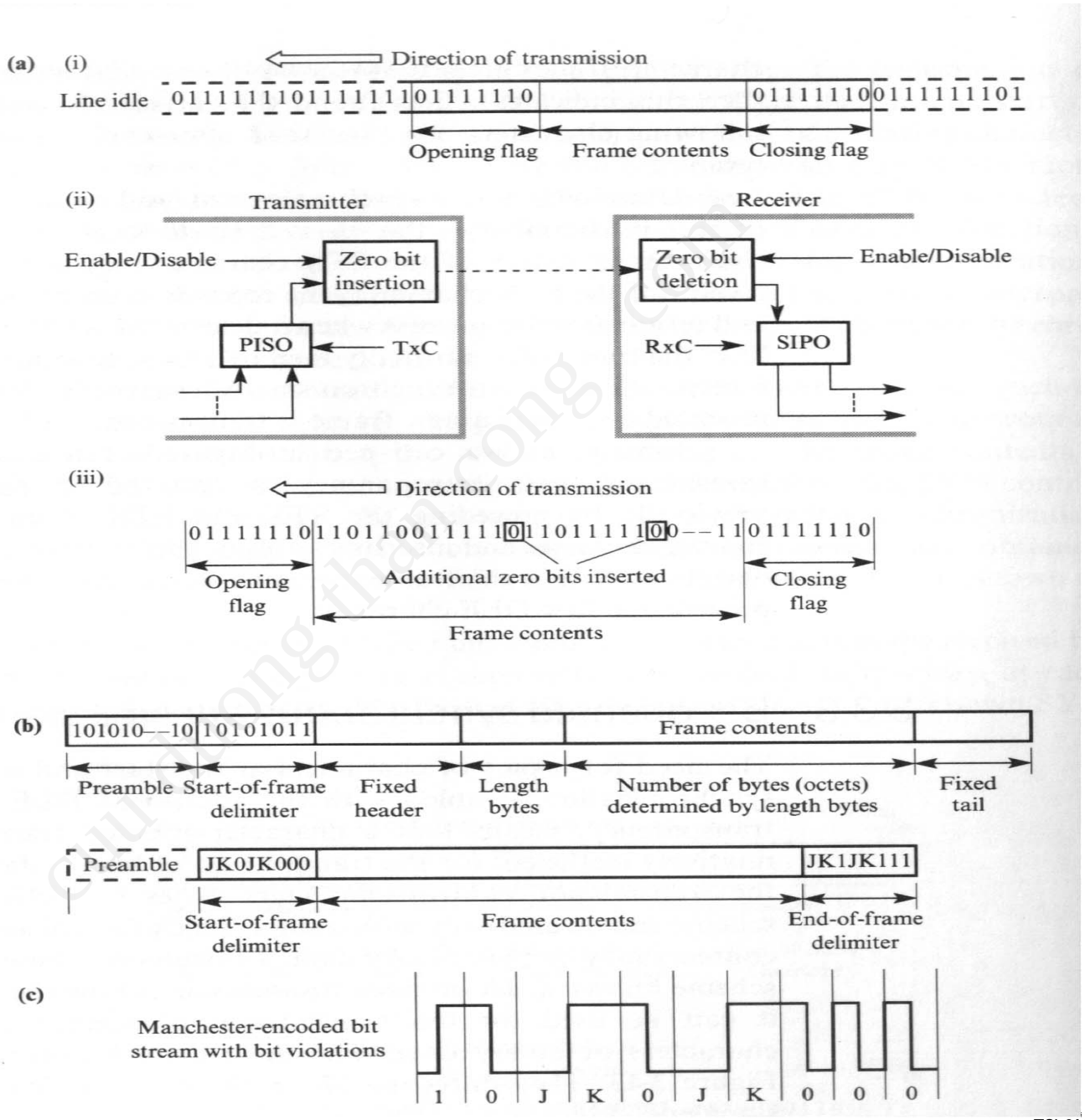
Truyền đồng bộ (Synchronous transmission)

- Bit-oriented : Điểm - điểm
  - Bắt đầu và kết thúc khung (start and end of frame) truyền chuỗi 8 bit 01111110 gọi là cờ (flag pattern).
  - Phía thu sẽthực hiện việc đồng bộ khung bằng cách tìm ký tự (chuỗi bit) cờ này theo nguyên tắc tìm từng bit (bit by bit basic).
  - Khi phía thu nhận được opening flag (cờ bắt đầu), phía thu sẽ bắt đầu nhận khung dữ liệu cho tới khi phát hiện được closing flag (cờ kết thúc), khi đó việc thu khung dữ liệu kết thúc.
  - Để thực hiện đồng bộ bit, phía phát sẽ gửi các byte rảnh (idle bytes :11111111) trước cờ khởi đầu của khung.(Chú ý: Các bit 1 được dùng mã đường dây nên sẽ có sự xáo trộn mức để bên thu thực hiện đồng bộ
  - Việc trong suốt dữ liệu được thực hiện bằng cách chèn bit 0 (zero bit insertion) thực hiện tại phía phát. Khi hoạt động, khối này sẽ kiểm tra xem trong chuỗi bit phát có chuỗi liên tiếp 5 bit 1 hay không, nếu có thì sẽ thực hiện chèn 1 bit 0 vào cuối chuỗi này. Khi đó trong chuỗi dữ liệu phát sẽ không thể có ký tự cờ 01111110. Phía thu sẽ thực hiện ngược lại, nếu thu được chuỗi dữ liệu bao gồm 5 bit 1 liên tiếp, sau đó là bit 0 thì nó sẽ loại bỏ bit 0 này bằng mạch zero bit deletion (mạch xoá bit 0).

Truyền đồng bộ (Synchronous transmission)

- Bit-oriented : Đa điểm
  - Trong cấu hình mạng đa điểm như mạng LAN thì phương pháp đồng bộ Bit Oriented có thể được thực hiện theo nhiều cách khác nhau như :
    - Để tất cả các trạm bám theo đồng bộ thì phát mẫu bit gọi là Preamble : 1010101010
    - Để xác định vị trí bắt đầu và kết thúc một frame thì dùng những mẫu bit như sau (Tùy theo cấu trúc) :
      - Start of frame delimiter : 10101011, hoặc
      - Start of frame :JK0JK000. End of frame : JK1JK100

      Trong đó J,K là những mẫu bit được mã hóa không đúng chuẩn với dòng bit truyền thực sự.



Vi du

Một tập tin nhị phân sau đây được phát như khối tin lên đường truyền nối tiếp:

LSB

1001001 1100001 1000001 0000100 1111111

Hãy trình bày cấu trúc khung hoàn chỉnh khi truyền khối tin với các kiểu truyền sau:

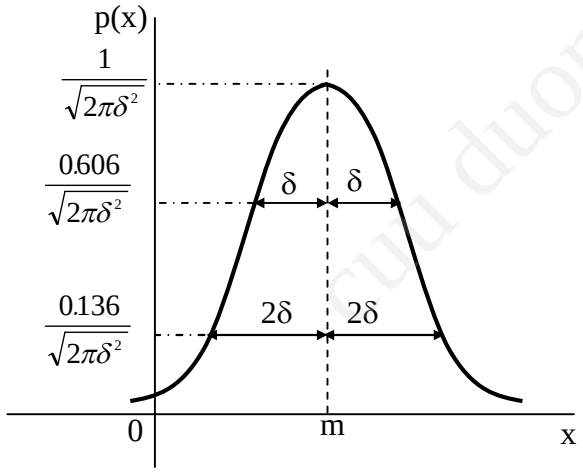
- Bất đồng bộ, mã ASCII, kiểm tra lẻ, 1 stop, 7 data.
- Đồng bộ : ịnh hướng bit, mã ASCII, kiểm tra lẻ

Bits 7654321	Character	Bits 7654321	Character	Bits 7654321	Character	Bits 7654321	Character
0000000	NUL	0100000	SP	1000000	@	1100000	`
0000001	SOH	0100001	!	1000001	A	1100001	a
0000010	STX	0100010	“	1000010	B	1100010	b
0000011	ETX	0100011	#	1000011	C	1100011	c
0000100	EOT	0100100	$	1000100	D	1100100	d
0000101	ENQ	0100101	%	1000101	E	1100101	e
0000110	ACK	0100110	&	1000110	F	1100110	f
0000111	BEL	0100111	'	1000111	G	1100111	g
0001000	BS	0101000	(	1001000	H	1101000	h
0001001	HT	0101001	)	1001001	I	1101001	i
0001010	LF	0101010	*	1001010	J	1101010	j
0001011	VT	0101011	+	1001011	K	1101011	k
0001100	FF	0101100	,	1001100	L	1101100	1
0001101	CR	0101101	-	1001101	M	1101101	m
0001110	SO	0101110	.	1001110	N	1101110	n
0001111	SI	0101111	/	1001111	O	1101111	o
0010000	DLE	0110000	0	1010000	P	1110000	p
0010001	DC1	0110001	1	1010001	Q	1110001	q
0010010	DC2	0110010	2	1010010	R	1110010	r
0010011	DC3	0110011	3	1010011	S	1110011	s
0010100	DC4	0110100	4	1010100	T	1110100	t
0010101	NAK	0110101	5	1010101	U	1110101	u
0010110	SYN	0110110	6	1010110	V	1110110	v
0010111	ETB	0110111	7	1010111	W	1110111	w
0011000	CAN	0111000	8	1011000	X	1111000	x
0011001	EM	0111001	9	1011001	Y	1111001	y
0011010	SUB	0111010	:	1011010	Z	1111010	z
0011011	ESC	0111011	;	1011011	[	1111011	{
0011100	FS	0111100	<	1011100	\	1111100	|
0011101	GS	0111101	=	1011101	]	1111101	}
0011110	RS	0111110	>	1011110	^	1111110	~
0011111	US	0111111	?	1011111	—	1111111	DEL

3.6 Nhiễu gauss và tỷ lệ lỗi bit

- Nhiễu Gauss :
  - Hàm mật độ công suất của nhiễu Gauss :

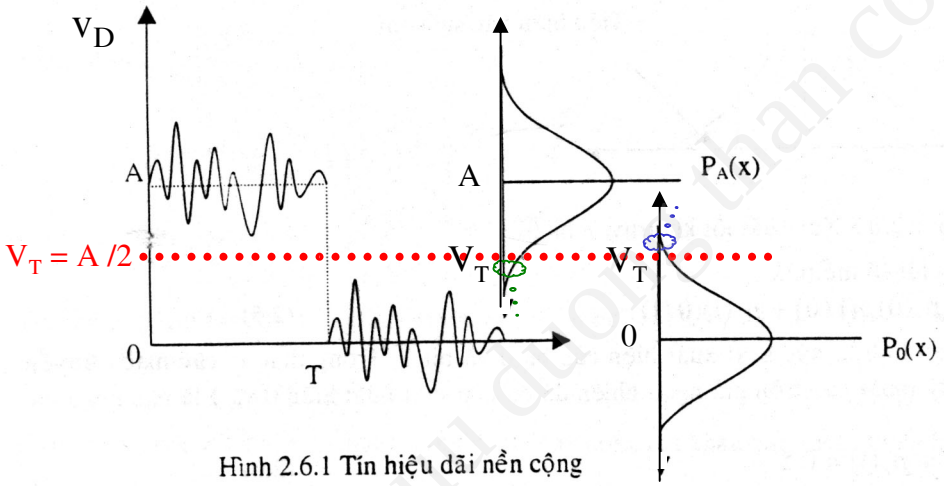
$$p(x) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-(x-m)^2/2\delta^2}$$



- m: giá trị trung bình (DC).
- δ : Độ lệch chuẩn (áp hiệu dụng)
- δ^2 : gọi là phương sai (công suất nhiễu)

Truyền đồng bộ (Synchronous transmission)

- Xét tín hiệu truyền là dãi nền, và chỉ chịu tác động của nhiễu Gauss bằng cách cộng trực tiếp vào tín hiệu, có dạng như sau :



Hình 2.6.1 Tín hiệu dãi nền cộng nhiễu AWGN

Nhiễu Gauss có thể biểu diễn qua hàm mật độ công suất pdf (power density function)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-m)^2/2\sigma^2} \qquad (2.1)$$

V_T: ngưỡng xác quyết.

$v_D > v_T$: Xác quyết mức '1'

$v_D < v_T$: Xác quyết mức '0'

Nếu δ càng lớn thì xác suất xác quyết nhầm càng cao.

Truyền đồng bộ (Synchronous transmission)

- Tín hiệu nhận được cộng luôn cả nhiễu nếu lớn hơn V_T thì xác quyết mức '1', ngược lại nhỏ hơn V_T thì xác quyết mức '0'. Do đó ta có :
  - Xác xuất lỗi khi truyền bit 1 sai là:

$$P_r(v_D < v_T) = p(0/1) = \int_{-\infty}^{v_T} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-A)^2}{2\sigma^2}} dx$$

  - Xác xuất lỗi khi truyền bit 0 sai là:

$$P_r(v_D > v_T) = p(1/0) = \int_{v_T}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} dx$$

  - Giả sử xác suất xuất hiện bit 1 và 0 là $p_r(1)$ và $p_r(0)$
- Xác suất lỗi 1 bit :

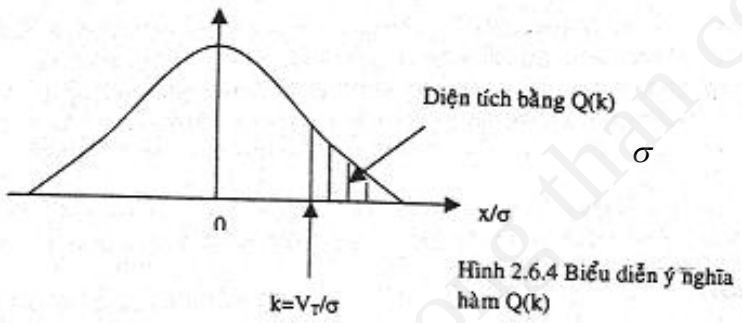
$$p_e = p_r(1)p(0/1) + p_r(0)p(1/0)$$

Nếu xác suất xuất hiện 0 và 1 là như nhau tức $p_r(0) = p_r(1) = 0.5$, thì

$$p_e = 0.5\ p(0/1) + 0.5\ p(1/0) = p(0/1) = p(1/0).$$

Truyền đồng bộ (Synchronous transmission)

- Để tính p(0/1) hay p(1/0) thì dựa vào hàm Q(k).
- Tính chất hàm Q(k)



Hình 2.6.4 Biểu diễn ý nghĩa hàm Q(k)

- Hàm Q(k) thực chất là hàm phân bố chuẩn với m = 0,σ = 1.

$$Q(k) \approx \frac{e^{-\frac{k^2}{2}}}{\sqrt{2\pi}k}$$

- Khi dùng hàm Q(k) để tính xác suất thì cần chuẩn hóa giá trị ngưỡng. Trong trường hợp này $v_T = A/2$ nên $k=v_T/\sigma$

  -> $p_e = p(0/1)=p(1/0) = Q(v_T/\sigma)=Q(A/2\sigma)$

Truyền đồng bộ (Synchronous transmission)

- Ngoài ra, xác suất lỗi có thể tính dựa vào $(S/N)_v$, hoặc $(S/N)_p$
  - $P_e = Q\,(A/2\sigma) = Q\,[(S/N)_V]$
  - $P_e = Q\,(A/2\sigma) = Q\,[(S/N)_P]$
- Xác suất sai k bit bất kỳ khi truyền khối n bit

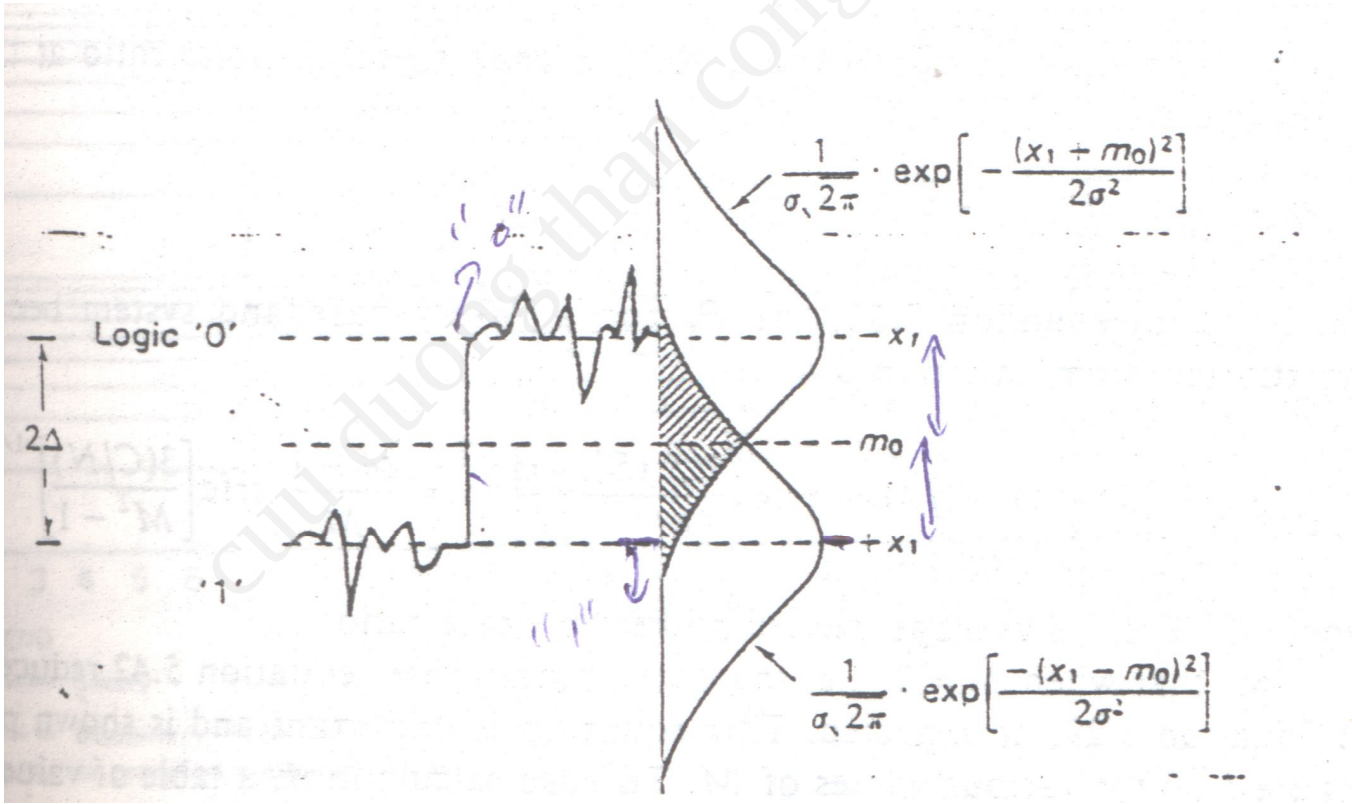
$$p_k = C_n^k p_e^k (1 - p_e^k)^{n-k}$$

- Nếu truyền n bits mà toàn bộ bị sai (k=0).

$p_r(\text{error}) = 1\text{-}p_0 = 1\text{-}(1\text{-}p_e)^n \approx np_e$. (do $p_e << 1$)

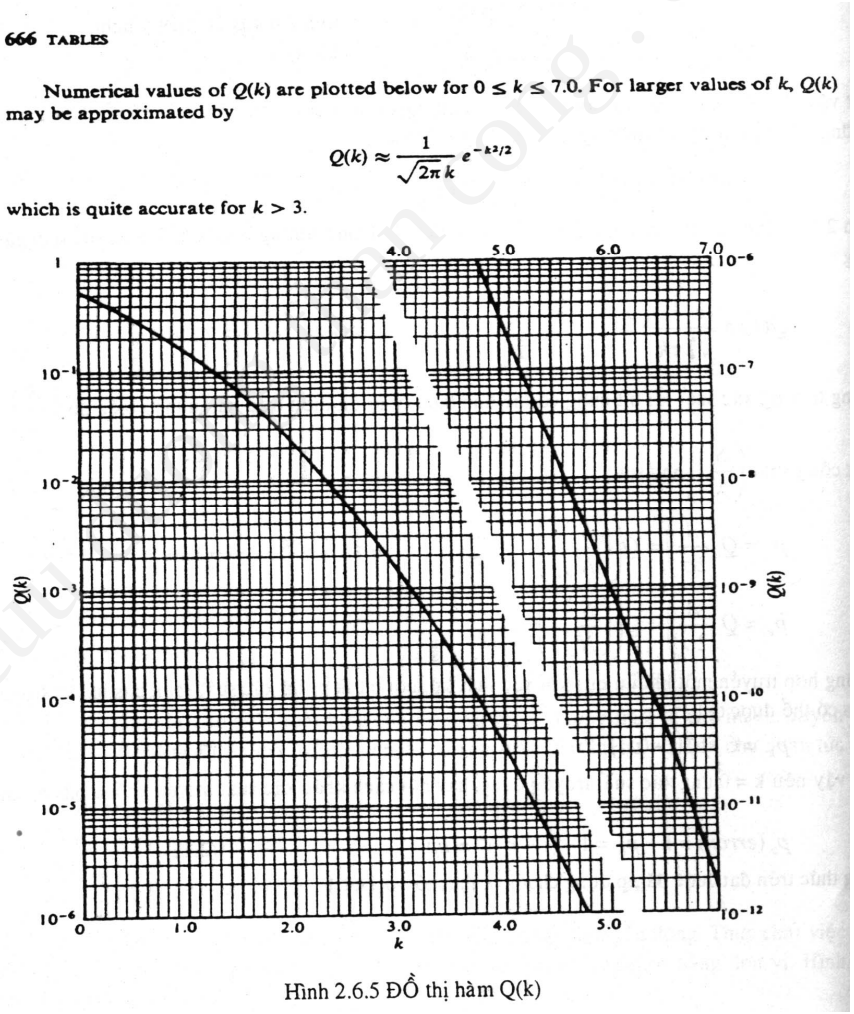
Truyền đồng bộ (Synchronous transmission)

- Trường hợp tổng quát $v_T = m_o$



Truyền đồng bộ (Synchronous transmission)

- Đồ thị tính hàm Q(k)



666 TABLES

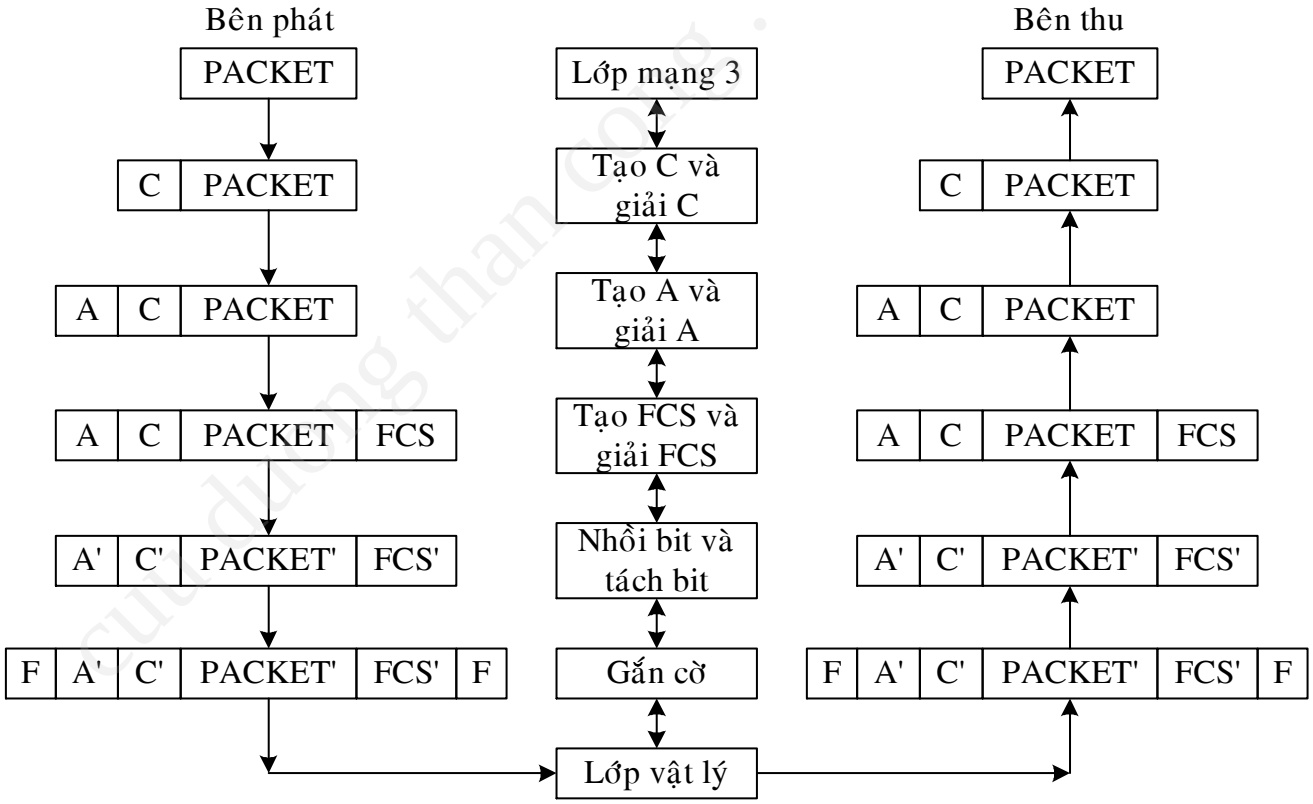
Numerical values of $Q(k)$ are plotted below for $0 \le k \le 7.0$. For larger values of k, $Q(k)$ may be approximated by

$$Q(k) \approx \frac{1}{\sqrt{2\pi}\,k} e^{-k^2/2}$$

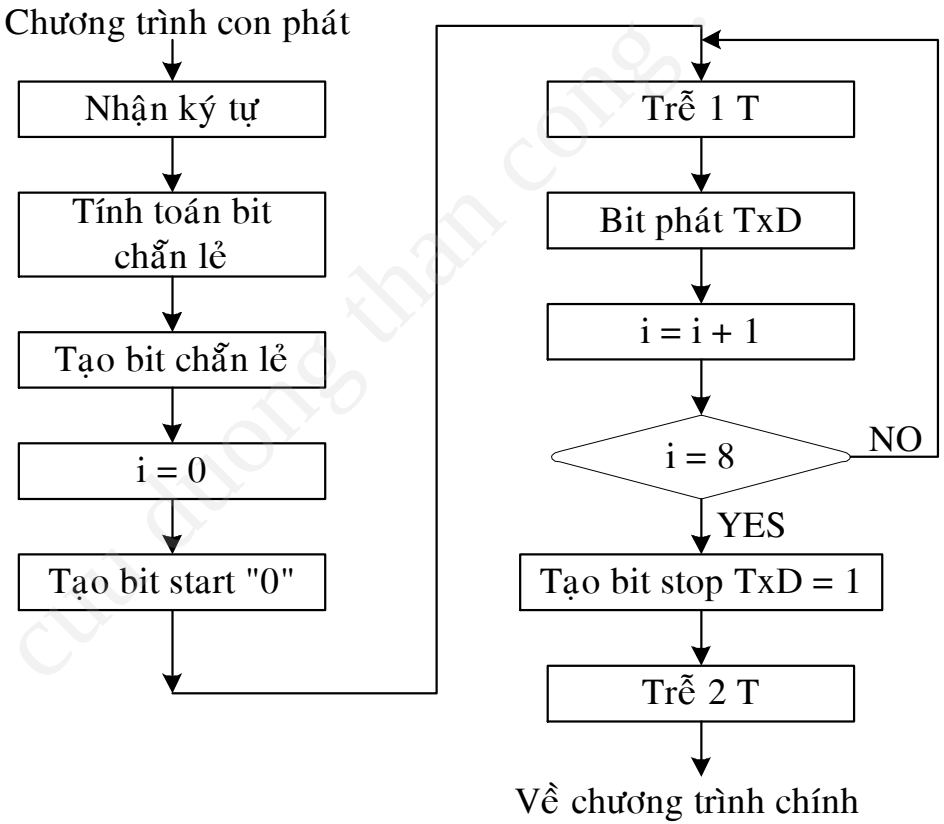
which is quite accurate for $k > 3$.

Hình 2.6.5 ĐỒ thị hàm Q(k)

Đồng bộ



UART – Bất đồng bộ



3.7 Các phương pháp phát hiện lỗi :

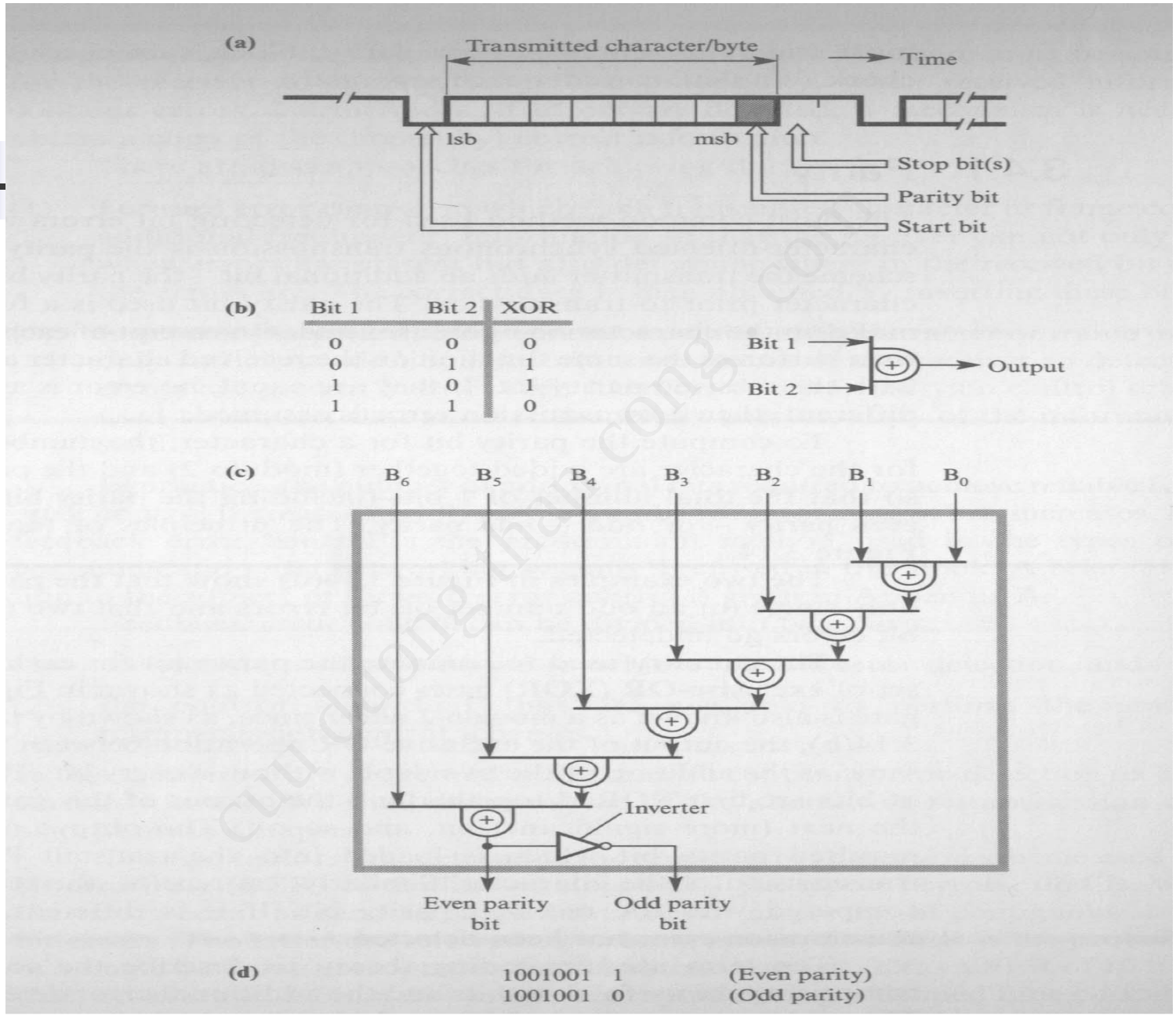
- Có 2 phương pháp phát hiện lỗi
  - Forward error control : Mỗi ký tự hoặc khung khi truyền chứa thêm thông tin bổ sung để phía thu khi nhận dựa vào thông tin này phát hiện dữ liệu nhận được có bị sai hay không, nếu sai thì phía thu tiến hành sửa (nếu có thể). Thường sử dụng cho các đường truyền rất xa có thời gian trễn do lan truyền lớn.
  - Feedback (backward) error control : Mỗi ký tự hoặc khung khi truyền chứa thêm thông tin bổ sung để phía thu khi nhận dựa vào thông tin này phát hiện dữ liệu nhận được có bị sai hay không, chứ không tiến hành sửa. Nếu sửa sai thì yêu cầu bên gửi phát lại, phương pháp này thường được sử dụng trong các hệ thống truyền thông.

Các phương pháp phát hiện lỗi

- Forward error control :
  - Parity check
  - Block sum check
  - Cyclic Redundant Check
  - Hamming

Các phương pháp phát hiện lỗi

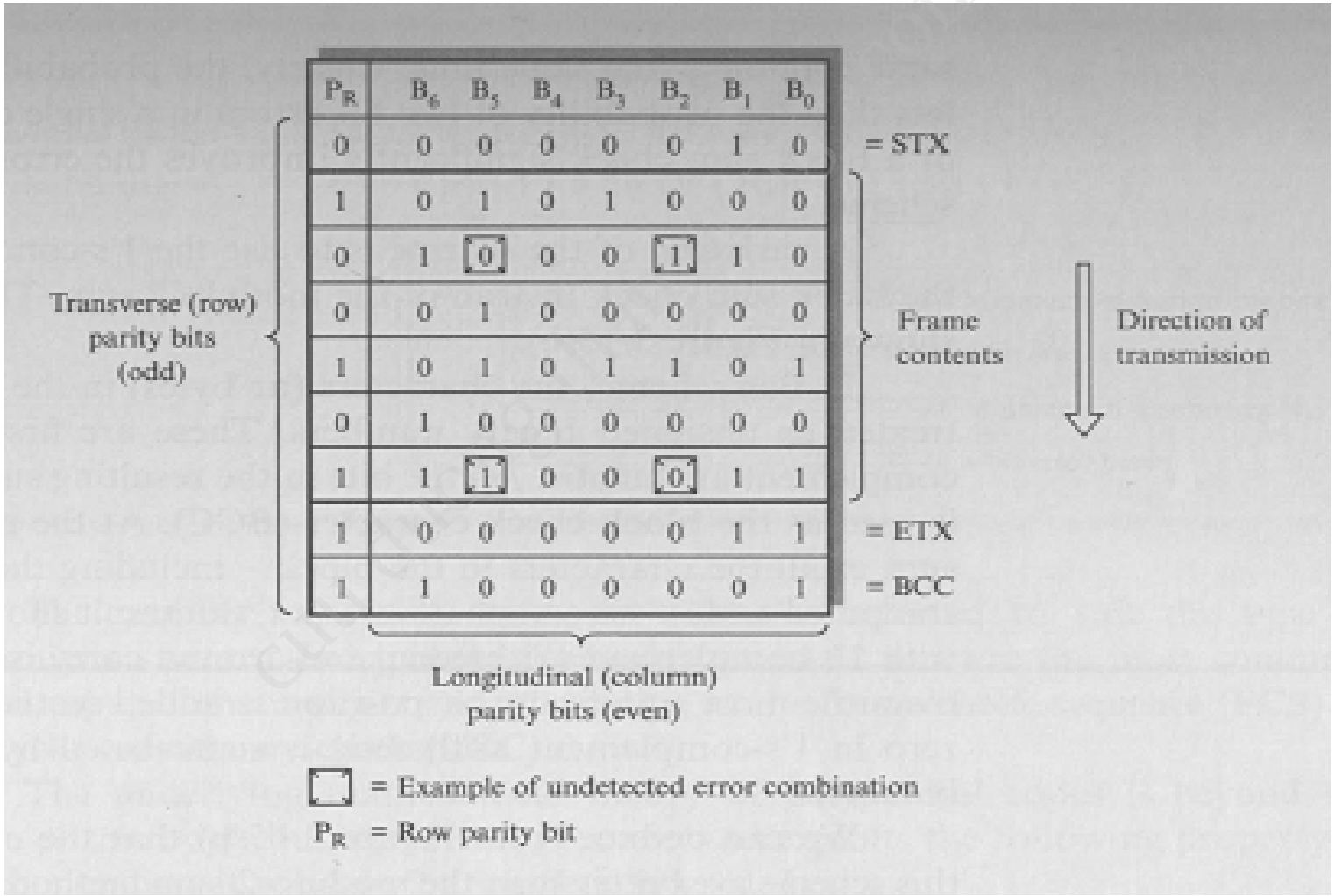
- Parity check :
  - Trong mỗi ký tự (7 hoặc 8 bit) truyền đi khối kiểm tra sẽ thực hiện việc chèn một bit (parity bit) vào cuối ký tự (ngay trước stop bit). Giá trị parity bit là 0 hay 1 tuỳ vào phương pháp kiểm tra là kiểm tra chẵn (even parity) hay kiểm tra lẻ (odd parity).
    - Kiểm tra chẵn : Tổng số bit 1 trong tất cả bit dữ liệu (không kể start và stop bit) và parity bit là số chẵn
    - Kiểm tra lẻ: Tổng số bit 1 trong tất cả bit dữ liệu (không kể start và stop bit) và parity bit là1 số lẻ.
  - Phía thu sẽ thực hiện việc tính lại parity bit sau đó so sánh parity bit nhận được, nếu khác nhau thì phía thu sẽ hiểu rằng đã có lỗi xảy ra trên đường truyền.
    - Phát hiện sai nếu tổng số bit sai là số lẻ
    - Không phát hiện sai nếu tổng số bit sai là số lẻ.



Các phương pháp phát hiện lỗi

- Block sum check :
  - Sử dụng khi truyền dữ liệu dưới dạng một khối các ký tự, trong kiểu kiểm tra này, mỗi ký tự truyền đi sẽ được phân phối 2 bit kiểm tra parity là parity hàng và parity cột. Các bit parity theo từng cột được gọi là ký tự kiểm tra khối BCC- block check character.
    - Phát hiện và sửa sai nếu lỗi bit đơn.
    - Không phát hiện sai nếu các bit sai kiểu chùm như : sai 4 bit, 2 bit cùng hàng và 2 bit cùng cột.
    - Các trường hợp còn lại thì phát hiện sai được
  - Thường sử dụng trong kiểu truyền bất đồng bộ.

Các phương pháp phát hiện lỗi



P_R	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	1	0	= STX
1	0	1	0	1	0	0	0	
0	1	0	0	0	1	1	0	
0	0	1	0	0	0	0	0	
1	0	1	0	1	1	0	1	
0	1	0	0	0	0	0	0	
1	1	1	0	0	0	1	1	
1	0	0	0	0	0	1	1	= ETX
1	1	0	0	0	0	0	1	= BCC

Các phương pháp phát hiện lỗi

- Cyclic Redundant Check :
  - Phía phát tạo ra một ký số kiểm tra khung FSC (frame sequence check) hay CRC, FSC được phát kèm theo phía sau của frame thông tin.
  - Phát hiện được tất cả lỗi bít đơn, bit đôi,bit lẻ hay bit chùm.
  - Thường sử dụng trong kỹ thuật truyền đồng bộ.
  - Giả sử gọi :
    - M(x) : bản tin cần truyền đi (the message to be transmitted)gồm kbit.
    - G(x) : đa thức sinh (the divisor or generator) gồm n+1bit
    - R(x) : số dư gồm n bit (k > n)
    - Q(x) : thương số của phép chia
    - T(x) : thông điệp truyền đi gồn (n+k) bit

Các phương pháp phát hiện lỗi

- Bên phát :
  - Bước 1 :Chuyển thông điệp nhị phân M thành đa thức M(x).
    - Nhân đa thức M(x) với x^n, tương đương với chuỗi bit nhị phân được dịch sang trái n bit.
  - Bước 2: Thực hiện phép chia $\frac{M(x).x^n}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$

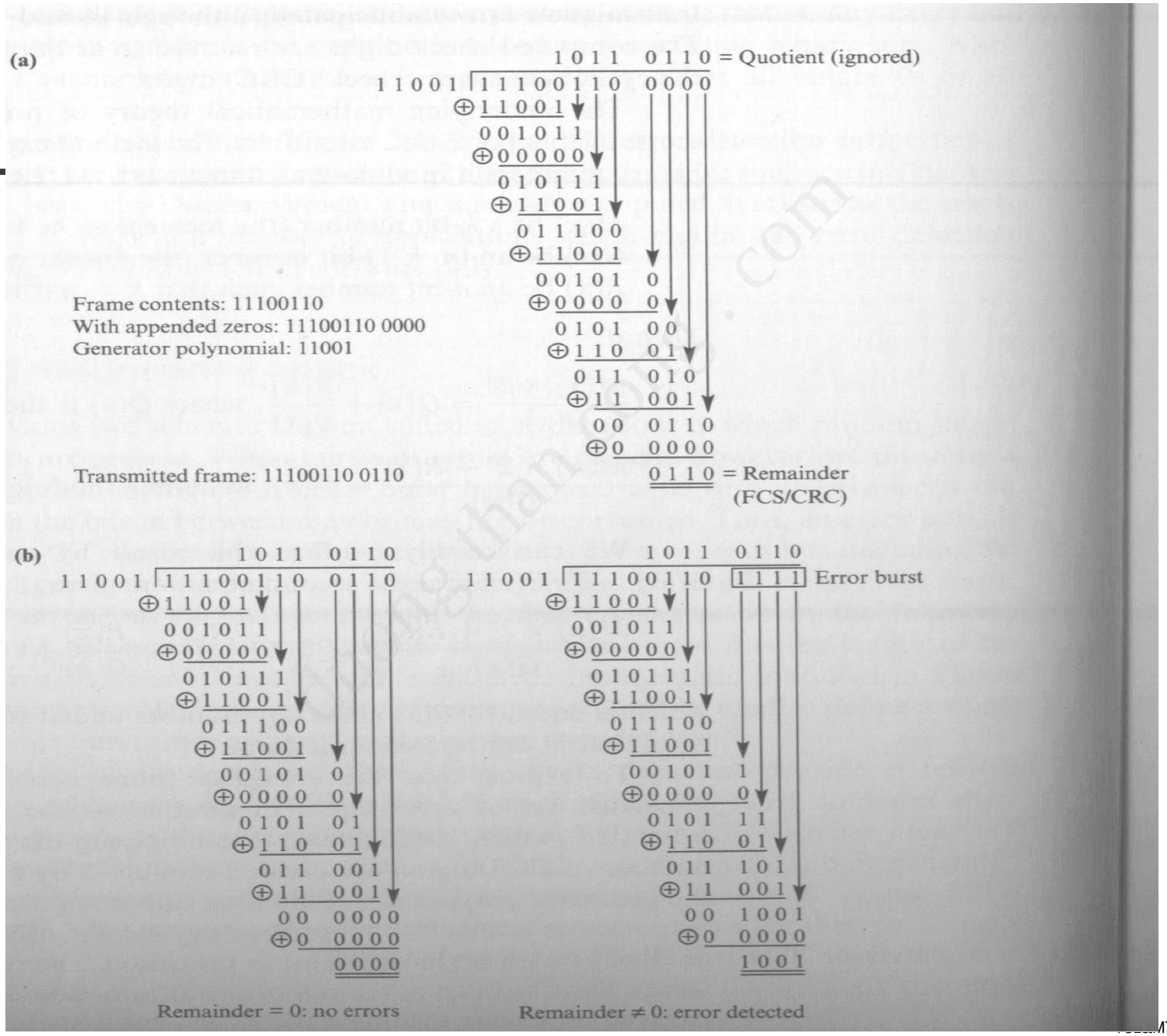
    Điều này được thực hiện theo đa thức hoặc theo modulo 2.
  - Bước 3: Thông điệp cần truyền đi là T(x):

    $T(x) = x^n . M(x) + R(x)$

    Điều này được thực hiện theo đa thức hoặc thêm n bit của số dư (R(x)) trong phép chia ở Bước 2 vào sau k bit của bản tin cần truyền
- Bên thu :
  - Việc phát hiện lỗi được thực hiện bằng cách lấy chuỗi dữ liệu thu được chia modulo 2 cho đa thức sinh G(x) như sau:

$$\frac{T(x)}{G(x)} = \frac{x^n M(x) + R(x)}{G(x)} = \frac{x^n M(x)}{G(x)} + \frac{R(x)}{G(x)} = Q(x) + \underbrace{\frac{R(x)}{G(x)} + \frac{R(x)}{G(x)}}_{=0} = Q(x)$$

    - Do trong phép cộng modulo 2 thì 2 số giống nhau cộng lại bằng 0
  - Như vậy nếu phần dư trong phép chia này bằng 0 thì phía thu xem như không có lỗi xảy ra, ngược lại nếu khác 0 thì phía thu phát hiện được lỗi xảy ra khi truyền dữ liệu.



Các phương pháp phát hiện lỗi

- Hamming :
  - Khoảng cách mã (khảong cách Hamming) là số bit khác nhau giữa 2 từ mã.
  - Mã Hamming là bộ mã mà thêm một số bit kiểm tra ở một số vị trí nhất định trong thông tin cần truyền để tạo thành mã Hamming.
  - Bên thu không chỉ phát hiện được sai mà còn có thể sửa sai ở một số vị trí nhất định
  - Ví dụ :
    - Để mã hóa các số thập phân từ 0 – F ta cần 4 bit. Gọi 4 bit đó là $m_3m_2m_1m_0$.
    - Trước khi truyền ta chèn vào 3 bit kiểm tra $c_2c_1c_0$., 3 bit này được tính bằng cách EX-OR :
      - $c_0 = m_0 + m_1 + m_3$
      - $c_1 = m_0 + m_2 + m_3$
      - $c_2 = m_1 + m_2 + m_3$
    - Đầu thu thực hiện kiểm tra bằng cách tính :
      - $p_0 = c_0 + m_0 + m_1 + m_3$
      - $p_1 = c_1 + m_0 + m_2 + m_3$
      - $p_2 = c_2 + m_1 + m_2 + m_3$

        + Nếu không sai thì $p_0 = p_1 = p_2 = 0$

        + Nếu sai thì thì số nhị phân $p_2p_1p_0$ là vị trí của bit sai (các bit trong từ mã Hamming truyền đi được đánh thứ tự từ 0 đến 7) và sửa bằng cách đảo bit này.

Hamming

Ví dụ: Mã Hamming (7, 4) gồm 4 bit mang tin $m_3\ m_2\ m_1\ m_0$ và 3 bit kiểm tra $c_2\ c_1\ c_0$.

$\Rightarrow$ Mã Hamming có dạng : $m_3\ m_2\ m_1\ m_0\ c_2\ c_1\ c_0$.

* Các bit kiểm tra c_2, c_1, c_0 được tính như sau :

$$c_0 = m_0 \oplus m_1 \oplus m_3$$
$$c_1 = m_0 \oplus m_2 \oplus m_3$$
$$c_2 = m_1 \oplus m_2 \oplus m_3$$

Ví dụ mã cho chữ A là : $m_3\ m_2\ m_1\ m_0 = 1010$

$$c_0 = 0 \oplus 1 \oplus 1 = 0$$

$\Rightarrow$ $c_1 = 0 \oplus 0 \oplus 1 = 1$

$$c_2 = 1 \oplus 0 \oplus 1 = 0.$$

$\Rightarrow$ Vậy mã Hamming truyền là : 1010010.

* Khi bộ phận thu thu được một từ mã, có thể tính lại 3 giá trị kiểm tra : p_0, p_1, p_2 như sau :

$$p_0 = c_0 \oplus m_0 \oplus m_1 \oplus m_3 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$$
$$p_1 = c_1 \oplus m_0 \oplus m_2 \oplus m_3 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$$
$$p_2 = c_2 \oplus m_1 \oplus m_2 \oplus m_3 = 0 \oplus 1 \oplus 0 \oplus 1 = 0.$$

Vì $p_0 = p_1 = p_2 = 0 \Rightarrow$ phép truyền không sai.

* Ví dụ : phía thu nhận được thông tin : 1110010.

$\Rightarrow$ Ta có : $p_0 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$

$$p_1 = 1 \oplus 0 \oplus 1 \oplus 1 = 1 \quad \Rightarrow p_2p_1p_0 = 110 \Rightarrow 6.$$
$$p_2 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

Vậy bit thứ 6 bị sai.