



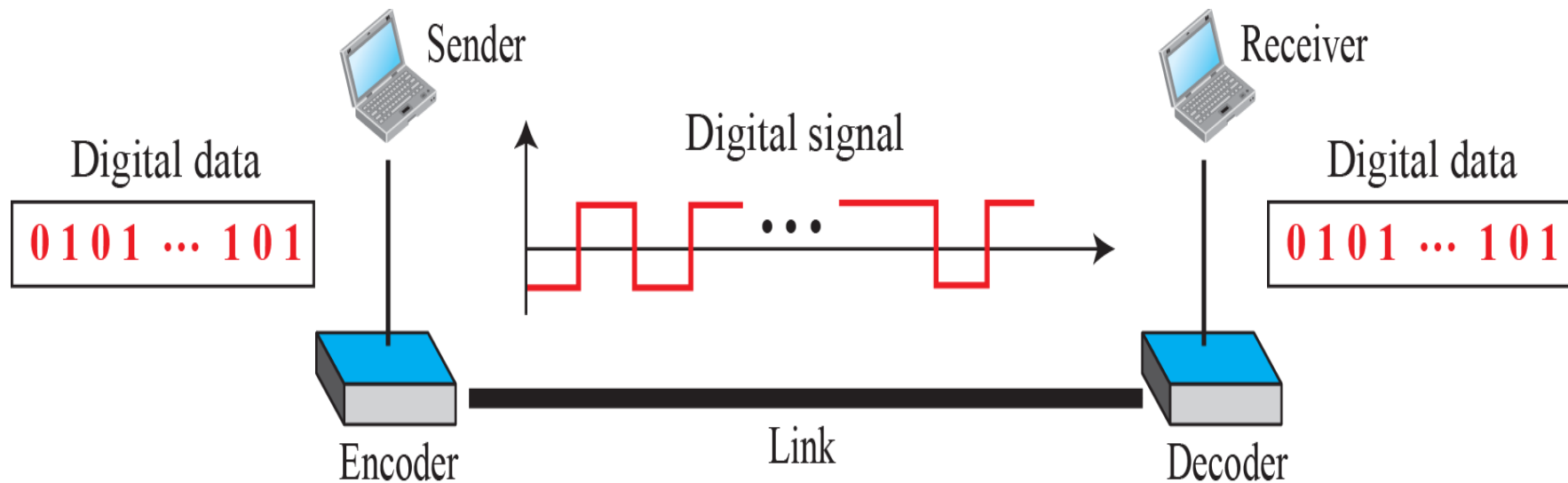
데이터 통신

Part 2. 물리층

Chapter 4. 디지털 전송

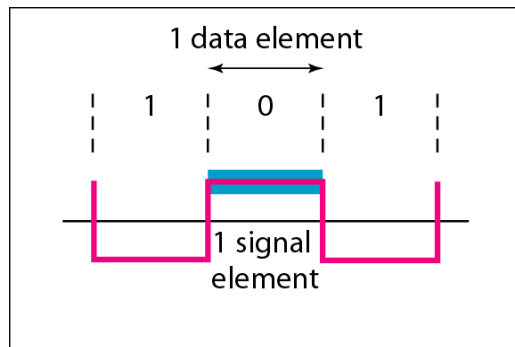
회선 코딩

- ◆ 회선 부호화 또는 회선 코딩(line coding)
 - 디지털 데이터를 디지털 신호로 바꾸는 작업

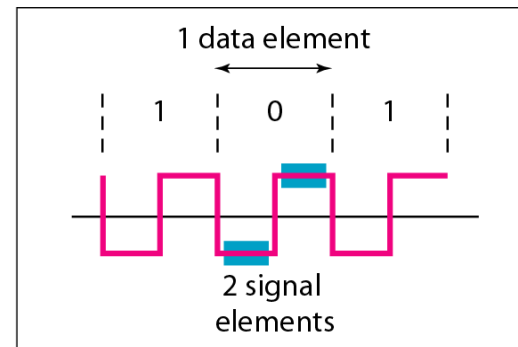


회선 코딩

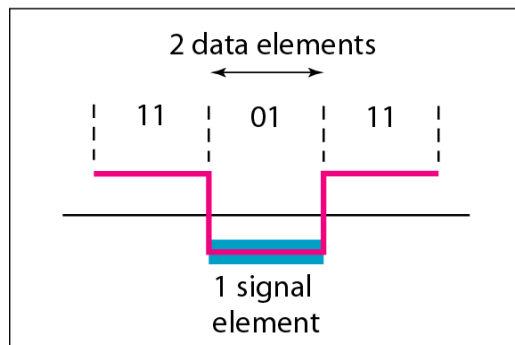
- ◆ 신호 요소 (파형의 변화) vs. 데이터 요소
 - r : 신호 요소당 데이터 요소(비트) 개수



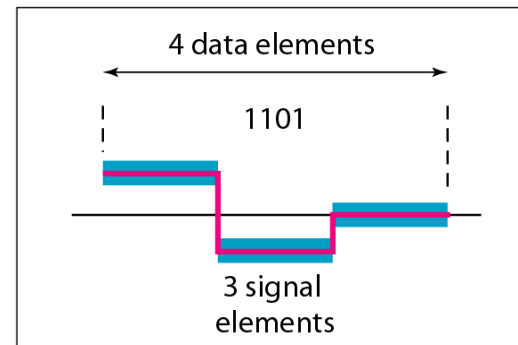
a. One data element per one signal element ($r = 1$)



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 ($r = \frac{4}{3}$)

회선 코딩

◆ 데이터 전송률 대 신호 전송률

- 데이터율 N (bps) : 초당 전송되는 데이터 요소의 개수
- 신호율 S (baud) : 초당 전송되는 신호 요소의 개수
 - 펄스율(pulse rate), 변조율(modulation rate), 보오율(baud rate)
- 데이터율(N)과 신호율(S)의 관계
(r : 신호요소당 데이터요소의 개수, c : 요인 변수)

$$S = c \times N \times \frac{1}{r} \text{ baud}$$

- 데이터통신의 목표는 신호율에 대한 데이터율의 증대

◆ 대역폭

- 보오율이 디지털 신호의 요구 대역폭을 결정
- 최대 데이터율

$$B_{\min} = c \times N \times \frac{1}{r}$$

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

회선 코딩

◆ 기준선(Base line)

- 수신자가 수신한 신호의 세기에 대한 평균값
- 기준선과 비교하여 데이터 요소의 값 결정
- 기준선이 표류(Wandering)하면 제대로 복호화 하기 어려움
- 좋은 회선 코딩은 기준선 표류 방지가 필요

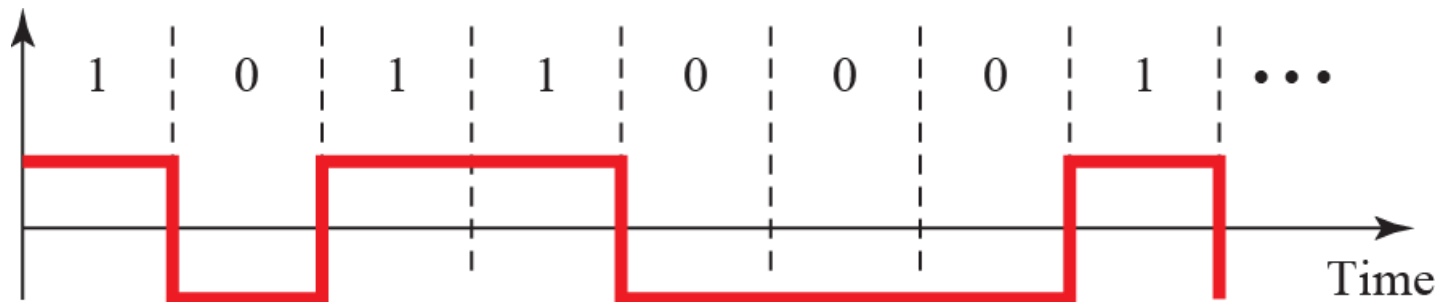
◆ 직류 성분(DC component)

- 주파수가 낮은 성분은 통과하지 못하는 시스템이 존재하므로, 직류 성분(0 주파수 주위에 생기는 주파수) 이 생기지 않는 방법 필요

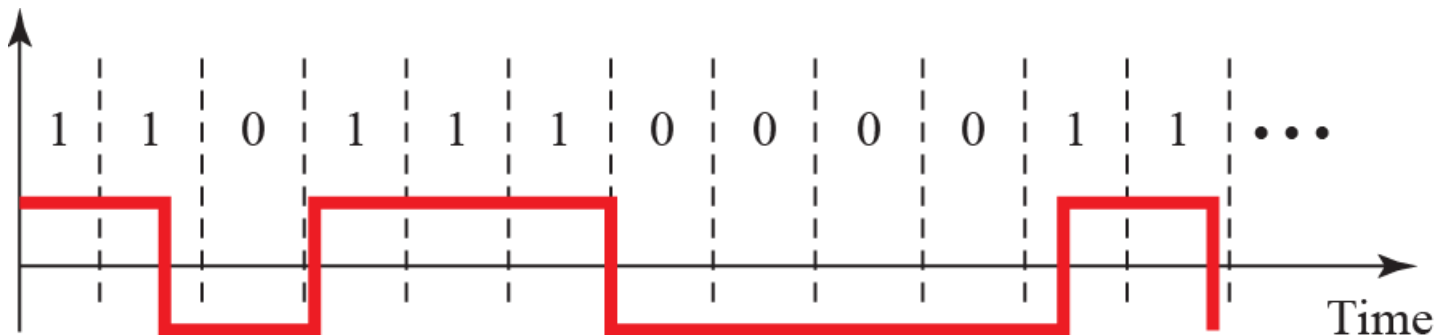
회선 코딩

◆ 자기 동기화(self-synchronization)

- 자기 동기화 디지털 신호는 전송되는 데이터에 타이밍 정보를 포함
 - 이유 : 발신자와 수신자의 비트 간격이 일치해야 함



a. Sent



b. Received

회선 코딩

- ◆ 자기 동기화

- 예제 4.3

- 어떤 디지털 전송에 있어서 수신자의 타이머가 발신자의 타이머보다 0.1% 빠르다고 한다. 만일 데이터 전송률이 1Kbps라면 수신자는 매초 얼마만큼의 추가 비트를 받게 되겠는가? 만일 데이터 전송률이 1Mbps라면?

1 Kbps인 경우:

1000 bits sent → 1001 bits received → 1 extra bps

1 Mbps인 경우:

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

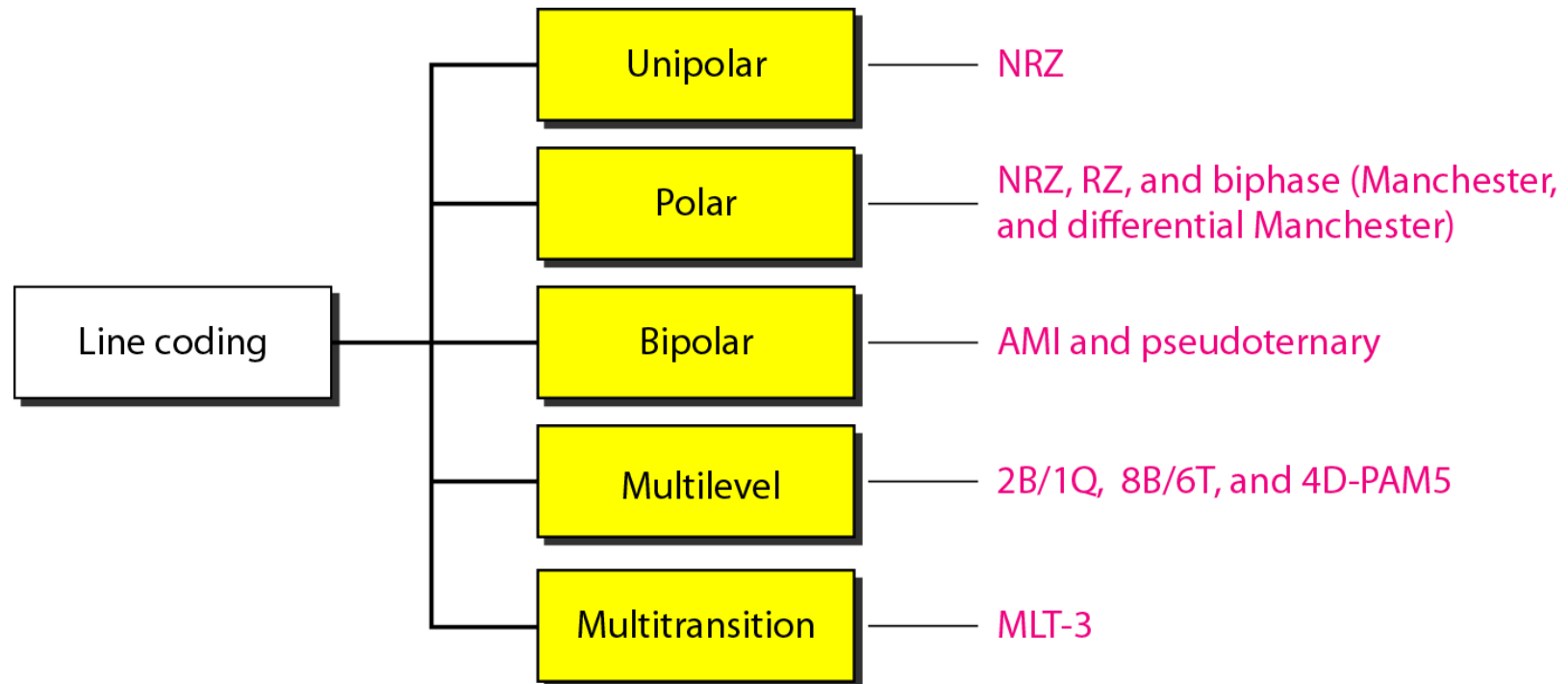
- ◆ 내장형 오류 발견

- ◆ 잡음과 간섭 신호에 대한 내성

- ◆ 복잡도

회선 코딩 방식

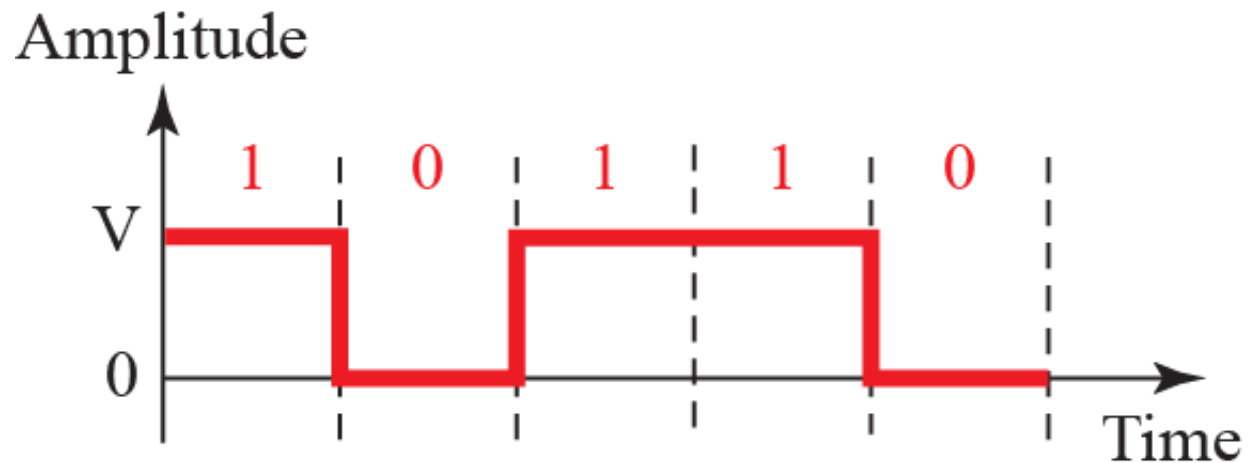
◆ 회선 코딩 방식



회선 코딩 방식

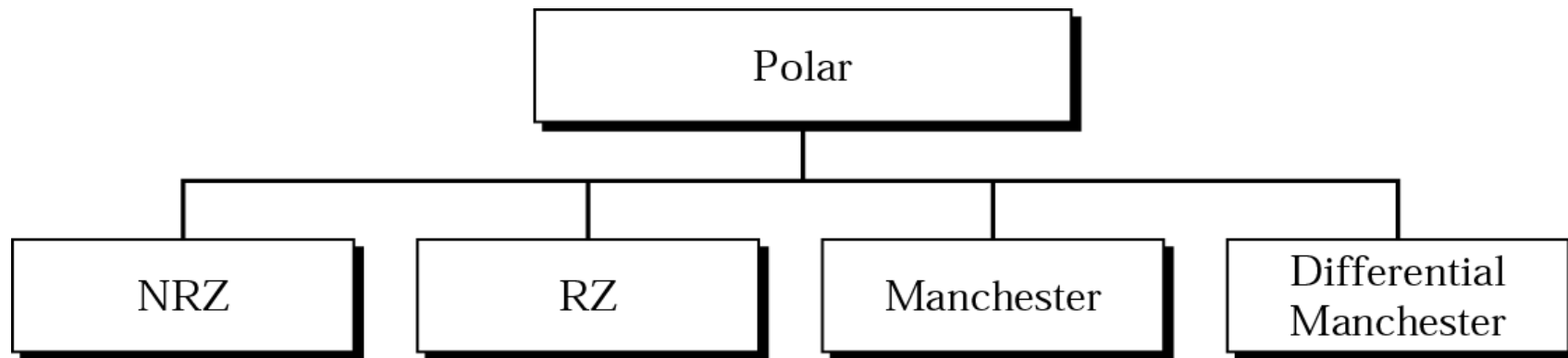
◆ 단극형(Unipolar)

- 단극형 부호화는 오직 한 준위의 값만 이용.
 - 1/0을 (-)전압과 0 또는 (+) 전압과 0으로만 표현
- 오랜 시간 동안 신호의 변화가 없는 경우 동기화에 어려움



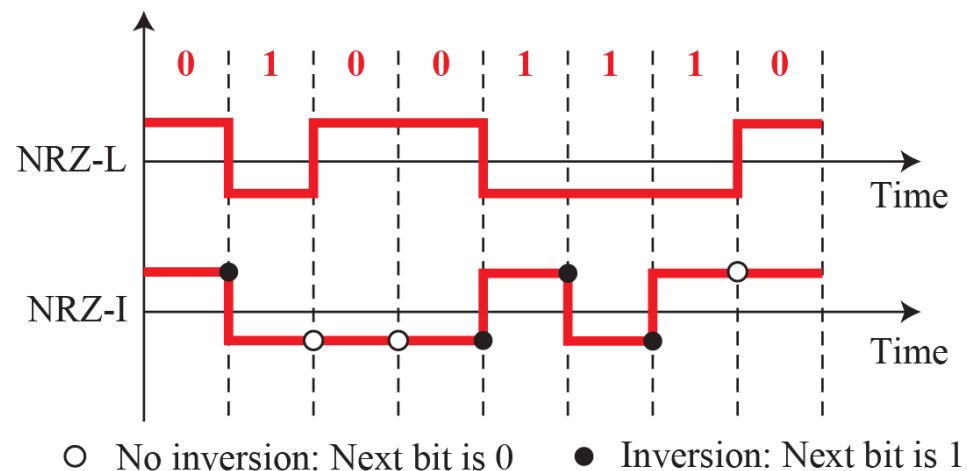
회선 코딩 방식

- ◆ 극형(Polar)
 - 양과 음의 두 가지 전압 준위를 사용.



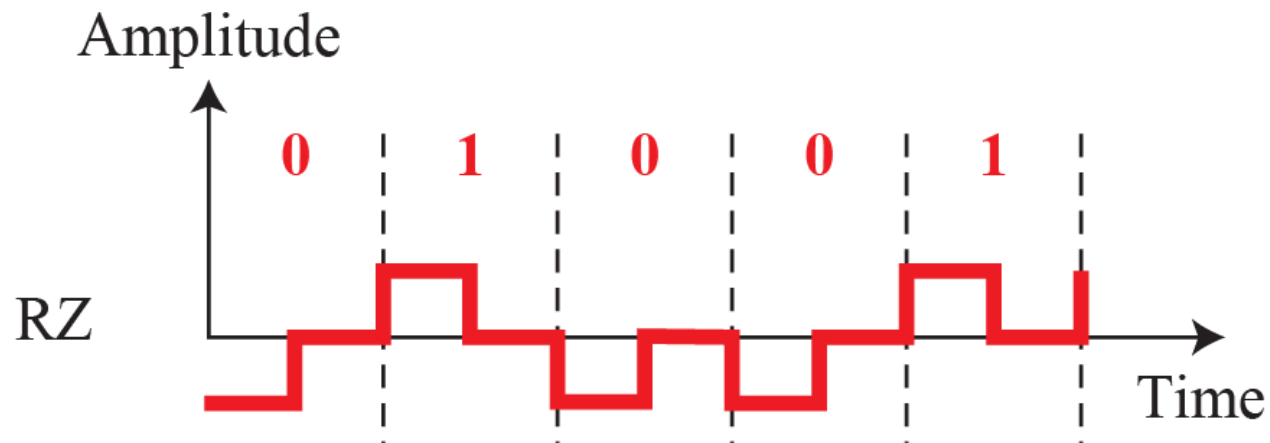
회선 코딩 방식

- ◆ 극형(Polar) – 비영복귀(NRZ, non-return to zero)
 - 신호의 준위는 항상 (+) 또는 (-)
 - NRZ-L (NRZ-Level)
 - 1/0 비트 값을 신호준위에 따라 표현
 - 긴 스트림의 데이터를 수신했을 경우, 송신자와 클럭 동기화 여부 불투명
 - NRZ-I (NRZ-Invert)
 - 1을 만나면 신호가 반전됨
 - 동기화 제공



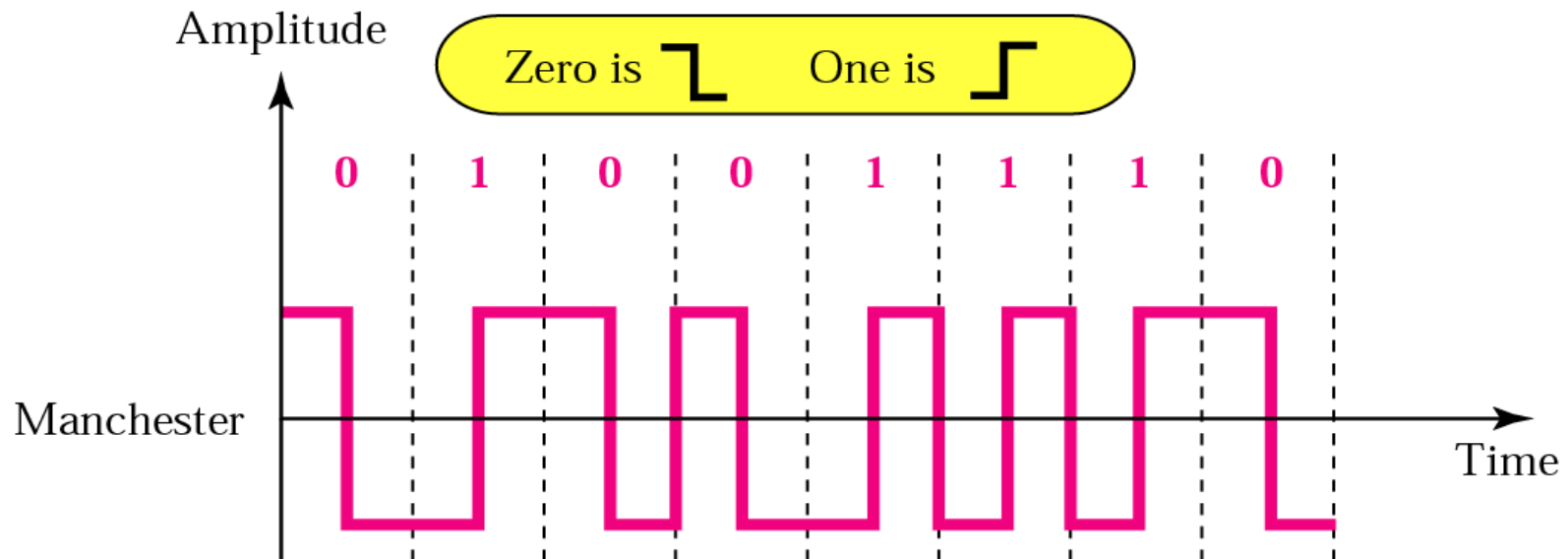
회선 코딩 방식

- ◆ 극형(Polar) – 영복귀(RZ, return to zero)
 - 비트마다의 구간 동안에 바뀜
 - (+, 비트 1), (-, 비트 0), 0 세 가지 전압을 사용
 - 비트 간격의 반을 지나면 0으로 복귀
 - 동기화
 - 단점 : 한 비트 표현을 위해 매번 두 번의 신호변화가 필요
 - 많은 대역폭 차지
 - 복잡성



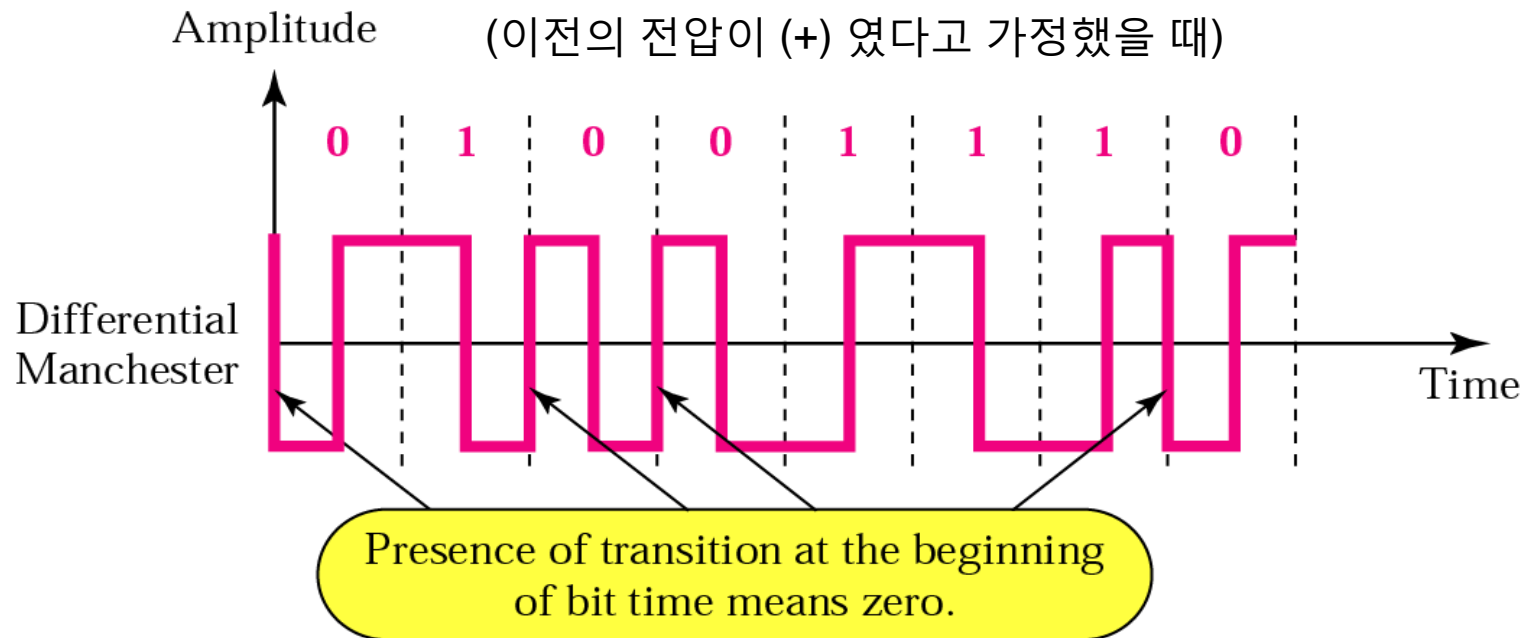
회선 코딩 방식

- ◆ 극형(Polar) – 맨체스터(Manchester)
 - 동기화와 비트 표현을 위해 비트 중간 지점에서의 신호 전이를 사용



회선 코딩 방식

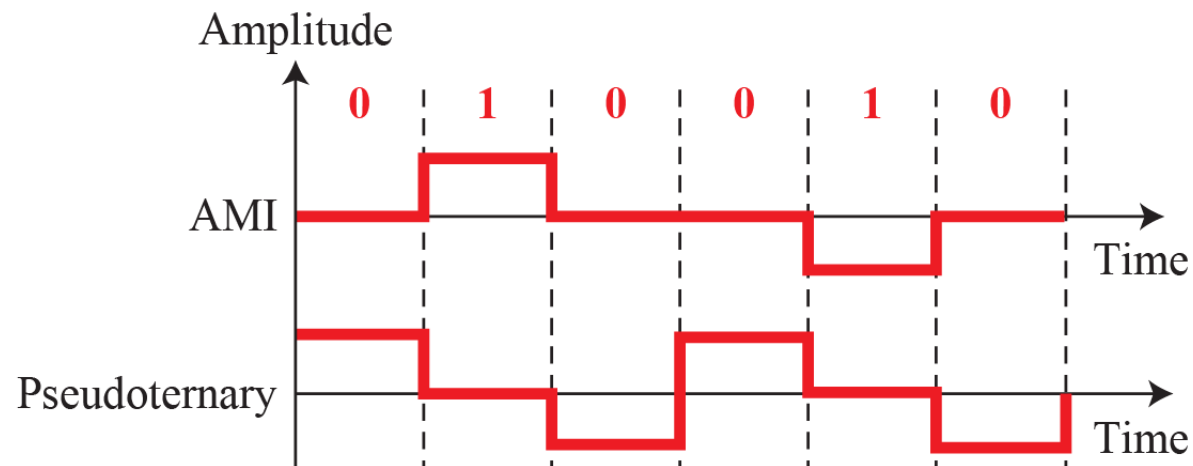
- ◆ 극형(Polar) – 차분 맨체스터(differential Manchester)
- 비트 간격 **중간에서의 반전은 동기화**를 위해 사용
- 비트 표현은 비트의 **시작점에서의 반전 유무**로 나타냄
 - 비트 시작 지점에서의 신호 변화는 0, 그렇지 않으면 1을 표현
 - 0을 표현하는데 두 번의 신호변화, 1을 표현하는데 1번의 변화



회선 코딩 방식

◆ 양극형(Bipolar)

- 양, 음, 영의 세 가지 준위를 사용.
- 교대표시반전(AMI, alternate mark inversion)
 - 교대로 나타나는 반전되는 신호(mark)는 1을 의미함.
 - 즉, 비트 1을 표현하는 신호는 이전의 비트 1의 신호의 반전된 형태



회선 코딩 – 다 준위 방식

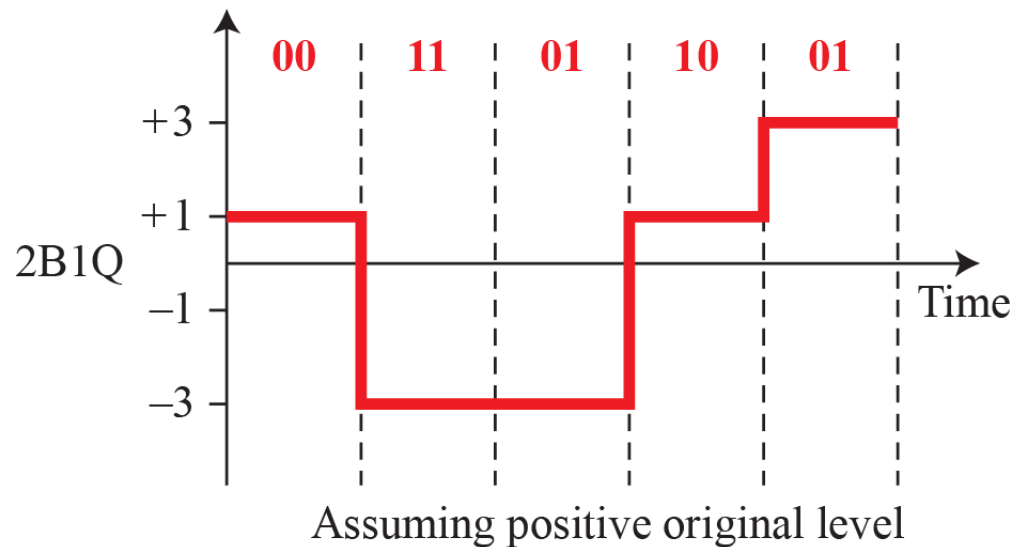
mBnL 방식에서는, m 개의 데이터 요소의 패턴이 $2^m \leq L^n$ 가 되는 n 개의 신호 패턴으로 부호화된다.

- ▶ n 개의 신호 요소 패턴을 사용하여 m 개의 데이터 요소 패턴을 표현하며 단위 보 당 비트 수 증가
- ▶ mBnL 부호화
 - m : 2진수 패턴의 길이
 - B : 2진수
 - n : 신호 패턴의 길이
 - L : 신호 준위의 수, 숫자대신 문자 사용
2진 – B(Binary), 3진 – T(Ternary), 4진 - (Quaternary)

회선 코딩 – 다 준위 방식

◆ 2B1Q(2binary, 1 quaternary)

- 네 개의 전압 준위를 사용
- 이전 전압의 상태에 따라 결정
- 각 펄스는 두 비트를 나타내어 각 펄스를 보다 효율적으로 사용



Next bits	Previous level:	
	positive	negative
Next level	Next level	Next level
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

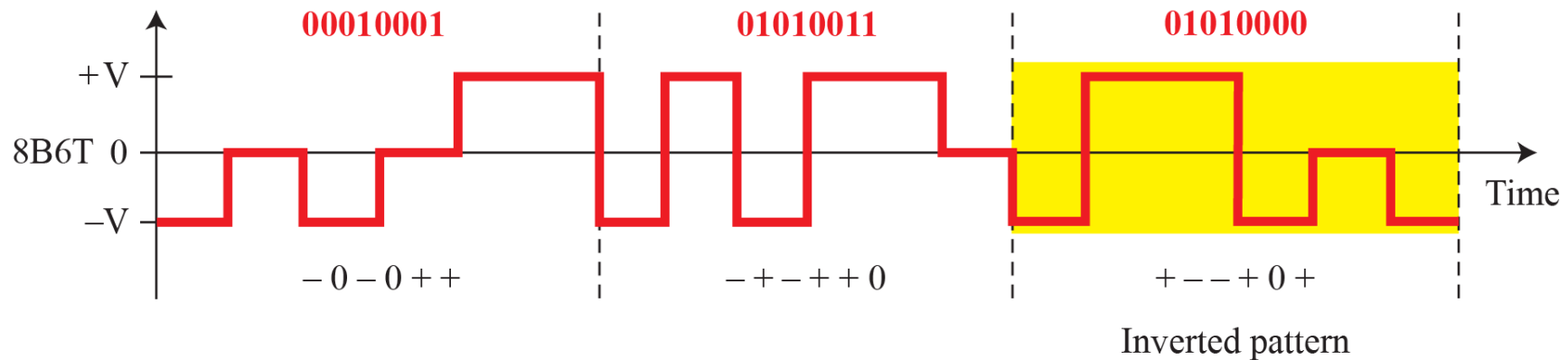
Transition table

회선 코딩 – 다준위 방식

◆ 8B6T : 8 Binary 6 Ternary

● 3개의 준위

- $2^8 = 256$ 개의 데이터 패턴
- $3^6 = 729$ 개의 신호 패턴
- $729 - 256 = 473$ 개의 신호는 동기화나 오류 검색에 사용

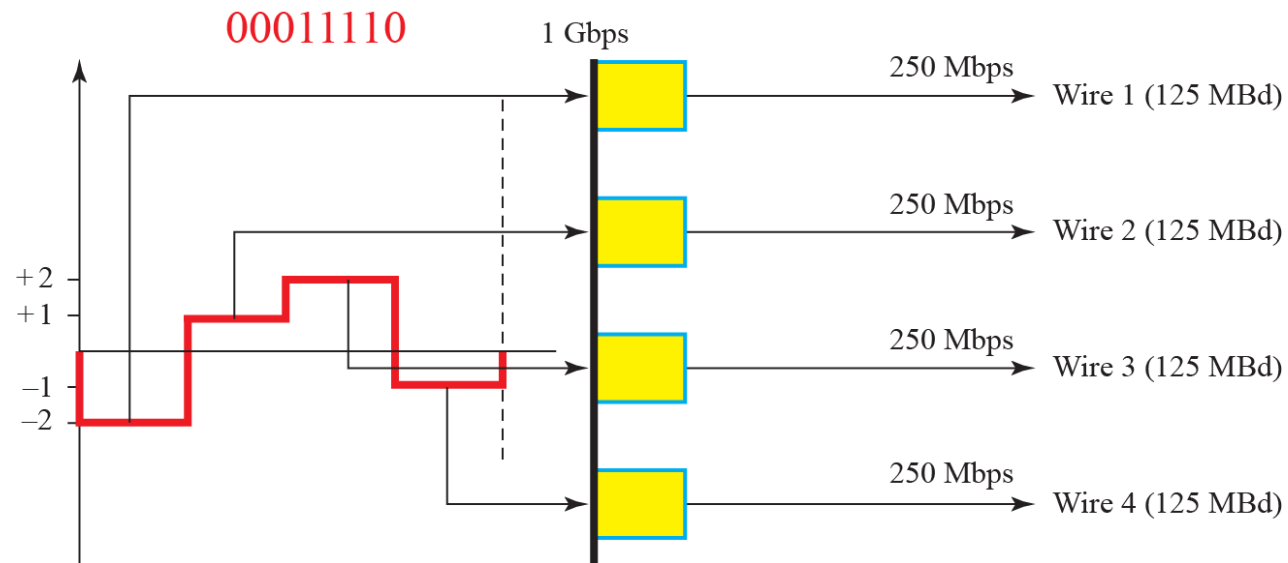


회선 코딩 – 다준위 방식

◆ 4차원 5준위 펄스 진폭 변조

(4D-PAM5 : 4 Dimensional 5 Level Pulse Amplitude Modulation)

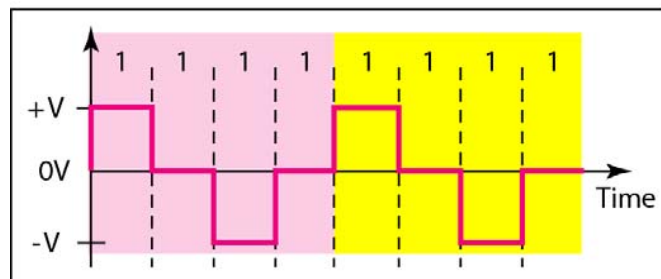
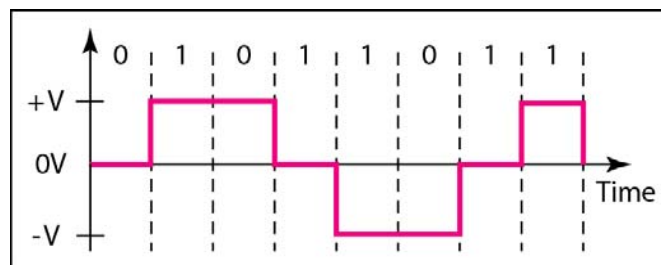
- 4D : 데이터가 4개의 회선으로 동시에 전송
- 5개의 준위 : -2, -1, 0, 1, 2
- 1G bps LAN에 사용



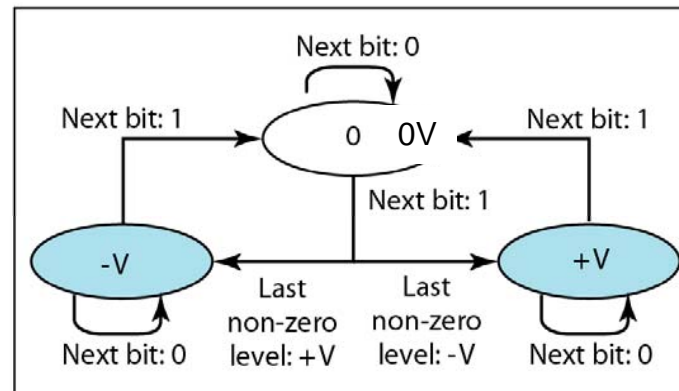
회선 코딩 – 다준위 방식

◆ MLT-3

- 다중회선 전송 3준위(multi-line transmission three level)
- NRZ-I와 매우 유사하지만 세 준위의 신호(+V, 0V, -V)를 사용
 - 다음 비트가 0이면 신호준위 변화 없음
 - 현재 준위가 0V가 아니고 다음 비트가 1이면 다음 신호준위는 0V
 - 현재 준위가 0V이고 다음 비트가 1이면 다음 준위는 마지막으로 0V가 아니었던 준위의 역



←직전 전압이 0V고 마지막 0이 아닌 전압이 -V라고 가정했을 때



회선코딩 요약

<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
Polar	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
Multitransition	MLT-3	$B = N/3$	No self-synchronization for long 0s

블록 코딩

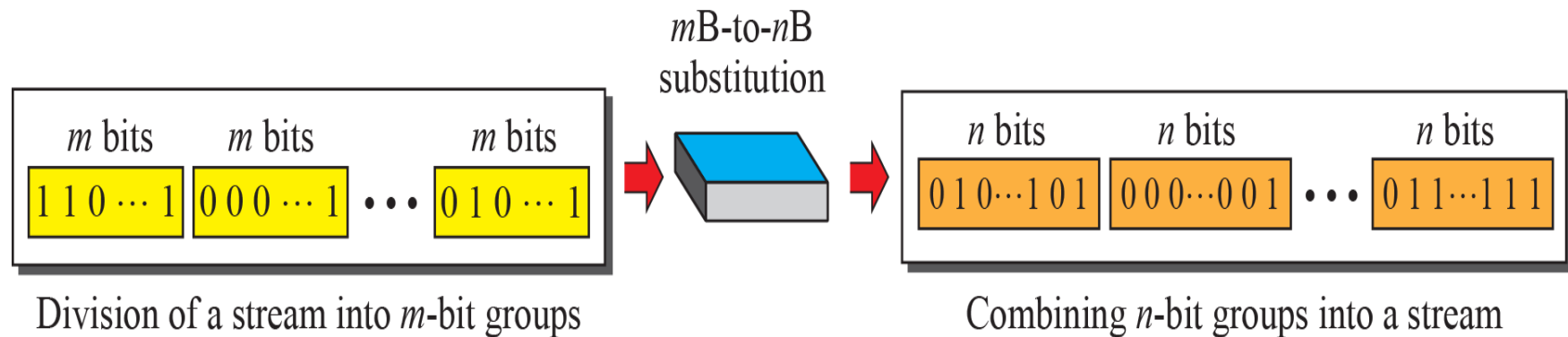
- m 비트를 n 비트 블록으로 바꾼다
- n 은 m 보다 크다
- mB/nB 부호화

블록 부호화는 보통 mB/nB 부호화라
불리며;
각 m -bit 그룹을 n -bit 그룹으로 바꾼다.

블록 코딩

◆ 개념

- 동기화를 확보하기 위해서 여분의 비트가 필요.
- 오류 탐지를 위해서도 여분의 비트가 필요.



블록 코딩

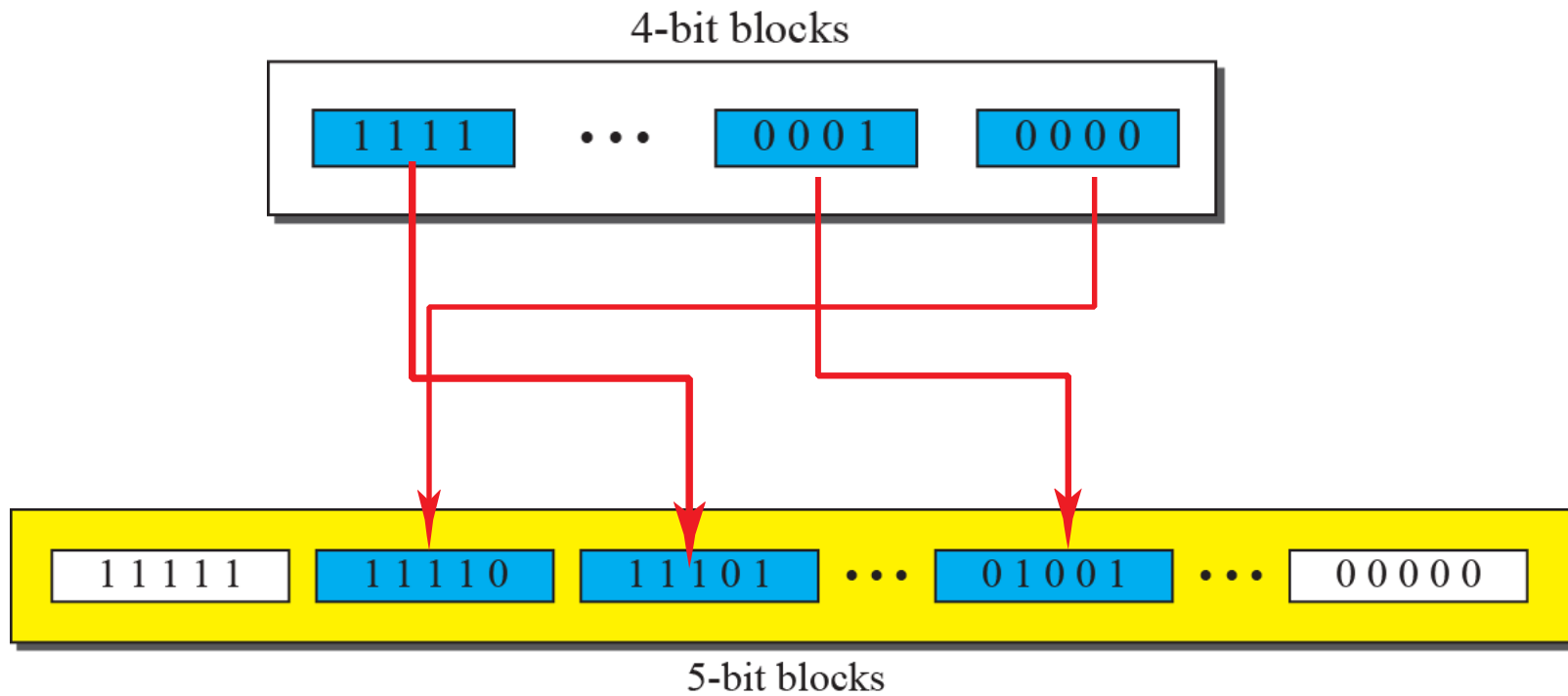
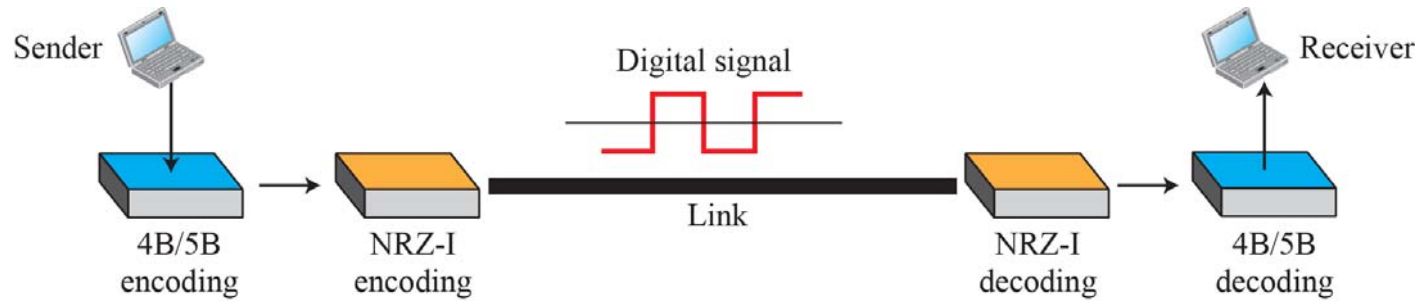
◆ 4B/5B(4Binary / 5Binary)

- NRZ-I 방식과 혼합하여 사용
- 4비트 데이터를 5비트 코드로 변환

<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		

블록 코딩

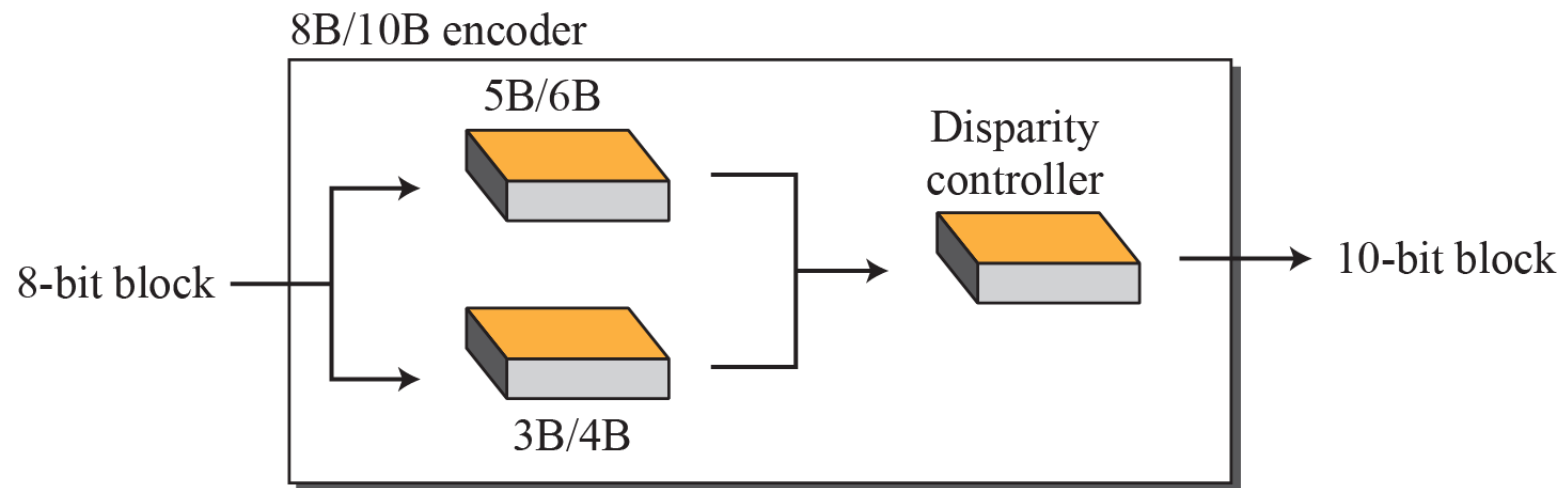
◆ 4B5B 코딩 예제



블록 코딩

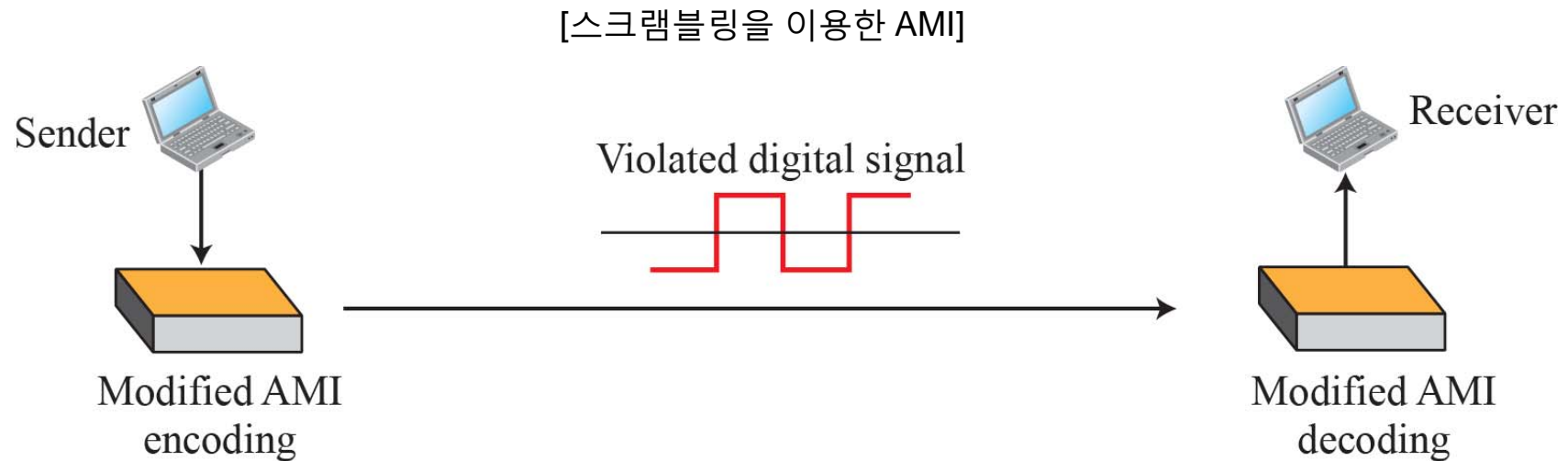
◆ 8B/10B

- 5B/6B와 3B/4B를 합한 것



스크램블링(Scrambling)

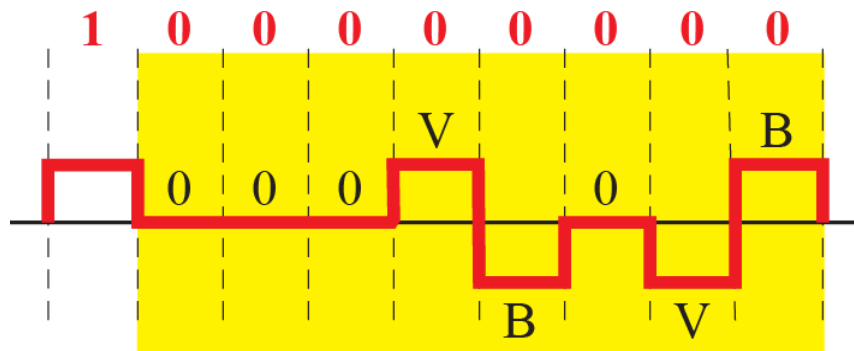
- ◆ 스크램블링 (뒤섞기)
 - 연속되는 0 으로 생기는 동기화 문제 해결 (예, AMI)
 - 다른 준위 신호들로 조합된 신호로 바꾸는 방식



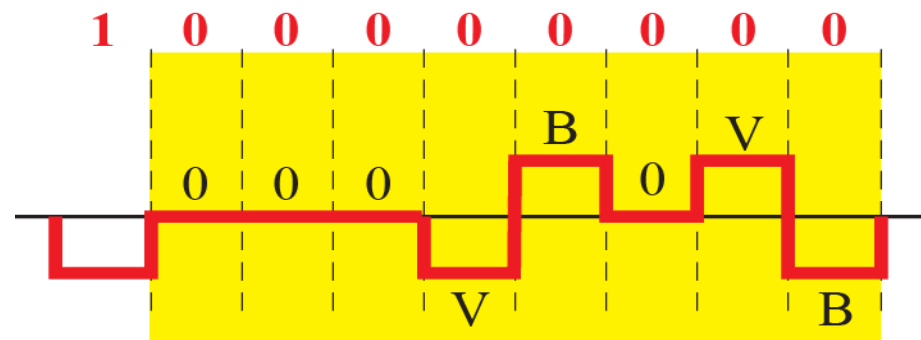
스크램블링(Scrambling)

◆ B8ZS

- 양극 8열 대치(Bipolar with 8 zero substitution)
- 8개의 연속된 0을 000VB0VB 신호로 대치



a. Previous level is positive.

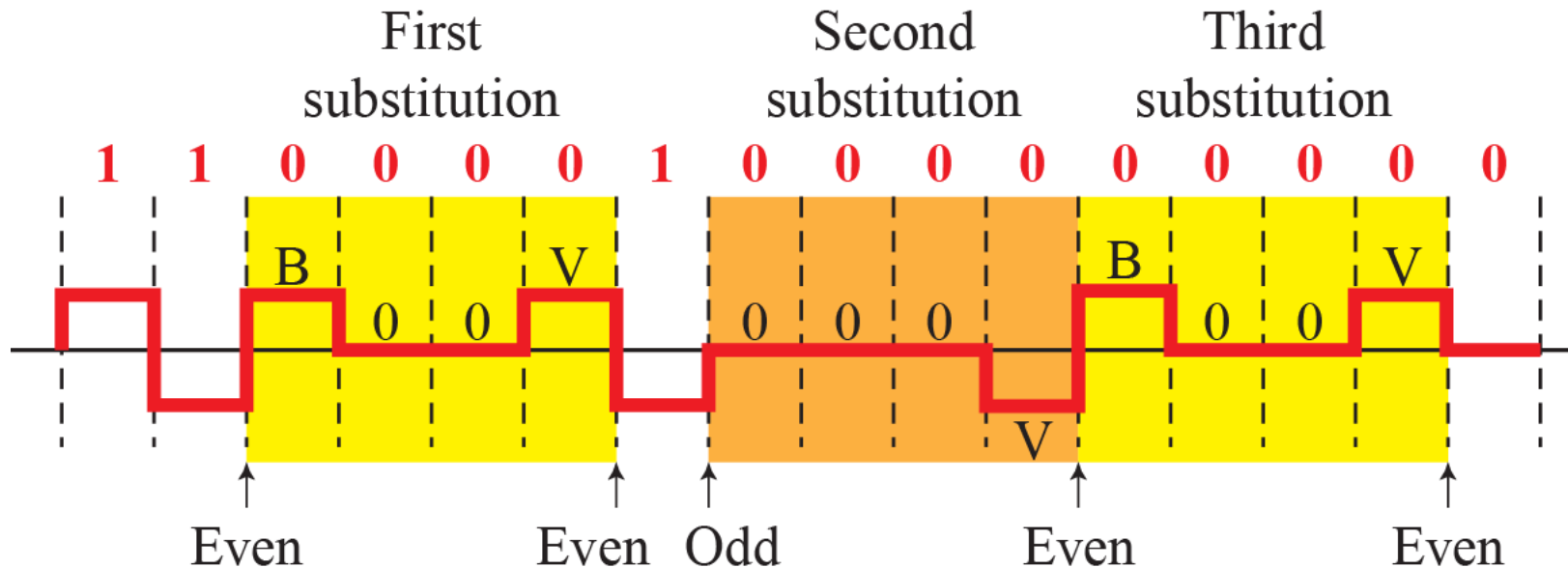


b. Previous level is negative.

스크램블링(Scrambling)

◆ HDB3

- 고밀도 양극 3 영(High-density bipolar 3-zero)
- HDB3은 연속된 4개의 0을 마지막 대체 이후의 0이 아닌 펄스의 개수에 따라 000V 나 B00V 로 대체

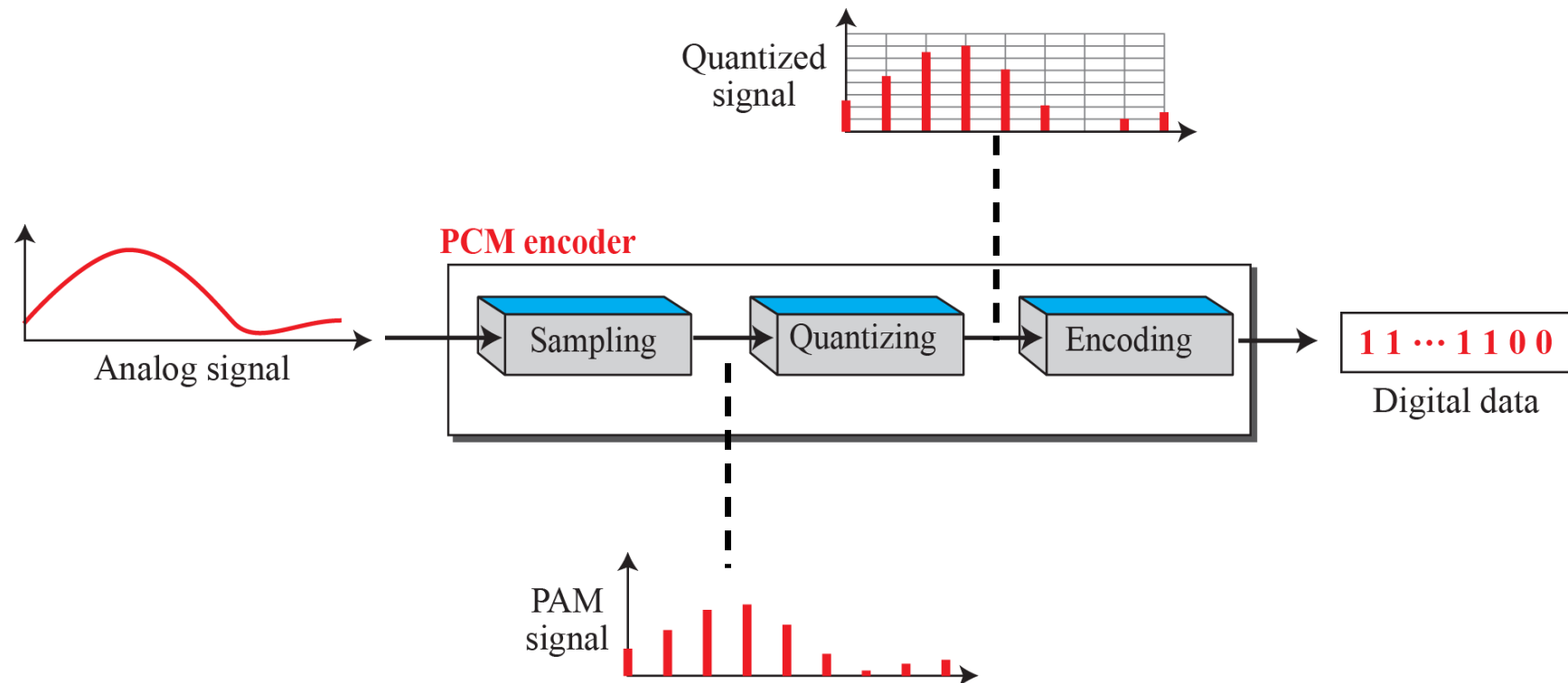


아날로그 대 디지털 변화

- ◆ 목적 : 아날로그 전기 신호를 디지털 방식으로 전송
- ◆ 오늘날의 추세는 아날로그 신호를 디지털 데이터로 바꾸는 것
- ◆ 디지털 신호는 잡음과 일그러짐에 덜 취약

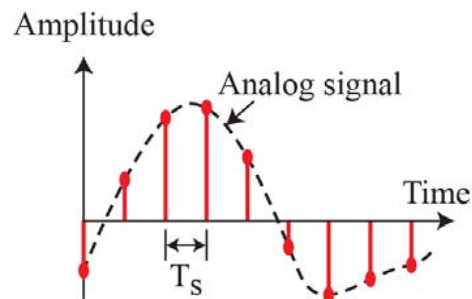
펄스 코드 변조

- ◆ 펄스 코드 변조(PCM, pulse code modulation)
 - 아날로그 대 디지털 부호화 기법
- ◆ 아날로그 신호로 표본을 채집(sampling)하고 정량화(quantizing)하여 디지털 데이터로 변환
 - 채집 : 일정한 간격마다 신호의 진폭을 측정하는 것을 말함

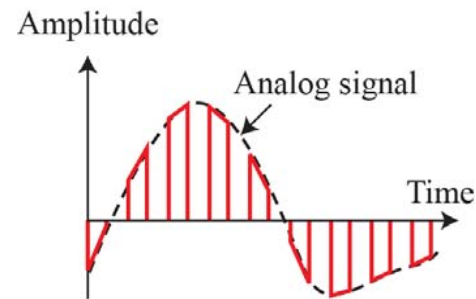


펄스 코드 변조 - sampling

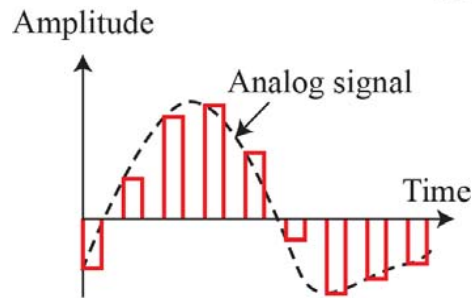
- ◆ 채집(sampling)
 - PCM의 첫 단계
 - 아날로그 신호를 매 T_s (초) 마다 채집
 - 3가지 서로 다른 채집 방법



a. Ideal sampling



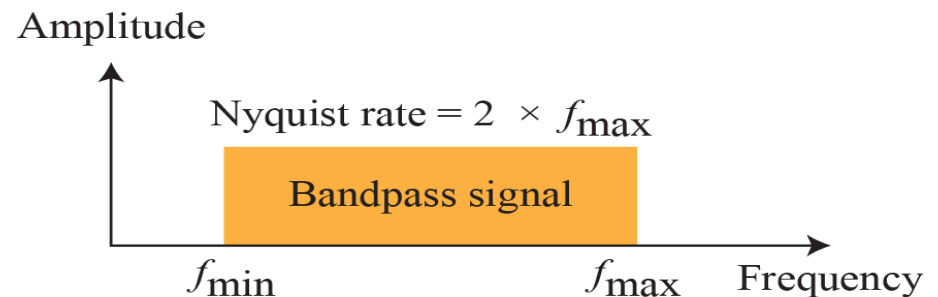
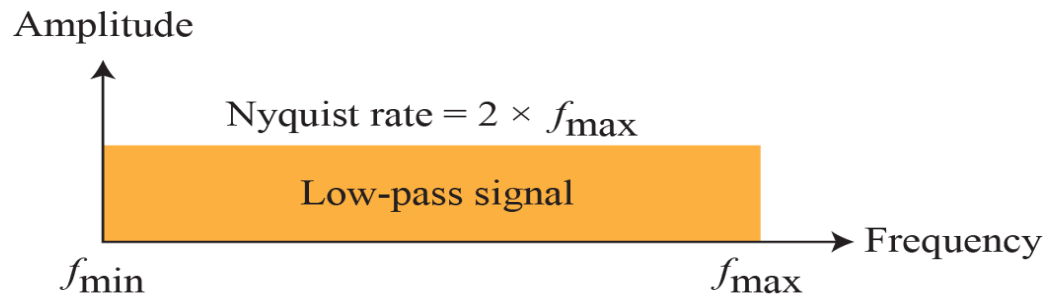
b. Natural sampling



c. Flat-top sampling

펄스 코드 변조 – sampling

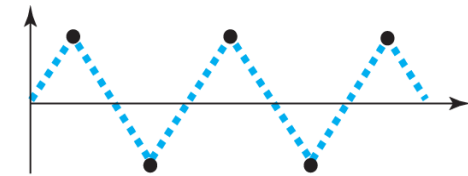
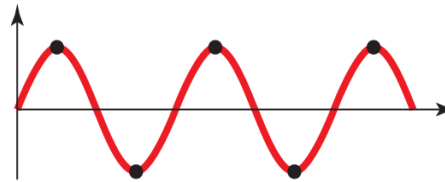
- ◆ 나이퀴스트 정리(Nyquist theorem)에 의하여 표본 채집률은 최소한 **최고 주파수의 두 배**가 되어야 함
 - 나이퀴스트 주파수에서 채집을 하면 원래 정현파에 근접한 결과를 가질 수 있게 됨
 - 예) 최고 4000Hz를 사용하는 전화 시스템의 표본 채집율은 8000개/초
→ 즉, $1/8000 = 125\mu s$ 의 시간 간격으로 채집



펄스 코드 변조 – sampling

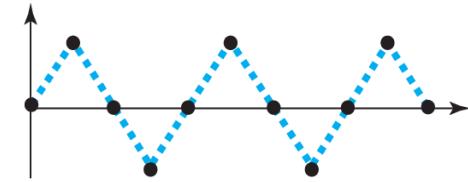
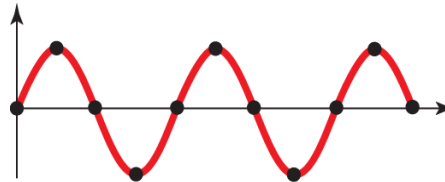
- ◆ 서로 다른 채집율을 가진 정현파 신호의 복구

- ◆ 원래 정현파와 근사 →



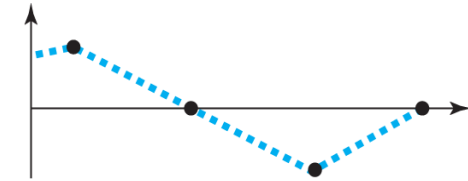
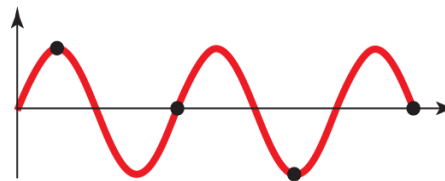
a. Nyquist rate sampling: $f_s = 2 f$

- ◆ 원래 정현파와 근사하지만 4배는 불필요 →



b. Oversampling: $f_s = 4 f$

- ◆ 원래 정현파 재생 못함 →

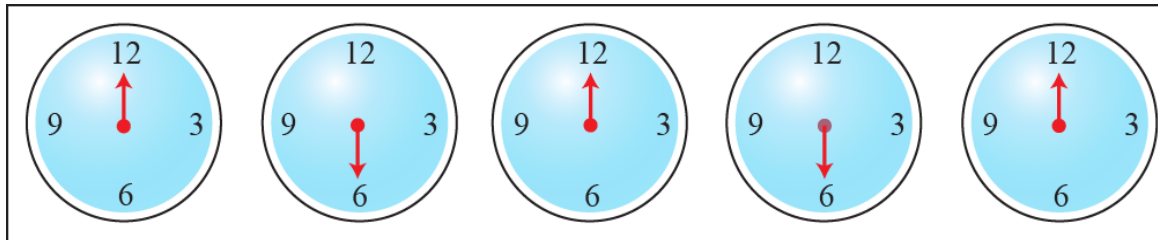


c. Undersampling: $f_s = f$

펄스 코드 변조 – sampling

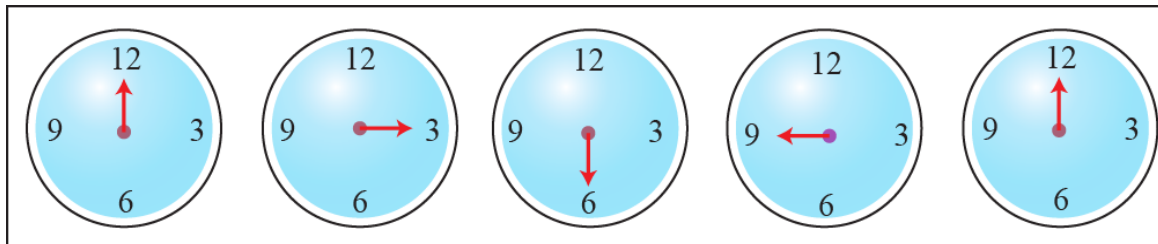
- ◆ [예 4.7] 시계 바늘을 생각해 보자 (그림 4.25). 시계의 분침은 60 초의 주기를 갖는다. 나이퀴스트 정리에 따르면, 매 30초 마다($T_s = T$ 또는 $f_s = 2f$) 채집해야 한다.
 - 그림 (a)에서, 순서대로 채집된 값은 12, 6, 12, 6, 12, 그리고 6이다.
 - 이 값들을 수신한 쪽은 시계가 앞으로 가는지 뒤로 가는지 알 수 없다.
 - 그림 (b)에서, 나이퀴스트 율보다 2배(매 15초)로 채집하였다.
 - 채집된 값은 12, 3, 6, 9, 그리고 12이다. 시계는 앞으로 가는 것이다.
 - 그림 (c)에서, 나이퀴스트 율보다 낮게($T_s = T$ 또는 $f_s = f$) 채집하였다. 채집된 값은 12, 9, 6, 3, 그리고 12이다.
 - 시계가 앞으로 가고 있지만 수신자는 시계가 뒤로 가는 것으로 생각한다.
- ◆ [예 4.8]
 - 영화는 초당 24 프레임이 돌아간다. 만약 자동차 바퀴가 매 초 12번 이상의 속도로 돌고 있는데, 이보다 낮은 속도로 채집을 하면 뒤로 돌고 있는 느낌을 받게 되는 것이다.

펄스 코드 변조 – sampling



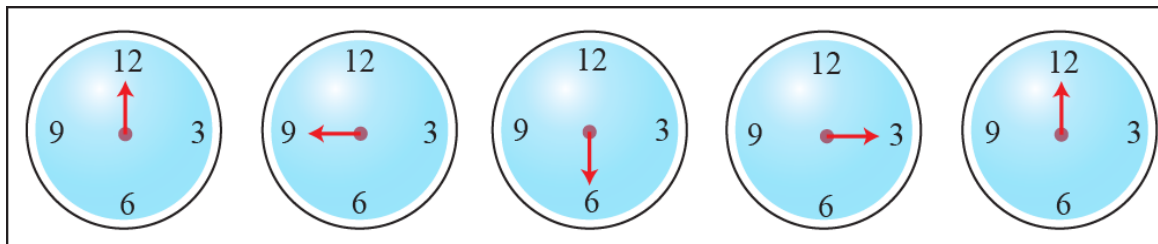
Samples can mean that the clock is moving either forward or backward.
(12-6-12-6-12)

a. Sampling at Nyquist rate: $T_s = T \frac{1}{2}$



Samples show clock is moving forward.
(12-3-6-9-12)

b. Oversampling (above Nyquist rate): $T_s = T \frac{1}{4}$



Samples show clock is moving backward.
(12-9-6-3-12)

c. Undersampling (below Nyquist rate): $T_s = T \frac{3}{4}$

펄스 코드 변조 – quantization

◆ 계수화(quantization)

- 채집된 값에 특정 범위에 속하는 정수 값을 할당하는 방법
- 원래의 아날로그 신호는 V_{min} 과 V_{max} 사이의 진폭 값을 순간적으로 가진다고 가정
- 전체 영역을 높이 Δ (델타)의 L 개의 구간으로 나눔

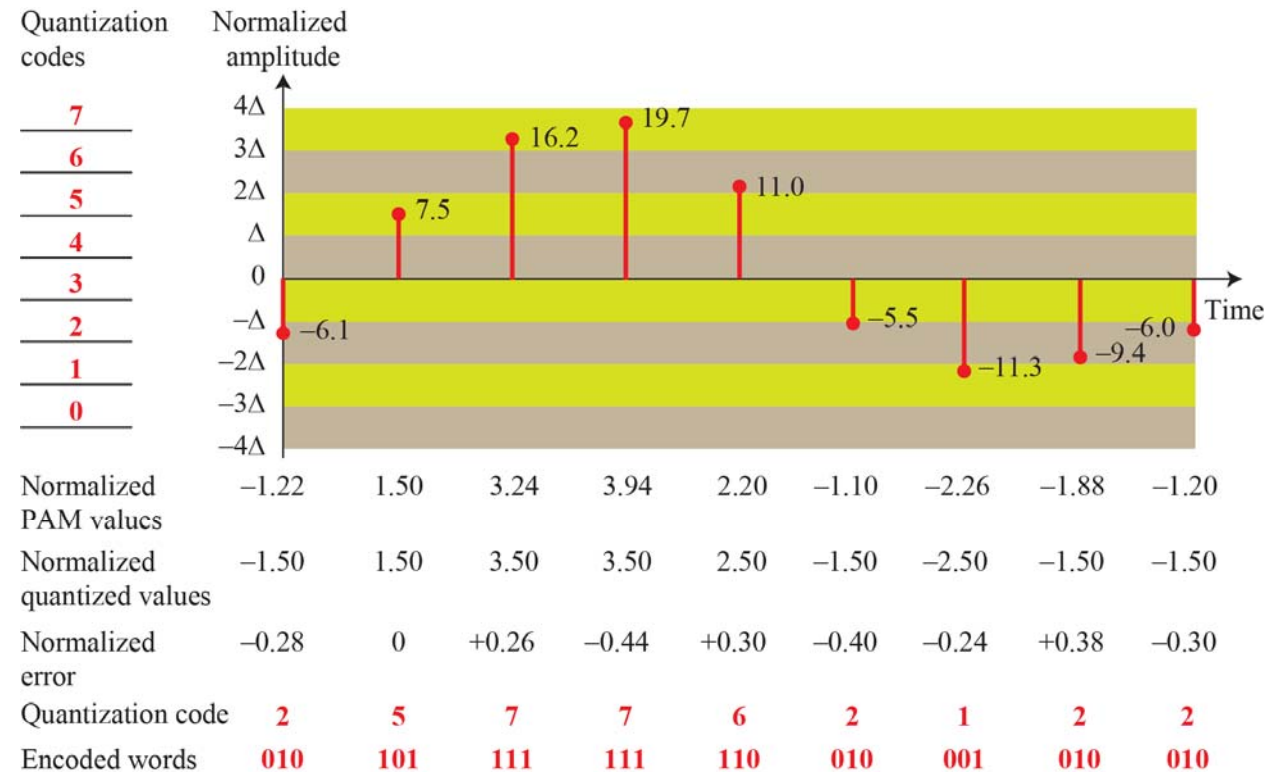
$$\Delta = \frac{V_{max} - V_{min}}{L}$$

➤ 예) -20v와 20v 사이의 진폭을 8개 구간으로 나눌 경우 $\Delta = 5v$

- 각 구간의 중간 지점에 0 부터 $L-1$ 까지의 계수화된 값 지정
- 채집된 신호의 진폭 값을 계수화된 하나의 근사치로 지정

펄스 코드 변조 – quantization

- ◆ 채집된 신호의 계수화와 부호화
 - 채집된 값을 특정 범위의 정량화 코드로 할당 → 계수화 준위
 - 필요한 진폭의 변화에 따라 계수화 준위를 설정
 - 표본당 비트 수 (n_b) = $\log_2 L$



예) 8개의 계수화 준위(L=8) 사용

펄스 코드 변조 – quantization

- ◆ 계수화 오차 (quantization error)
 - 계수화는 근사값을 만드는 과정
 - 출력 값은 구간의 중간 값으로 선택
 - $-\Delta/2 \leq \text{오차} \leq \Delta/2$
- ◆ 계수화 오차 정도는 계수화 준위 개수 또는 표본당 비트 수(n_b)에 의해
 - $\text{SNR}_{\text{dB}} = 6.02(n_b) + 1.76 \text{ dB}$
- ◆ [예 4.13] 전화 가입자 회선은 SNR_{dB} 값이 40이 넘어야 한다. 채집당 최소 비트의 수는 얼마인가?

$$\text{SNR}_{\text{dB}} = 6.02n_b + 1.76 = 40 \rightarrow n = 6.35$$

- 즉, 7비트 이상이어야 함

펄스 코드 변조 – quantization

- ◆ 부호화(encoding) : 각 표본이 n_b 비트의 부호로 바뀜
 - 각 표본에 할당되는 비트의 수는 계수화 준위의 개수 (L)에 의해 결정
($n_b = \log_2 L$)
 - Bit rate = sampling rate x number of bits per sample = $f_s \times n_b$
- ◆ [예 4.14] 사람의 목소리를 디지털화하고자 한다. 채집당 8 비트를 가정하면 비트율은 얼마인가?
 - 비트율 = 표본채집률 X 표본당 비트 수
 - 사람의 목소리는 보통 0에서 4000 Hz사이의 주파수

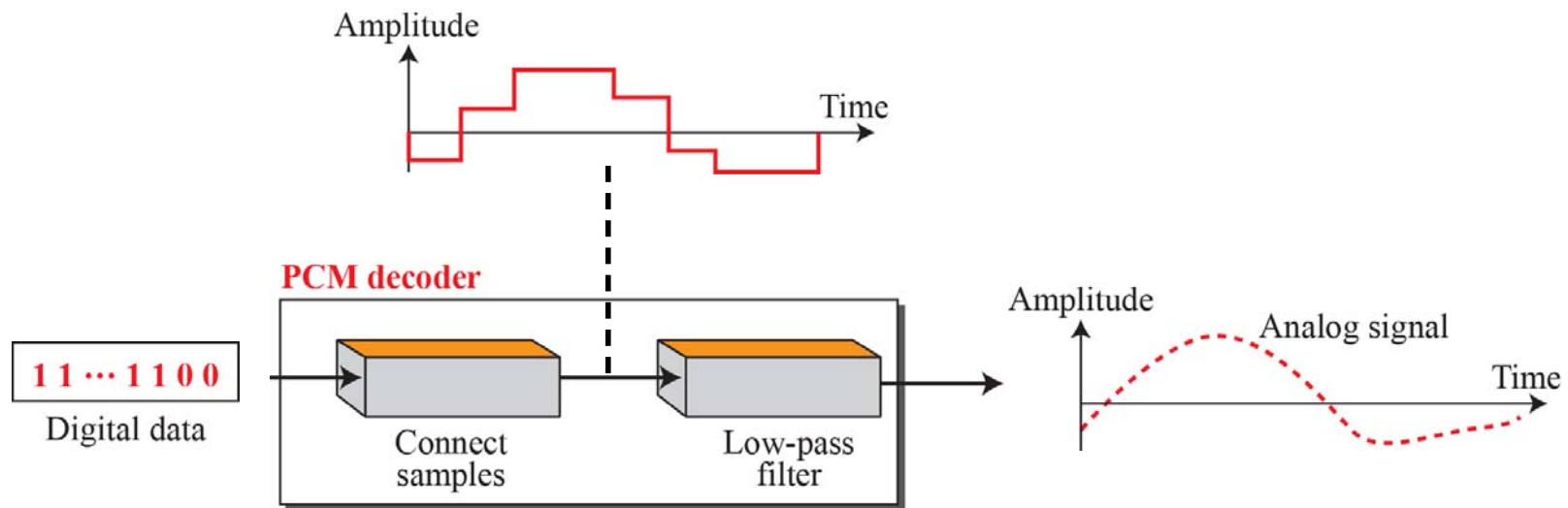
$$\text{Sampling rate} = 4000 \times 2 = 8000 \text{ samples/s}$$

$$\text{Bit rate} = 8000 \times 8 = 64,000 \text{ bps} = 64 \text{ kbps}$$

펄스 코드 변조 - decoder

◆ 신호의 복구

- 수신된 코드를 다음 펄스 까지 일정 값으로 유지하는 펄스로 변환
 - 계단형 신호가 생성
- 저대역 통과 필터를 통해 계단형 신호를 부드러운 아날로그 신호로 변경



펄스 코드 변조 - 대역폭

◆ 회선부호화를 거친 신호의 최소 대역폭

- $B_{\min} = c \times N \times 1/r = c \times f_s \times n_b \times 1/r = c \times n_b \times 2 \times B_{\text{analog}} \times 1/r$

만약 $r=1$ (NRZ 또는 양극 신호), $c=1/2$ (평균) 일 때

- $B_{\min} = n_b \times B_{\text{analog}}$

◆ 채널의 최대 데이터 전송률

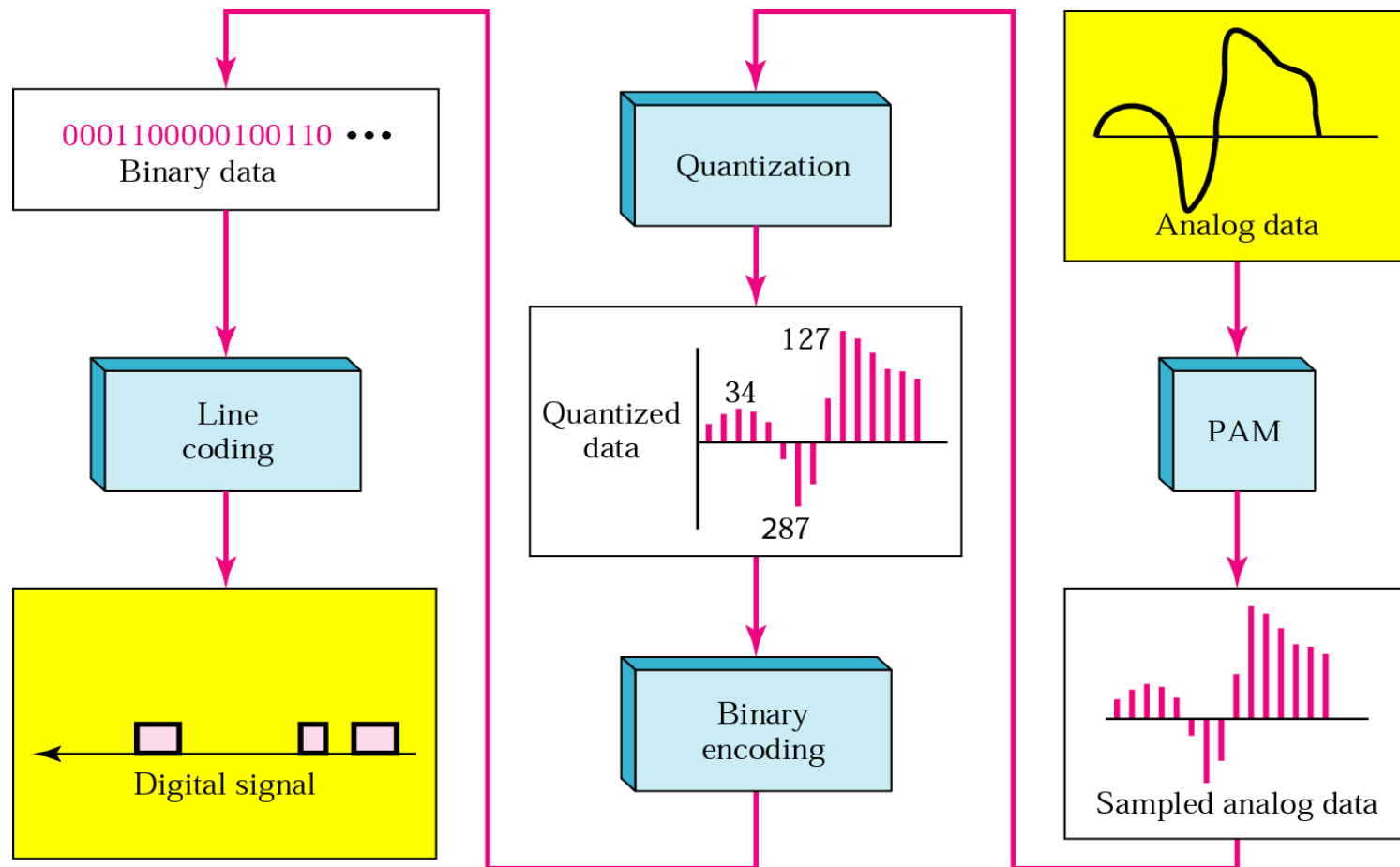
- $N_{\max} = 2 \times B \times \log_2 L \text{ bps}$

◆ 최소 요구 대역폭

- $B_{\min} = N / (2 \times \log_2 L) \text{ Hz}$

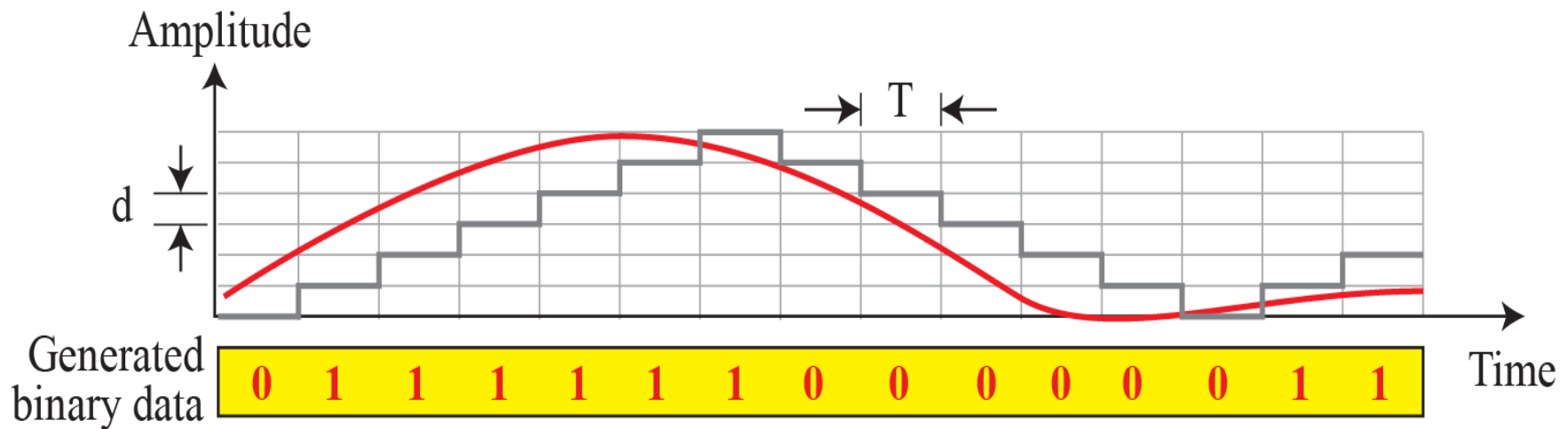
펄스 코드 변조

- ◆ 아날로그 신호에서 PCM 디지털 코드까지



델타 변조

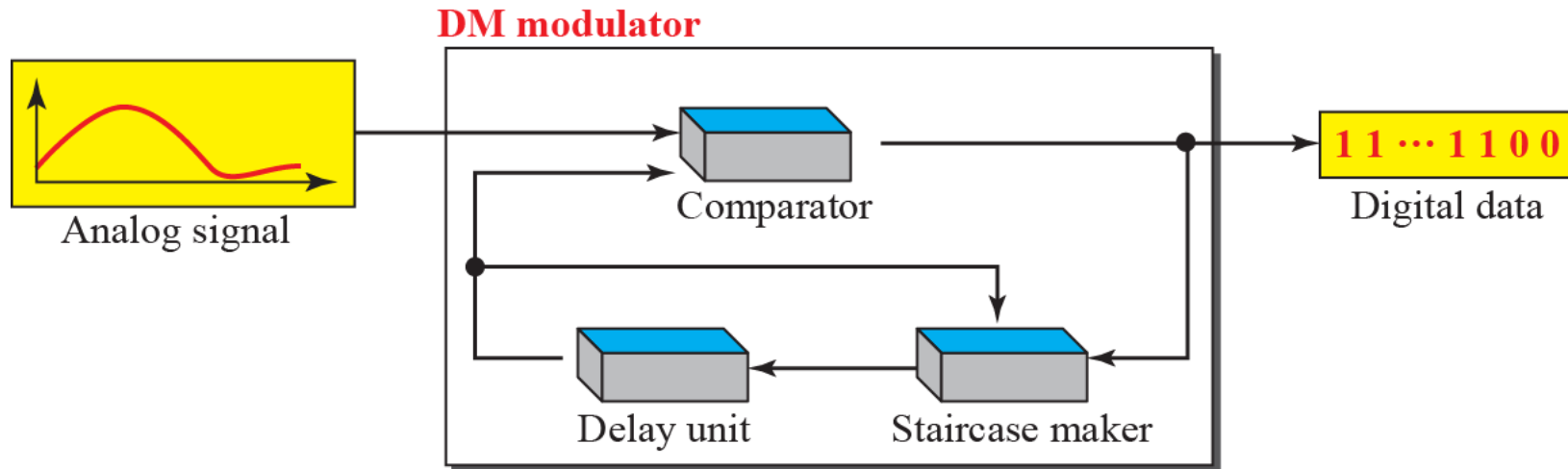
- ◆ 델타 변조 (Delta Modulation, DM)
 - PCM의 복잡도를 낮추기 위해 개발된 기술
 - DM은 직전 표본 값과의 차이를 이용



델타 변조

◆ 변조기

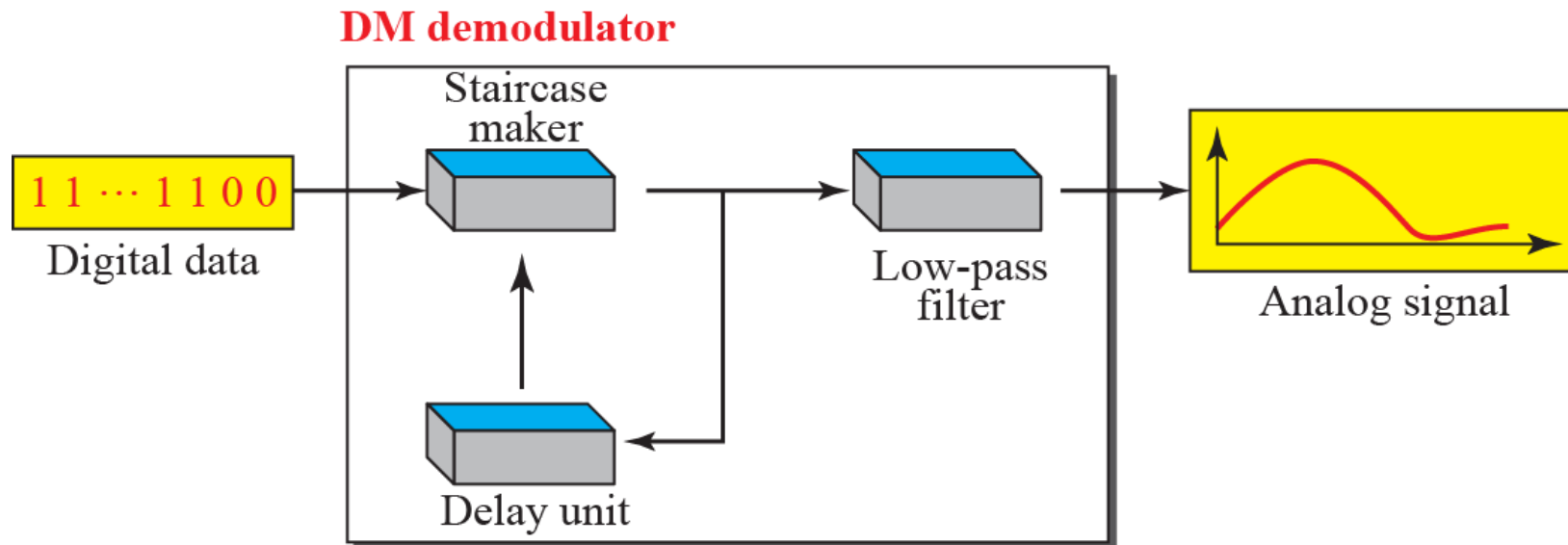
- 아날로그 신호로부터 비트 스트림을 만들기 위해 사용
- 델타라는 작은 양 또는 음의 차이를 기록하는 과정
 - 델타가 양이면 1, 음이면 0



델타 변조

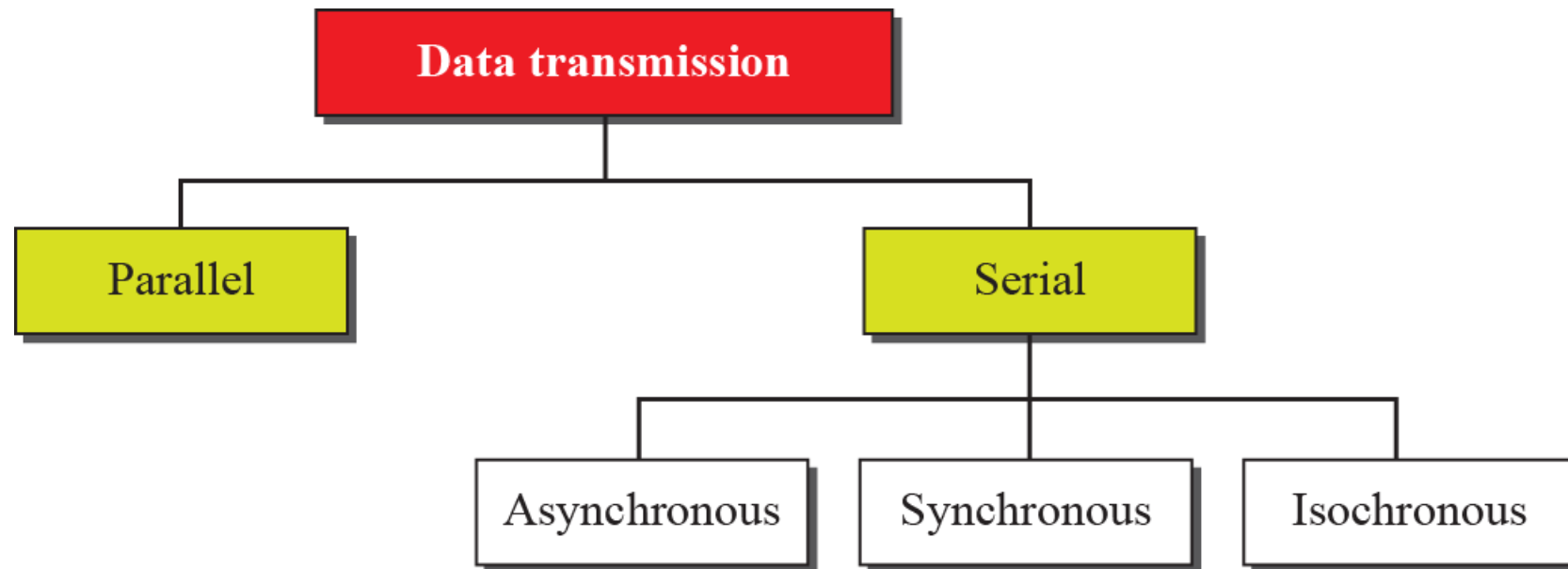
◆ 복조기

- 디지털 신호를 원래의 아날로그 신호로 재생
- 재생된 신호는 저대역-통과 필터를 거침



전송방식

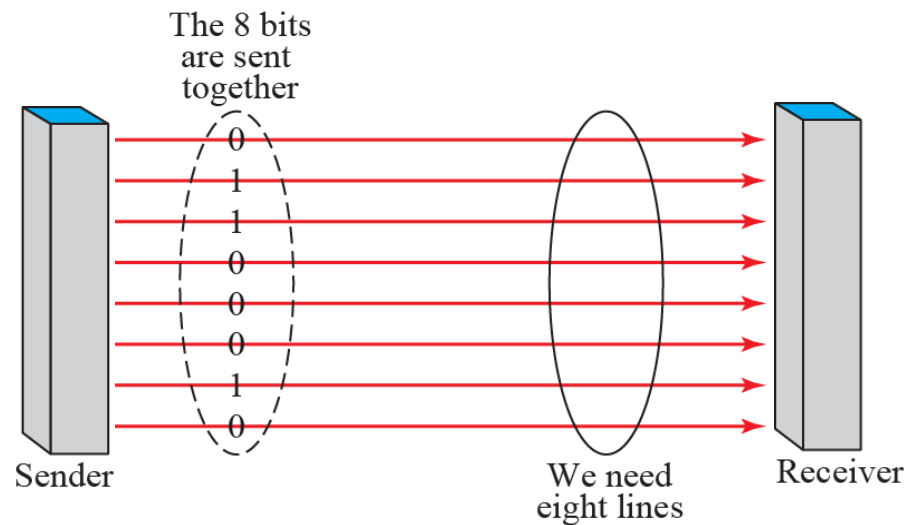
- ◆ 하나의 장치에서 매체로 연결된 다른 장치까지 데이터 전송 방법



전송방식

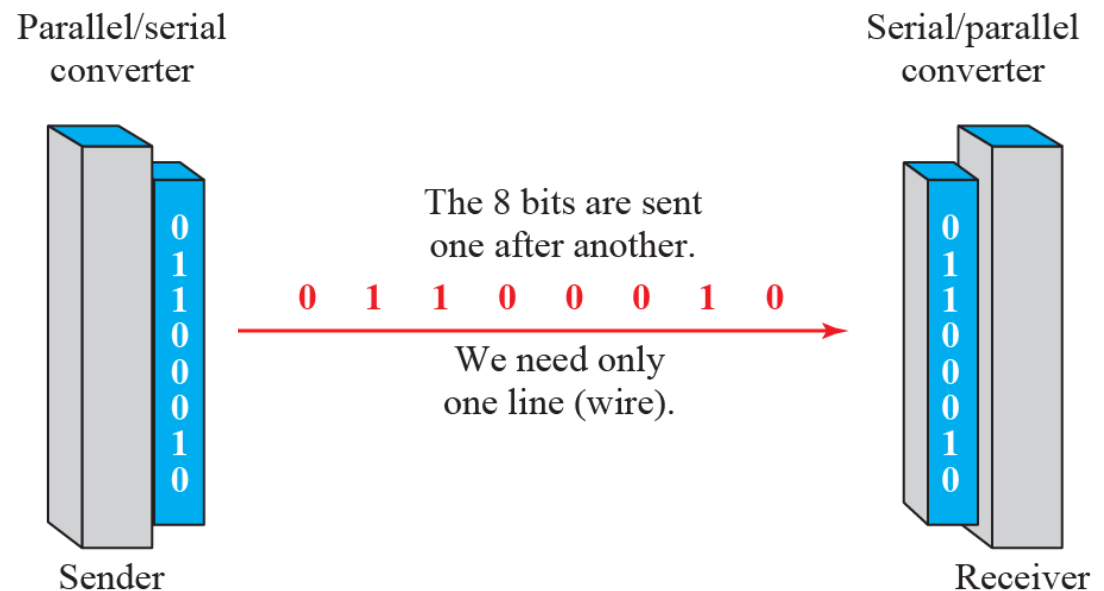
◆ 병렬(parallel) 전송

- 한 비트가 아닌 n비트 그룹으로 데이터를 동시에 전송
- n개의 통신 채널 필요
- 장점 : 속도(speed)
- 단점 : 비용



전송방식

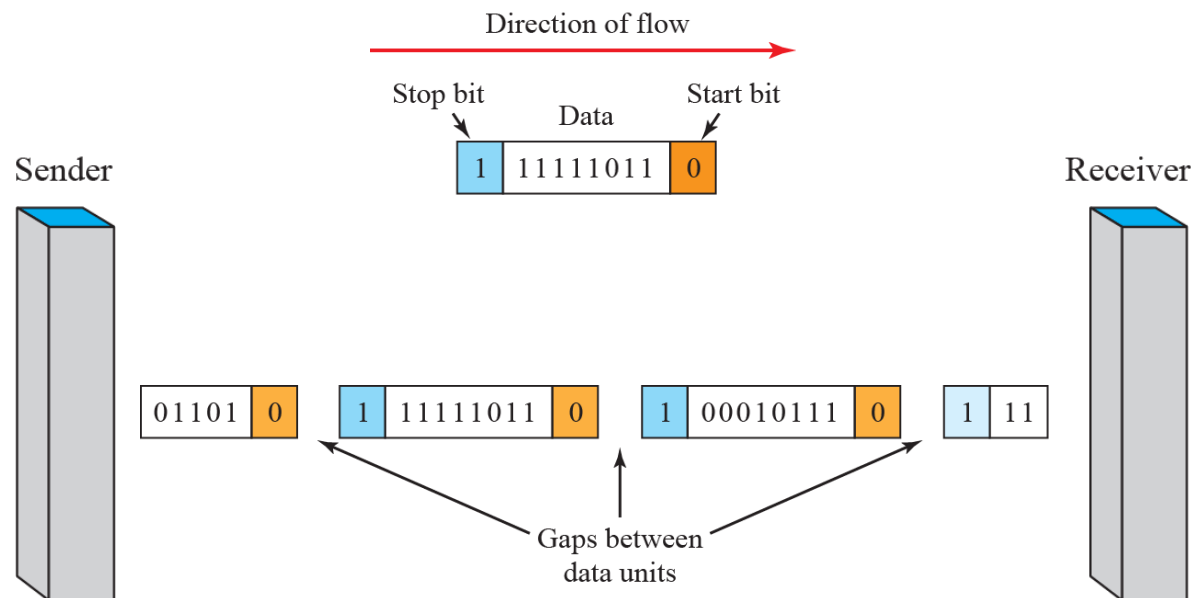
- ◆ 직렬(serial) 전송
 - 한 비트씩 전송
 - 하나의 채널 필요
 - 장점 : 단일 채널, 저가
 - 단점 : 인터페이스 변환기 요구
- ◆ 동기식, 비동기식 또는 등시식 방법으로 구성



전송방식

◆ 비동기(asynchronous) 전송

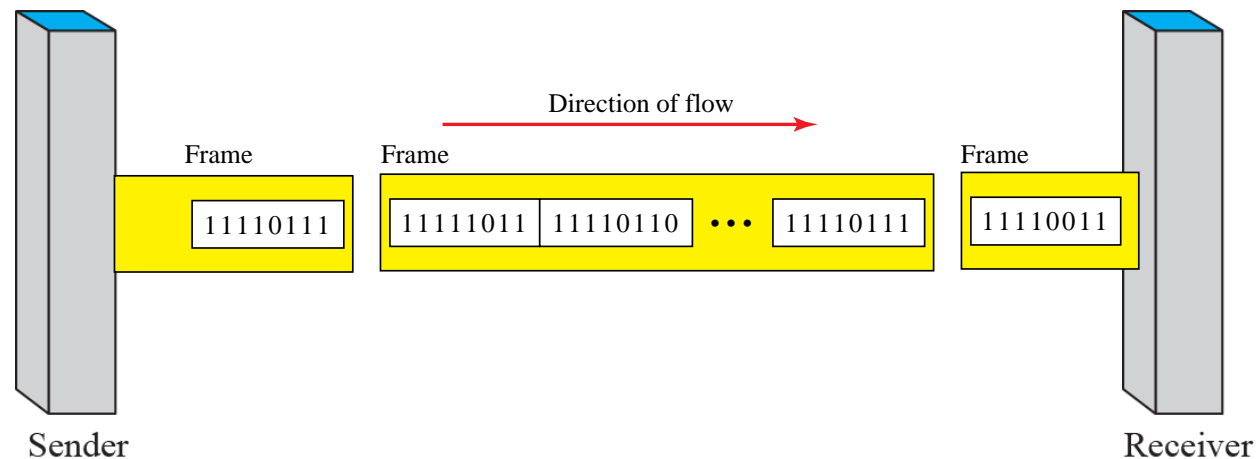
- 바이트 단위의 비동기 전송을 의미
 - 비트들은 동기화 되고 지속시간이 일정함
- 신호의 타이밍을 고려하지 않고 합의된 패턴으로 정보를 송수신
- 각 바이트의 시작에는 **시작 비트(0)**와 끝부분에는 **종료 비트(1)** 전송
- 수신 장치는 각각의 수신 바이트마다 재동기화




전송방식

◆ 동기전송(synchronous)

- 다수의 바이트로 구성(프레임)되고, 바이트와 바이트 사이의 간격이 없음
- 수신자가 복호화를 위한 바이트를 비트로 분리
 - 0과 1로 끊임없이 수신된 문자열을 수신자가 재구성
- 장점
 - 비동기식에 비해 속도가 빠르다
 - 고속 응용에 유리
 - 데이터 링크 계층에서 이루어짐





눈으로 남을 볼 줄 아는 사람은 훌륭한 사람이다.

그러나 귀로는 남의 이야기를 들을 줄 알고,

머리로는 남의 행복에 대해서 생각할 줄 아는

사람은 더욱 훌륭한 사람이다



Question?