

데이터 통신과 네트워킹

6장 대역폭 활용

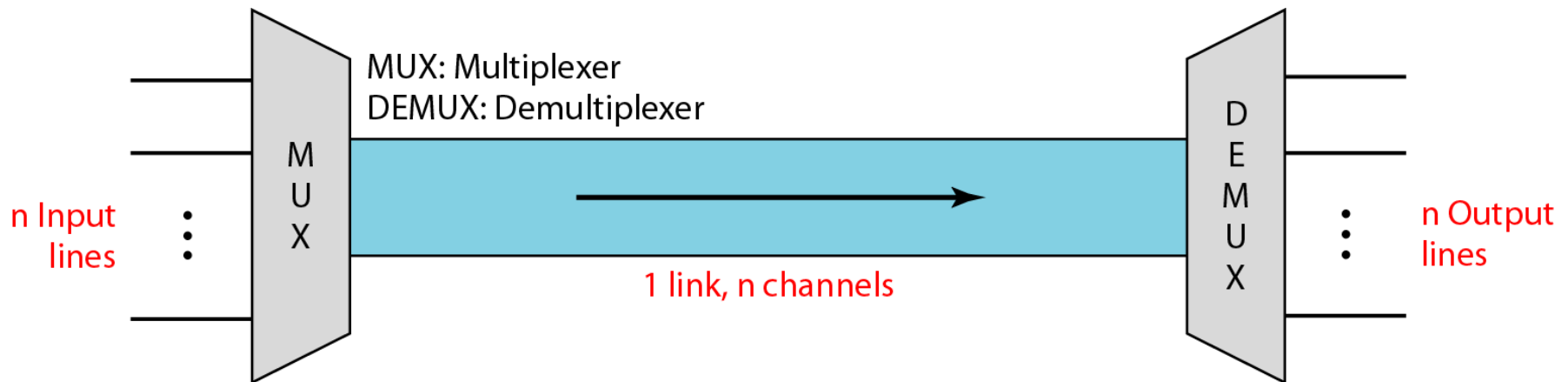
최 홍 인

6장 대역폭 활용

1. 다중화
2. 대역 확산

1. 다중화

- ▶ 다중화(multiplexing)는 단일 링크를 통하여 여러 개의 신호를 동시에 전송할 수 있도록 해주는 기술
- ▶ 그림 6.1은 다중화 시스템 기본 형식을 보여준다.



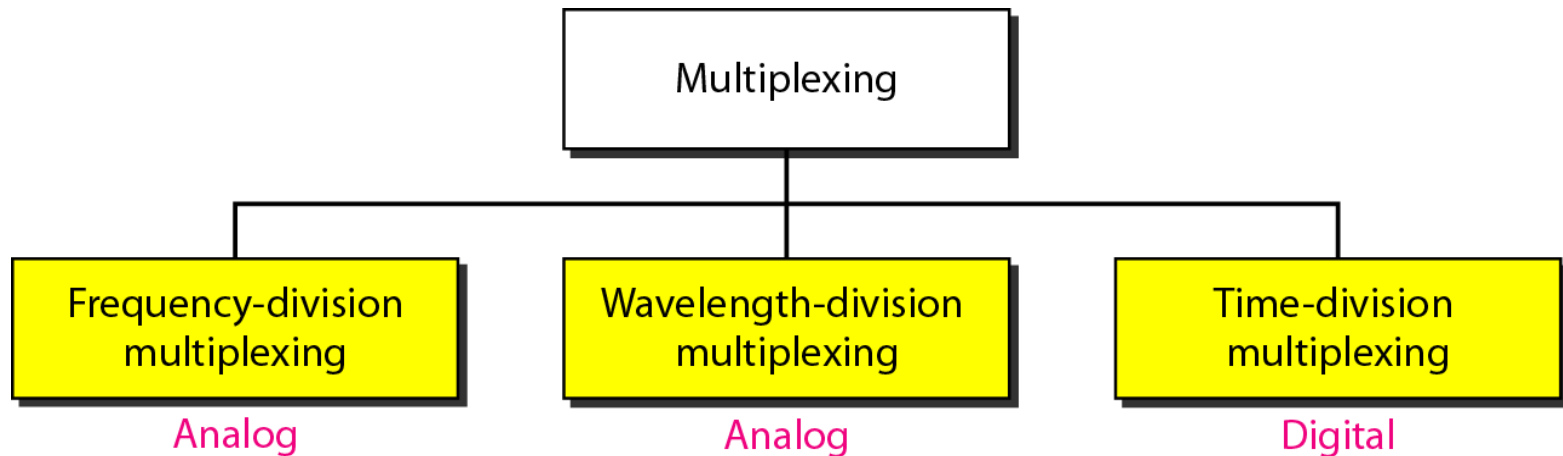
다중화

- ▶ 다중화기(MUX, Multiplexer)
 - 전송 스트림을 단일 스트림으로 결합(many to one)
- ▶ 다중복구기(DEMUX, Demultiplexer)
 - 스트림을 각각의 요소로 분리(one to many)
 - 전송 스트림을 해당 수신장치에 전달
- ▶ 링크(Link)
 - 물리적인 경로
- ▶ 채널(Channel)
 - 한 쌍의 장치간에 전송을 위한 하나의 경로

다중화

▶ 다중화의 범주

1. 주파수 분할 다중화(Frequency-Division Multiplexing)
2. 파장 분할 다중화(Wavelength-Division Multiplexing)
3. 시분할 다중화(Time-Division Multiplexing)

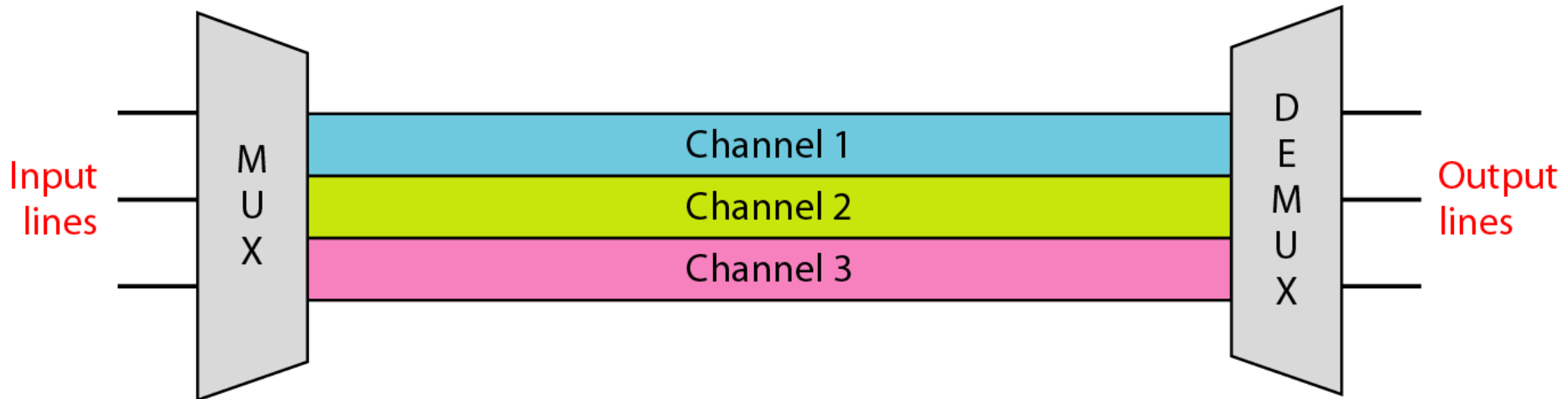


주파수 분할 다중화

- ❖ 주파수 분할 다중화(FDM, Frequency-division Multiplexing)은 아날로그 신호를 조합하는 아날로그 다중화 기술.
- ❖ 링크의 대역폭이 전송되는 조합 신호의 대역폭 보다 클 때 적용할 수 있는 아날로그 기술
- ❖ 신호가 겹치지 않도록 보호대역(guard band)만큼 떨어져 있어야 한다.
- ❖ 그림 6.3은 FDM의 개념을 보여준다.

주파수 분할 다중화

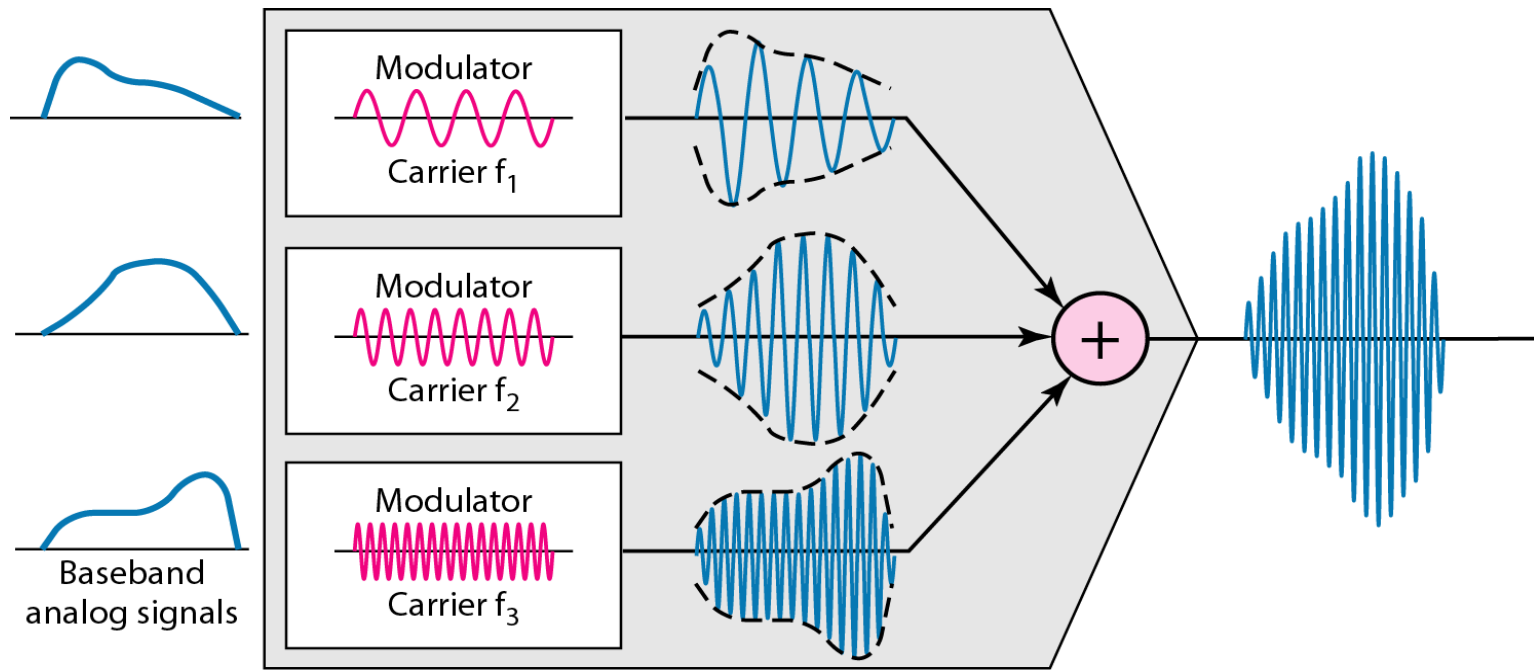
❖ 그림 6.3 주파수 분할 다중화



주파수 분할 다중화

❖ 다중화 과정

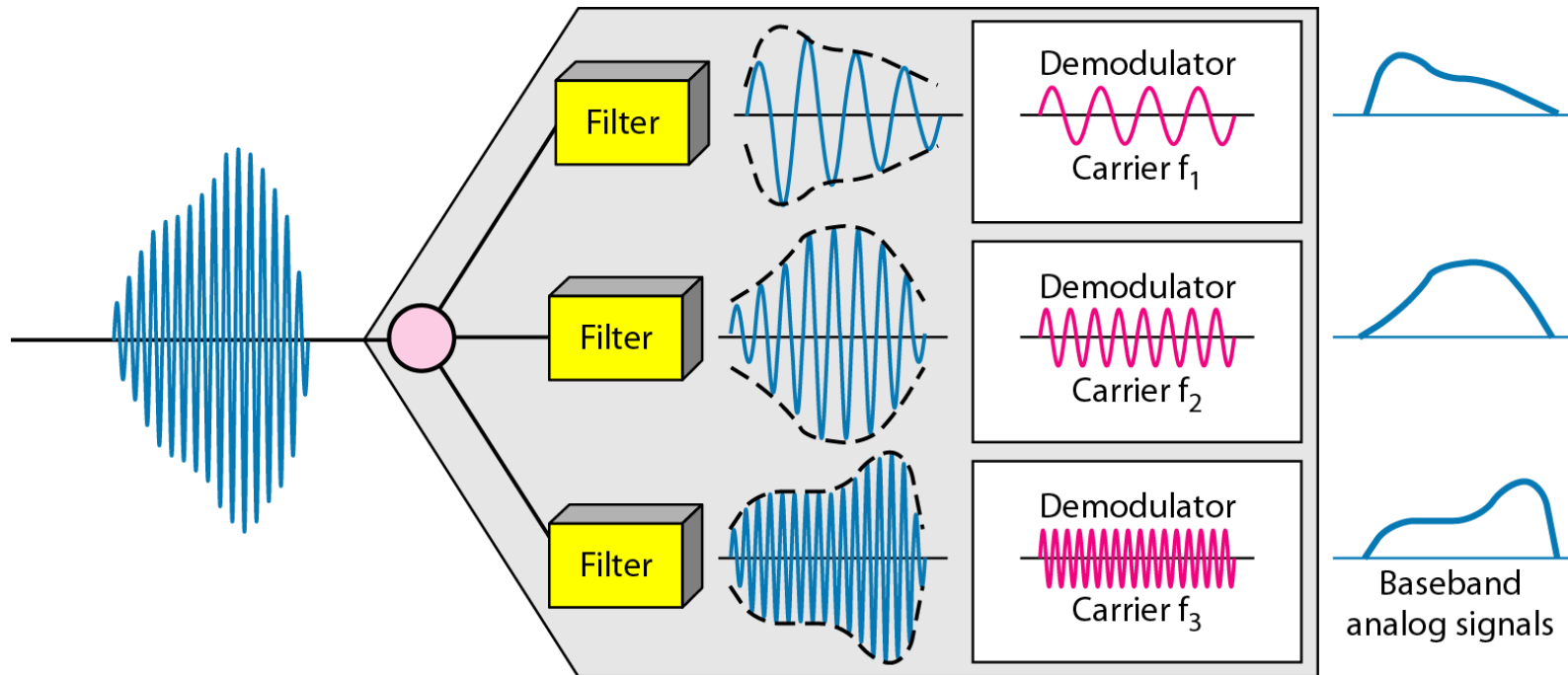
- 각 전화기는 비슷한 범위의 주파수 대역의 신호 발생
- 이 신호는 서로 다른 반송 주파수로 변조된다(f_1, f_2, f_3)



주파수 분할 다중화

❖ 다중화 복구 과정

- 개개의 신호를 분리하여 수신기에 전달



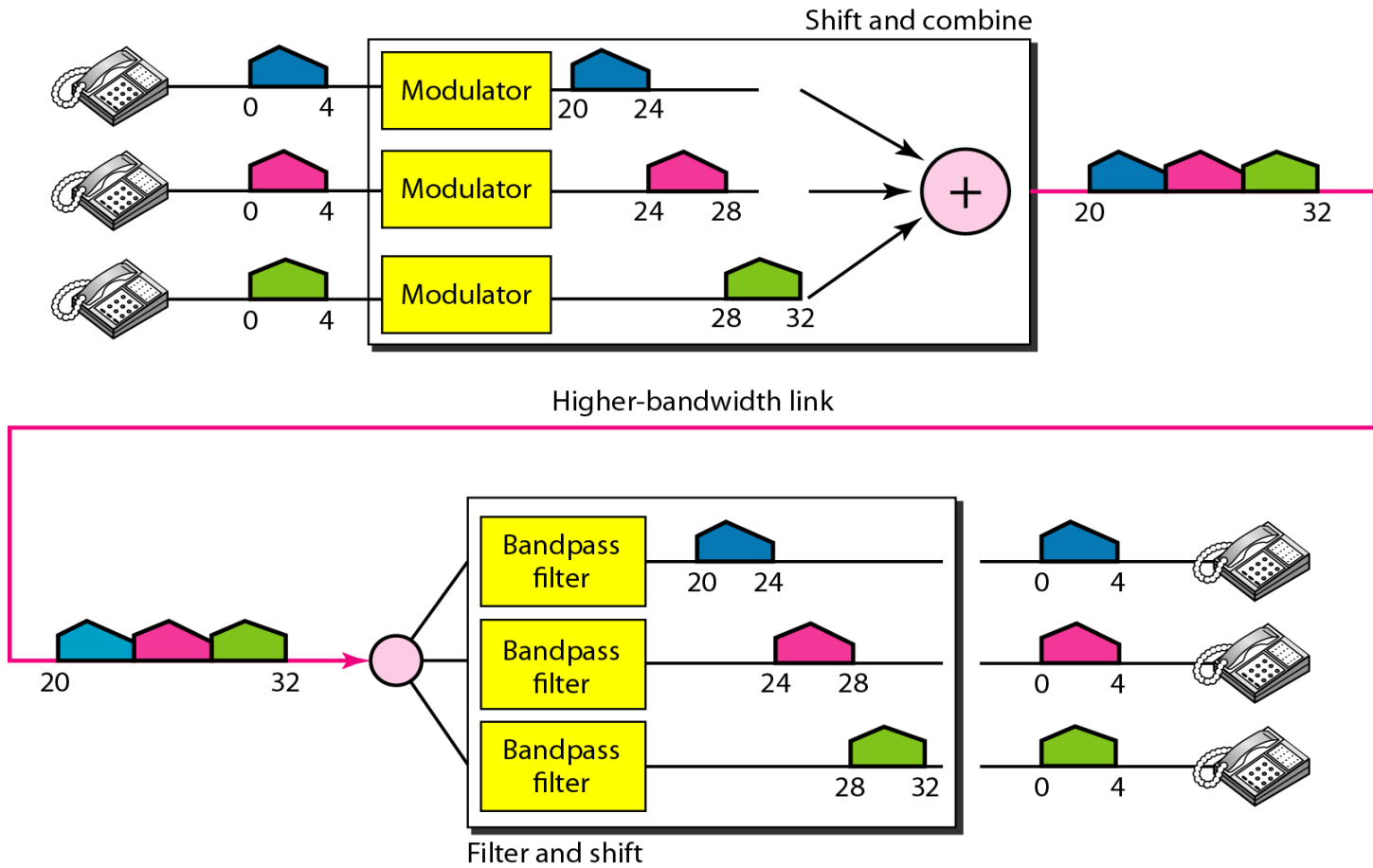
예제 6.1

- ▶ 음성 채널이 4kHz의 대역폭을 차지한다고 가정하자. 주파수 20kHz에서부터 32kHz에 걸친 대역폭을 사용하는 링크를 통해 3개의 음성 채널을 합해서 보낸다고 하자. 주파수 영역을 이용하여 구성을 나타내어라. 보호 대역은 없는 것으로 간주한다

▶ *Solution*

- 그림 6.6에서와 같이 3개의 음성 채널을 서로 다른 대역폭으로 이동(변조)시킨다. 첫 번째 채널에는 20~24kHz의 대역을, 두 번째 채널에는 24~28kHz의 대역을, 그리고 세 번째에는, 28~32kHz의 대역을 사용한다. 그 후에 그림과 같이 조합한다. 수신 측에서는 필터를 통해 각자의 신호만 받는다.

그림 6.6 예제 6.1



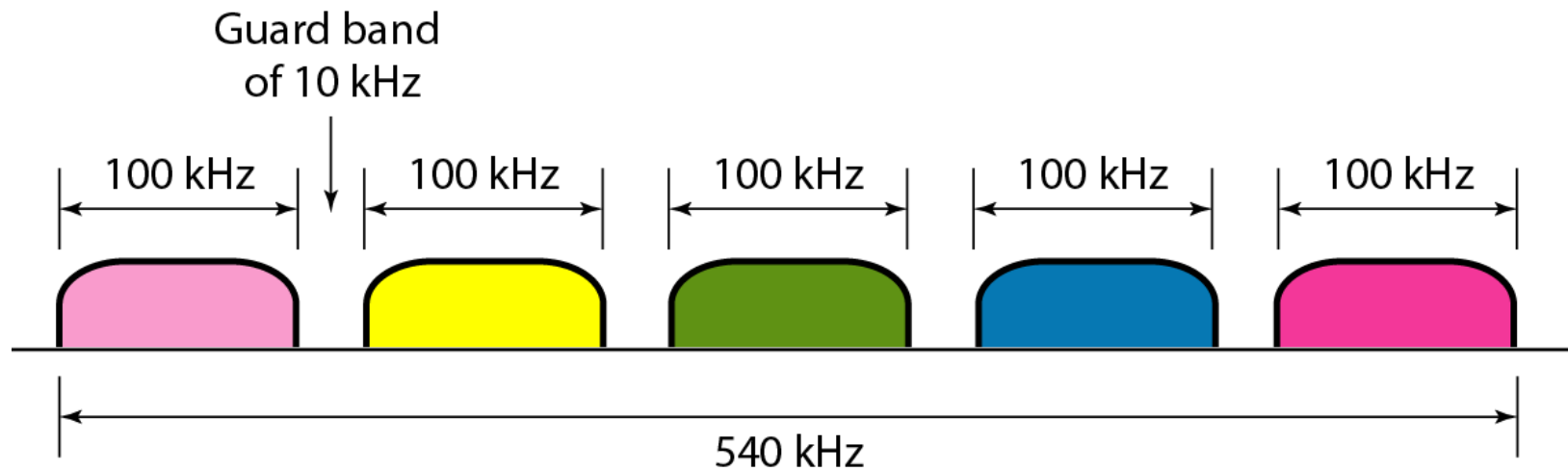
예제 6.2

- ▶ 각각 100kHz의 대역폭을 갖는 5개의 채널을 함께 다중화해서 보낸다. 만일 서로 간의 간섭을 피하기 위해 채널 사이에 10kHz의 보호 대역이 필요하다면 최소 얼마만큼의 대역폭이 필요한가?

▶ *Solution*

- 5개의 채널의 간섭을 피하기 위해 최소 4개의 보호 대역이 필요하다. 따라서 필요한 대역폭은 그림 6.7에 보인 것처럼 최소 $5 \times 100 + 4 \times 10 = 540 \text{ kHz}$, 이다.

그림 6.7 예제 6.2



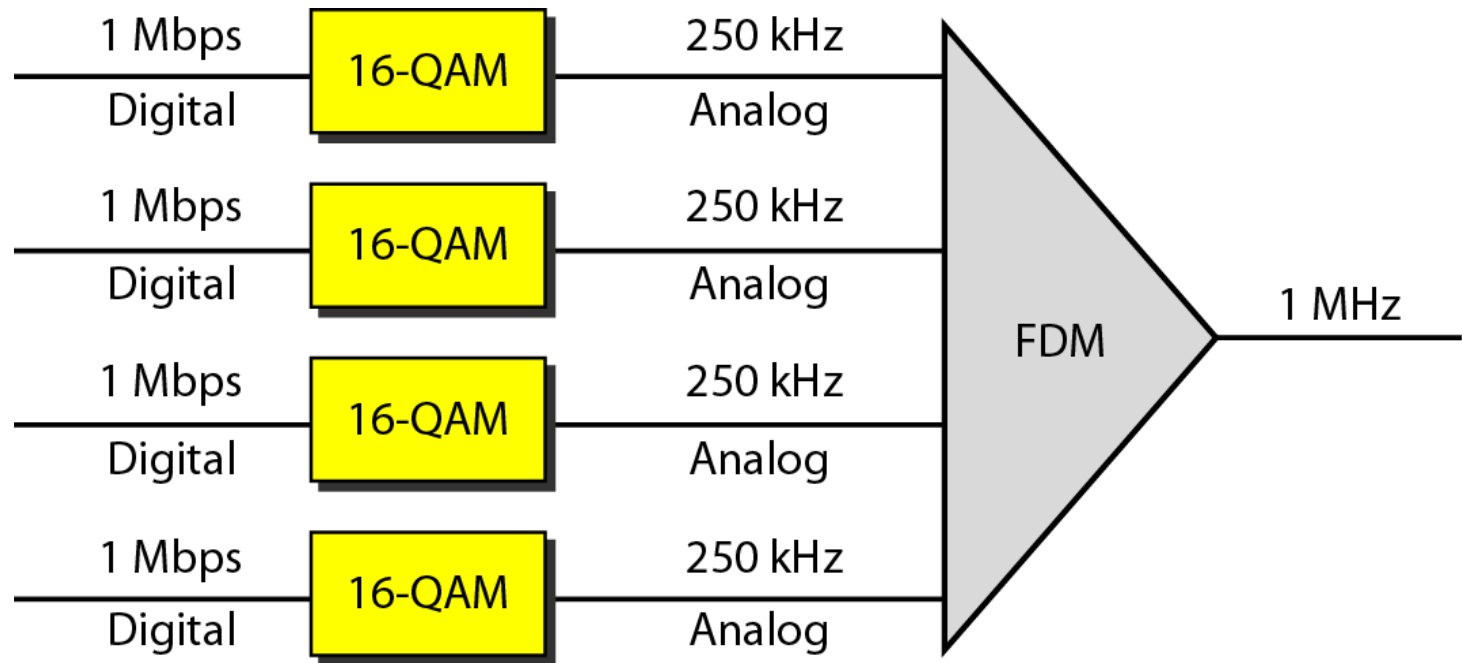
예제 6.3

- ▶ 4개의 디지털 데이터 채널이 1MHz의 채널을 사용하는 위성을 사용하여 각각 1Mbps의 전송속도로 전송한다. FDM을 사용하여 적절한 구성을 설계하라.

▶ *Solution*

- 위성채널은 아날로그이다. 각각 250kHz 대역폭을 갖는 4개의 채널로 나눈다. 1Mbps의 디지털 채널을 4비트가 1Hz에 해당되도록 변조한다. 16-QAM 변조가 한 가지 방법이다. 그림 6.8이 가능한 한 가지 구성을 보여준다.

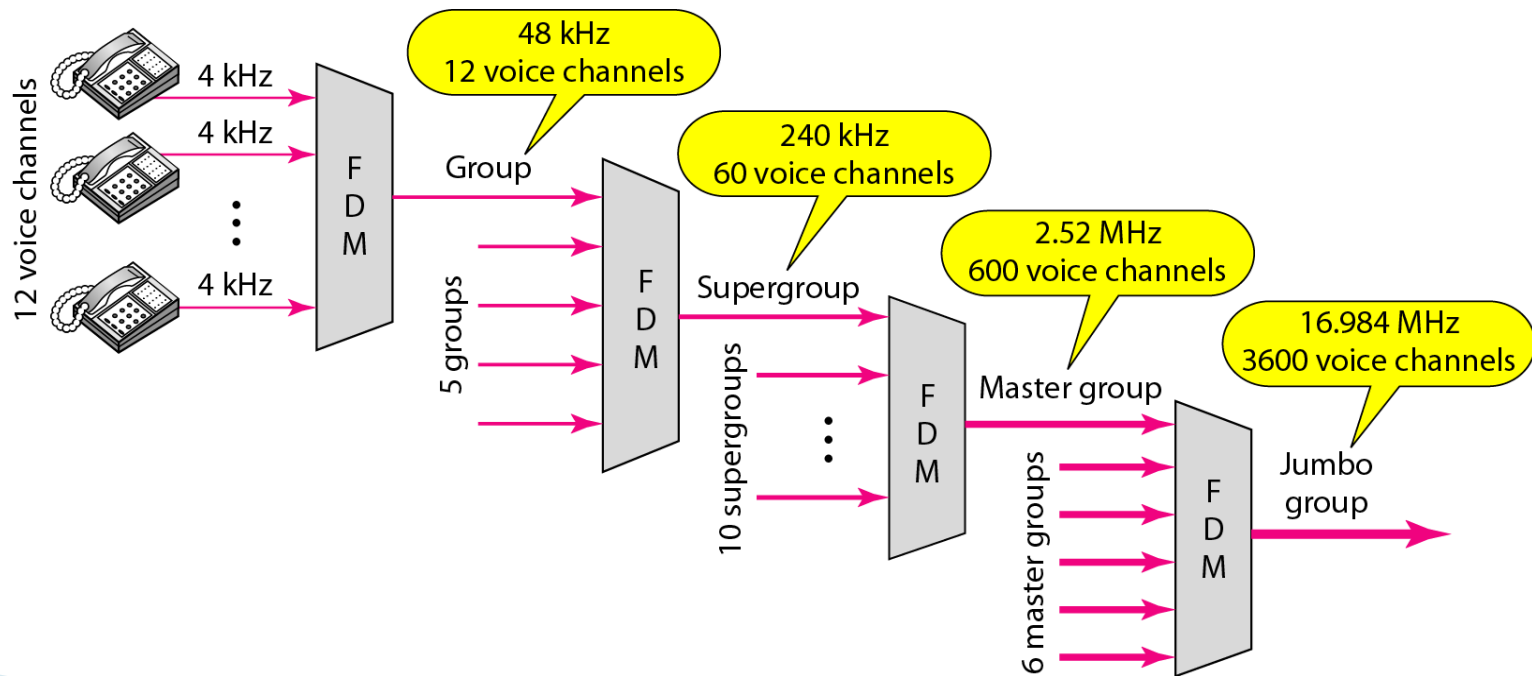
그림 6.8 예제 6.3



주파수 분할 다중화

▶ 아날로그 반송파 구조

- 전화회사는 효율성을 위해 낮은 대역폭 회선들을 높은 대역폭의 회선들로 다중화해 왔다.



주파수 분할 다중화

❖ FDM의 다른 응용

- 라디오
 - AM : 방송국당 10 kHz
 - FM : 방송국당 200 kHz
- TV : 채널당 6MHz
- 1 세대 이동전화 : 3kHz의 대역을 갖는 음성신호가 FM을 사용하여 변조된다. 사용자마다 60 kHz (수신 : 30 kHz, 발신 : 30 kHz)

예제 6.4

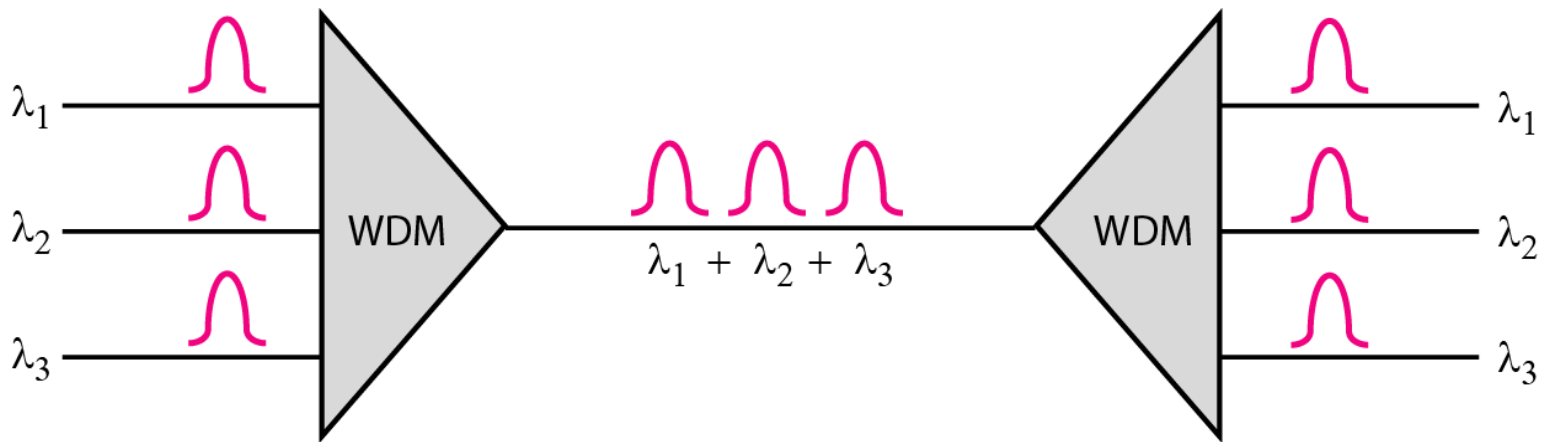
- ▶ 고급 이동 전화 시스템(AMPS, Advanced Mobile Phone System)은 2개의 대역을 사용한다. 첫 번째 대역은 824에서 849MHz로 전송에 사용되고, 869에서 894MHz는 수신에 사용된다. 3kHz의 음성 신호가 FM 변조를 통해 30kHz의 변조 신호를 만들어낸다. 얼마나 많은 사람이 동시에 이동전화를 사용할 수 있는가?

▶ *Solution*

- 각 대역은 25MHz이다. 25MHz를 30kHz로 나누면 833.33이 나온다. 실제로는 이 대역을 832채널로 나눈다. 이 중에서 42개의 채널은 제어 목적으로 사용되고 이동 전화 사용자들에게는 790개의 채널이 제공된다.

파장 분할 다중화

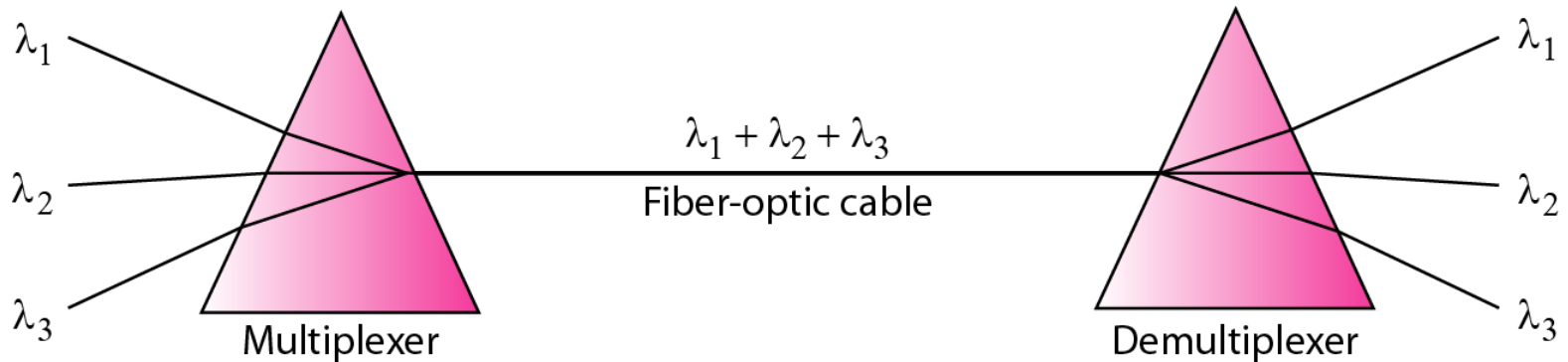
- ❖ 파장 분할 다중화(WDM, Wavelength Division Multiplexing)의 기본 개념은 FDM과 같으며, 광섬유의 고속 전송률을 이용하기 위해 설계
- ❖ 그림 6.10은 WDM의 기본개념을 보여준다.



파형 분할 다중화

❖ WDM

- 다중 빛 소스를 단일 빛으로 결합
- 단일 빛은 다중 빛 소스로 분리
- 프리즘 이용 : 임계각과 주파수 기반
- 그림 6.11은 파장 분할 다중화와 다중화 복구를 위한 프리즘을 보여준다.

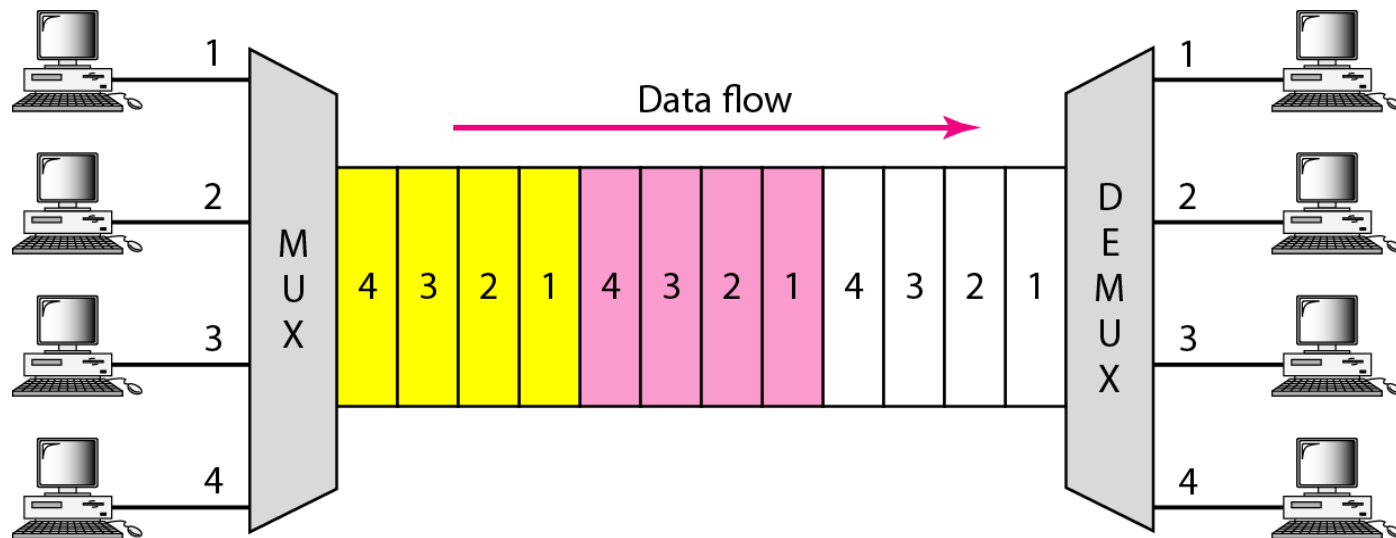


시분할 다중화

- ▶ 시분할 다중화(TDM, Time Division Multiplexing)는 링크의 높은 대역폭을 여러 연결이 공유할 수 있게 하는 디지털 과정이다.
- ▶ FDM에서 대역의 일부를 공유하는 대신에 TDM은 시간을 공유한다.
- ▶ 그림 6.12는 TDM의 개념을 보여준다.
- ▶ TDM은 동기(synchronous) 방식과 통계(statistical) 방식으로 나눌 수 있다.

시분할 다중화

그림 6.12 TDM



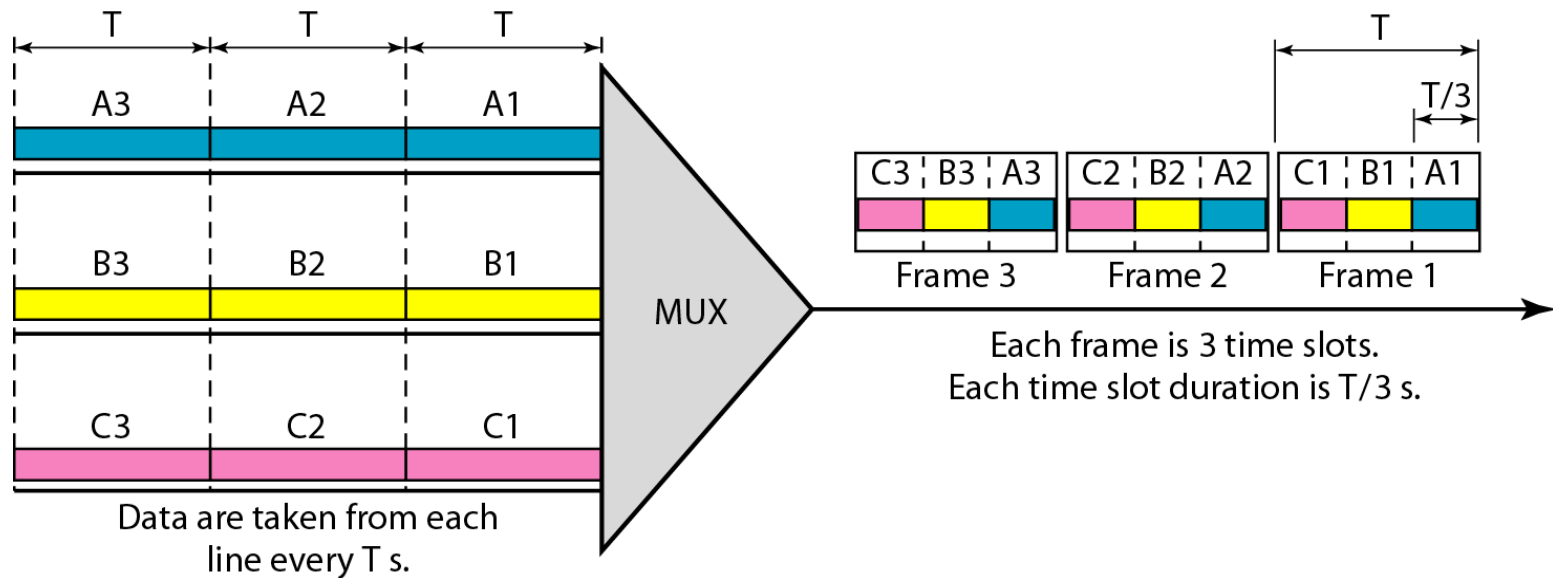
시분할 다중화

▶ 동기 TDM

- 동기 TDM에서는 각 입력 연결은 데이터가 없어도 해당 출력 창구를 갖게 된다.
- 시간슬롯(또는 틱새)와 프레임
 - 동기 TDM에서는 n 개의 연결로부터 데이터를 전송하는 링크의 전송률은 데이터 흐름을 보장하기 위해 각 연결의 전송률의 최소 n 배가 되어야 한다.
 - 비트 기간과 비트 전송률이 서로 역이기 때문에 한 연결의 한 단위 기간은 한 프레임의 시간슬롯의 기간의 n 배가 된다.
 - 그림 6.13은 그 예를 보여준다.

시분할 다중화

그림 6.13 동기식 시분할 다중화



예제 6.5

- ▶ 그림 6.13에서의 각 입력 연결의 데이터 전송률은 1kbps 이다. 한 번에 한 비트씩 다중화 된다고 하면(단위는 비트이다), (1) 각 입력 슬롯, (2) 각 출력 슬롯, (3) 각 프레임의 기간은 얼마인가?

▶ *Solution*

- 다음과 같이 답할 수 있다.
 - 1) 각 입력 연결의 데이터 전송률은 1kbps이다. 이것은 비트 기간이 $1/1000\text{s}$ 또는 1ms 를 의미한다. 입력시간 슬롯기간은 1ms 이다.
 - 2) 각 출력 시간슬롯의 기간은 입력 시간 슬롯의 $1/3$ 이다. 즉 출력 시간슬롯은 $1/3\text{ms}$
 - 3) 각 프레임은 3개의 출력시간을 운반한다. $3 \times 1/3\text{ms}$ 또는 1ms 이다.

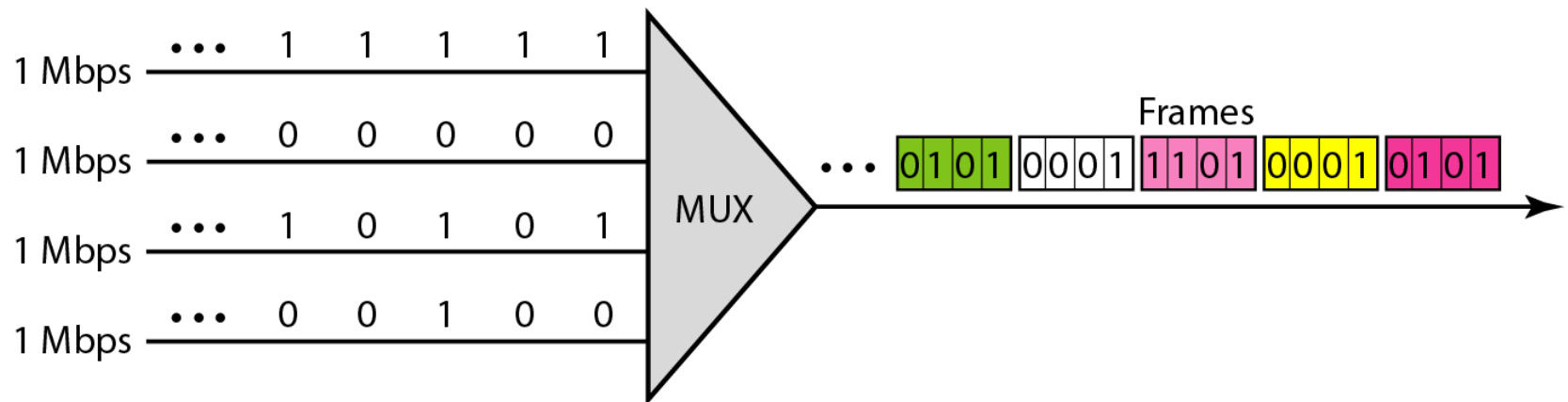
예제 6.6

- ▶ 그림 6.14는 각 입력에서 오는 데이터와 출력 데이터 스트림을 갖는 동기 TDM을 보여준다. 데이터 단위는 1비트이다. (1) 입력 비트 기간, (2) 출력 비트 기간, (3) 출력 비트율, (4) 출력 프레임율을 구하라.

▶ *Solution*

- 다음과 같이 답할 수 있다.
 - 1) 입력 비트 기간은 비트율의 역이므로 $1/1\text{Mbps} = 1\mu\text{s}$.
 - 2) 출력 비트 기간은 입력 비트 기간의 $1/4$ 이므로 $1/4\mu\text{s}$.
 - 3) 출력 비트율은 출력 비트 기간의 역이므로 4Mbps이다. 이는 출력 비트율이 입력 비트율의 4배라는 점에서부터 알 수 있다.
 - 4) 프레임율은 항상 입력 비트율과 같다. 따라서 프레임율은 매 초 1,000,000프레임이다.

그림 6.14 예제 6.6



예제 6.7

- ▶ 4개의 1kbps 연결이 다중화되고 있다. 각 단위는 1비트이다. (1) 다중화 전의 1비트의 기간, (2) 링크의 전송속도, (3) 시간 슬롯의 기간, (4) 프레임의 기간.

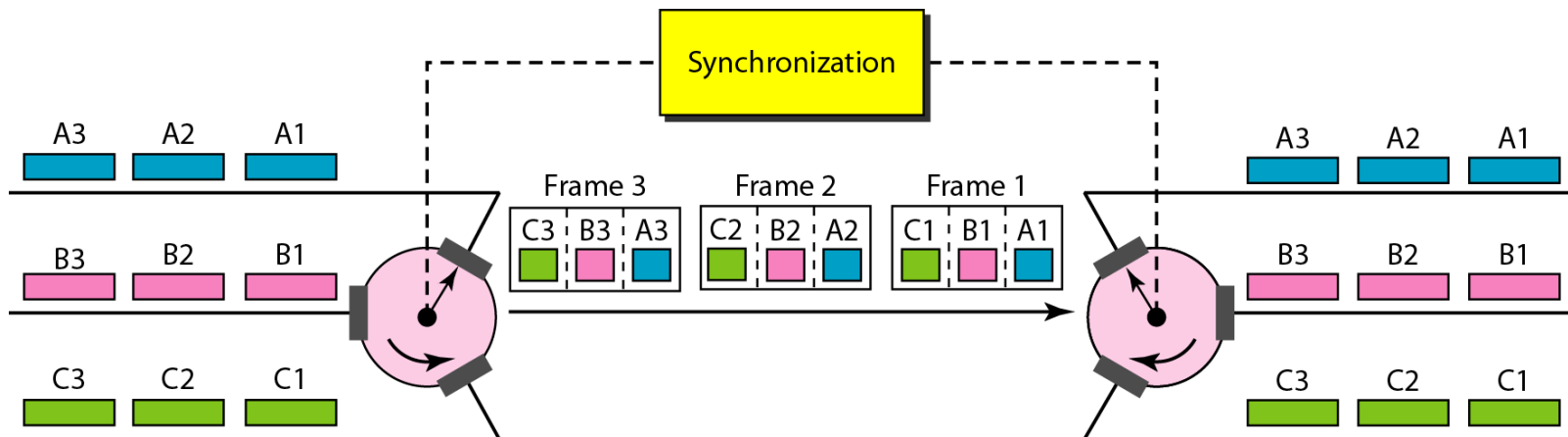
▶ *Solution*

- 다음과 같이 답할 수 있다.
 - 1) 다중화 전의 1비트의 기간은 $1/1\text{kbps} = 1\text{ms}$.
 - 2) 링크의 전송 속도는 각 연결에서의 전송 속도의 4배이므로 4kbps.
 - 3) 각 시간 슬롯의 기간은 다중화 전의 각 비트의 기간의 $1/4\text{ms}$ 또는 $250\mu\text{s}$ 이다. 이는 링크의 전송속도에서 계산할 수 있다.
 - 4) 프레임 기간은 항상 다중화 이전의 입력 비트율과 같아서 1ms이다.

시분할 다중화

❖ 끼워넣기(interleaving)

- 스위치 장치들은 일정한 비율로 빠르게 정해진 순서대로 이동하여 연결을 열어 전송 기회를 부여한다.
- 그림 6.15는 6.13에 보인 연결에 대한 끼워넣기 과정을 보여준다.



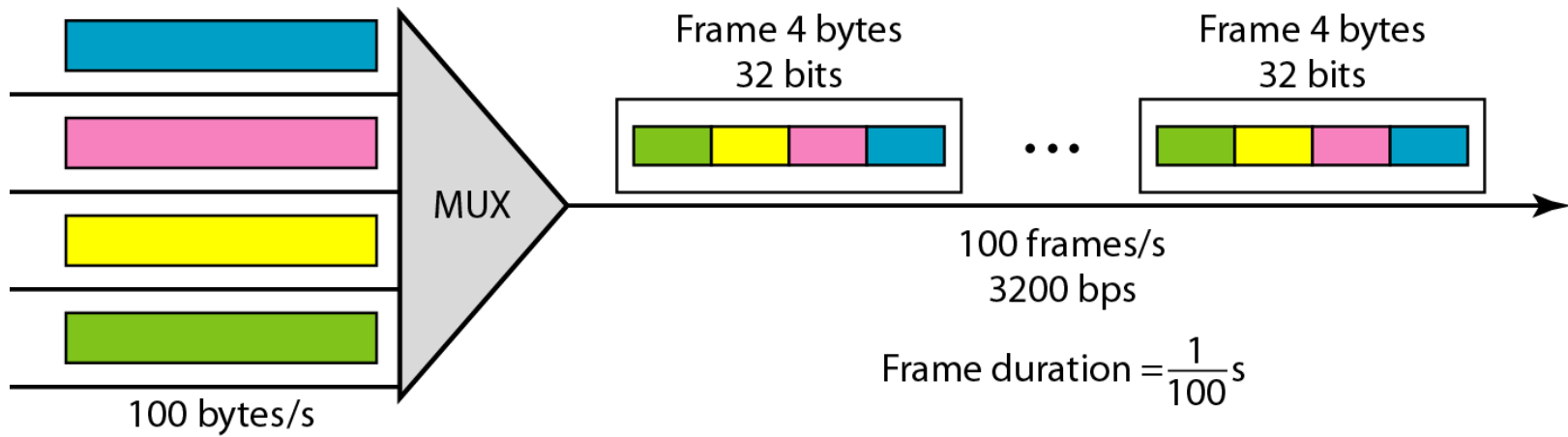
예제 6.8

- ▶ TDM을 사용하여 4개의 채널을 다중화한다. 각 채널이 100byte/s의 속도로 전송하고 각 채널마다 1바이트씩 다중화하는 경우에 대해 링크 상에 움직이는 프레임의 크기와 프레임의 기간과 프레임 속도와 링크의 비트 전송률을 보여라.

▶ *Solution*

- 다중화기는 그림 6.16에서 볼 수 있다. 각 프레임은 각 채널로부터 1바이트 씩 전송한다. 따라서 프레임의 크기는 4바이트 또는 32비트이다. 각 채널이 매초 100바이트 전송하며 프레임은 각 채널로부터 1바이트씩 나르므로 프레임 속도는 매초 100프레임이어야 한다. 그러므로 프레임 기간은 1/100초이다. 링크는 매초 100프레임을 나르며 각 프레임은 32비트이므로 비트 전송률은 100×32 또는 3200bps이다.

그림 6.16 예제 6.8



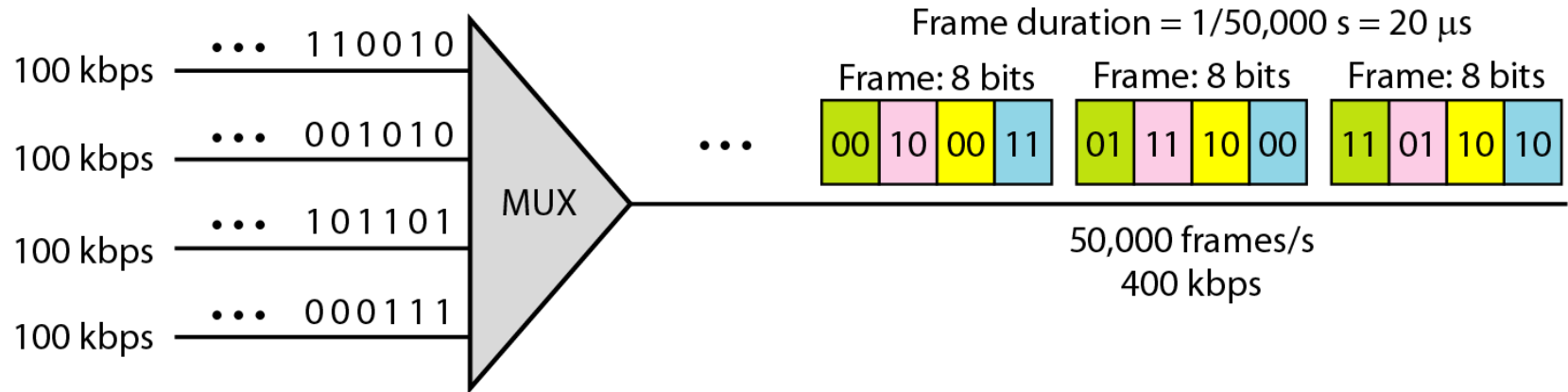
예제 6.9

- ▶ 어떤 다중화기가 시간슬롯마다 2비트씩 실어 4개의 100kbps 채널을 다중화한다. 임의의 4개의 입력에 대해 해당 출력을 보여라. 프레임 속도는 얼마인가? 프레임 기간은 얼마인가? 비트 전송률은? 비트기간은 얼마인가?

▶ *Solution*

- 그림 6.17은 임의의 입력에 대한 출력을 보여준다. 채널당 2비트씩 포함하므로 링크는 매초 50,000 프레임을 실어나른다. 그러므로 프레임 기간은 $1/50,000$ 초 또는 $20\mu\text{s}$ 이다. 프레임 속도는 매초 50,000 프레임이며, 각 프레임은 8비트를 나른다. 비트 전송률은 $50,000 \times 8 = 400,000$ 비트 또는 400kbps이다. 비트 기간은 $1/400,000$ 초 또는 $2.5\mu\text{s}$ 이다.

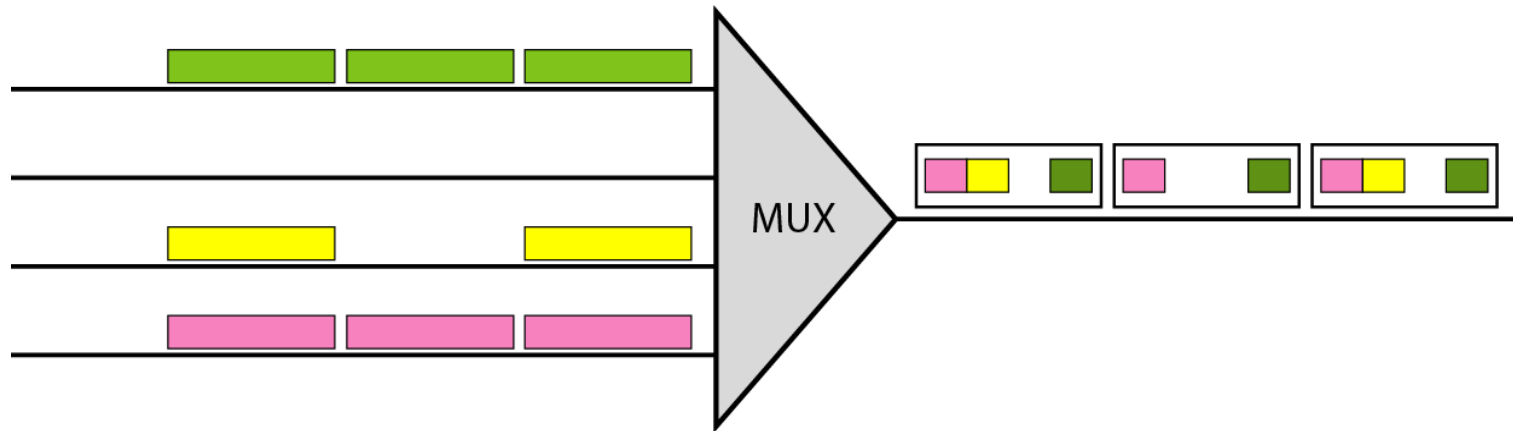
그림 6.17 예제 6.9



시분할 다중화

❖ 빈 슬롯

- 발신자가 전송할 데이터가 없다면 해당 슬롯이 비게 된다.
- 그림 6.18은 입력 회선 중 하나가 전송할 데이터가 전혀 없는 것을 보여준다.



시분할 다중화

❖ 데이터 전송률 관리

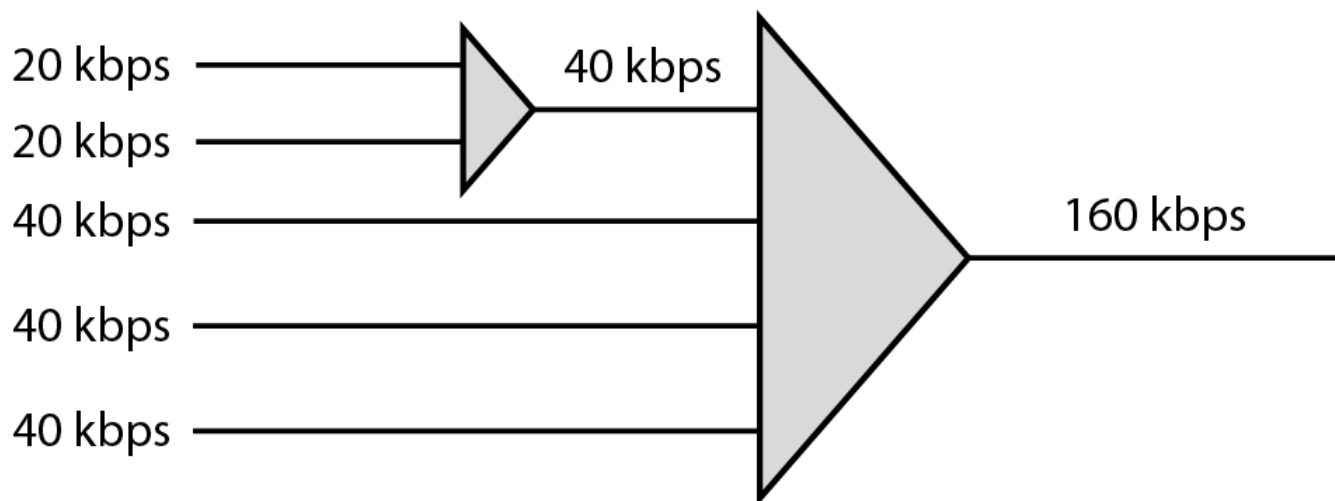
❖ 입력 측 데이터 전송률이 서로 다른 경우 해결 방법

- 1) 다단계 다중화(multilevel multiplexing)
- 2) 복수 틸새 할당(multiple-slot allocation)
- 3) 펄스 채워 넣기(pulse stuffing)

데이터 전송률 관리

❖ 다단계 다중화

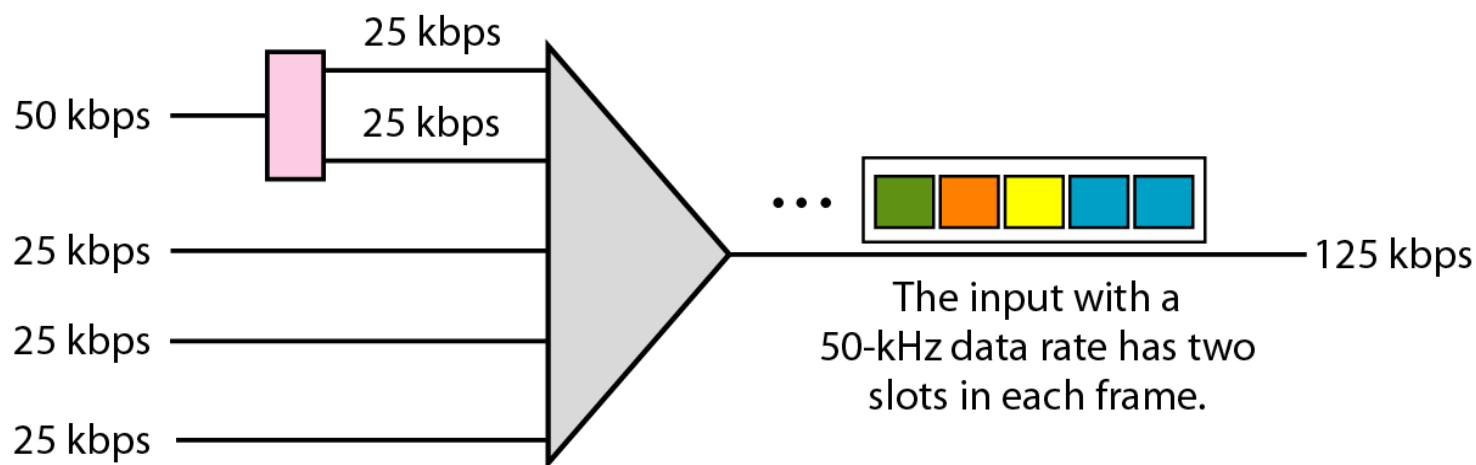
- 어느 입력의 데이터율이 다른 것들에 비해 정수 배 만큼 빠를 때 사용하는 기술
- 그림 6.19에서 다단계 다중화를 보여준다.



데이터 전송률 관리

❖ 복수 슬롯 할당(multiple-slot allocation)

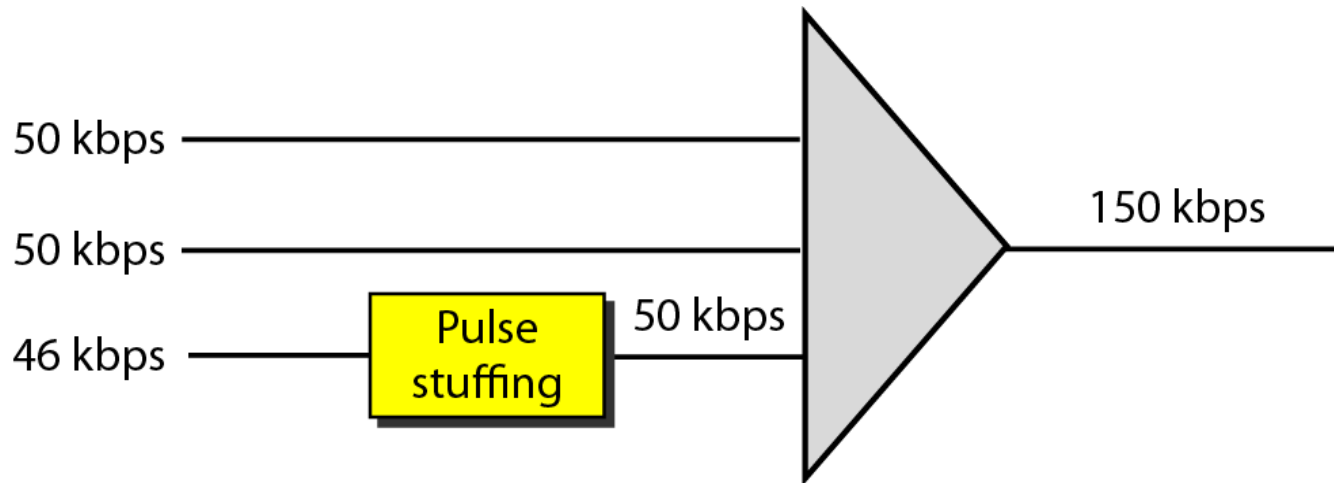
- 입력회선에 한 개 보다 더 많은 틈새를 할당하는 것
- 그림 6.20에는 50kbps의 입력 회선에 2개의 슬롯을 할당한다.



데이터 전송률 관리

❖ 펄스 채우기(pulse stuffing)

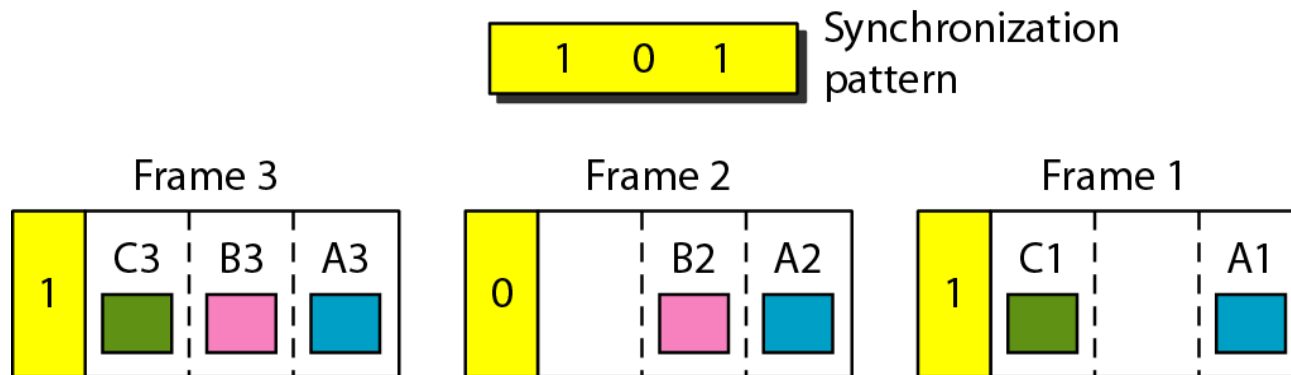
- 가장 높은 데이터 율에 맞추기 위해 공 비트를 끼워 넣는 것
- 비트 패딩(bit padding), 비트 채우기(bit stuffing) 라고도 함
- 그림 6.21은 펄스 채우기를 보여준다.



시분할 다중화

❖ 프레임 동기화(Frame synchronization)

- 프레임 구성비트(framing bits)는 다중화 푸는 장치가 들어오는 채널에 보조를 맞추어 시간슬롯을 정확하게 분리해 낼 수 있도록 해준다.
- 그림 6.22에서 처럼 0과 1을 반복하여 섞어 쓴다.



예제 6.10

- ▶ 각각 매초 250개의 문자를 생산하는 4개의 채널이 있다. 끼워 넣는 단위가 문자이고 1비트의 동기화 비트가 각 프레임에 더해진다면, (1) 각 채널의 데이터 전송률, (2) 각 채널의 각 문자의 기간, (3) 프레임 속도, (4) 각 프레임의 기간, (5) 각 프레임의 비트 수, (6) 링크의 전송률을 구하라.

▶ *Solution*

- 1) 각 채널의 데이터 전송률은 $250 \times 8 = 2000\text{bps} = 2\text{kbps}$.
- 2) 각 채널은 매초 250개의 문자를 보낸다. 그러므로 문자의 기간은 $1/250\text{s}$ 또는 4ms 이다.
- 3) 각 프레임은 각 채널로부터 하나의 문자를 받으므로 링크는 매초 250개의 프레임을 보내야 한다.

- 4) 각 프레임의 기간은 $1/250$ 초 또는 4ms이다. 프레임의 기간은 각 채널로부터 전송되는 문자의 기간과 같다.
- 5) 각 프레임은 4개의 문자 외에 1비트의 동기화 비트를 나른다. 이는 각 프레임은 $4 \times 8 + 1 = 33$ 비트를 나른다.
- 6) 링크는 매초 250개의 프레임을 전송하고 각 프레임은 33비트를 포함한다. 이는 링크의 전송률은 250×33 또는 8250bps이라는 것을 말한다.

예제 6.11

- ▶ 100kbps의 전송률을 갖는 채널과 200kbps의 전송률을 갖는 채널을 다중화해야 한다. 어떻게 가능하겠는가? 프레임 속도는? 프레임 기간은 링크의 전송률은?

▶ *Solution*

- 1개의 시간슬롯을 첫 번째 채널에 할당하고 2개의 슬롯을 두 번째 채널에 할당할 수 있다. 각 프레임은 3비트를 나른다. 첫 번째 채널로부터 1비트를 나르므로 프레임 속도는 매 초 100,000프레임이다. 프레임 기간은 $1/100,000$ 또는 $10\mu\text{s}$ 이다. 비트 전송률은 $100,000 \times 3 = 300\text{kbps}$ 이다.

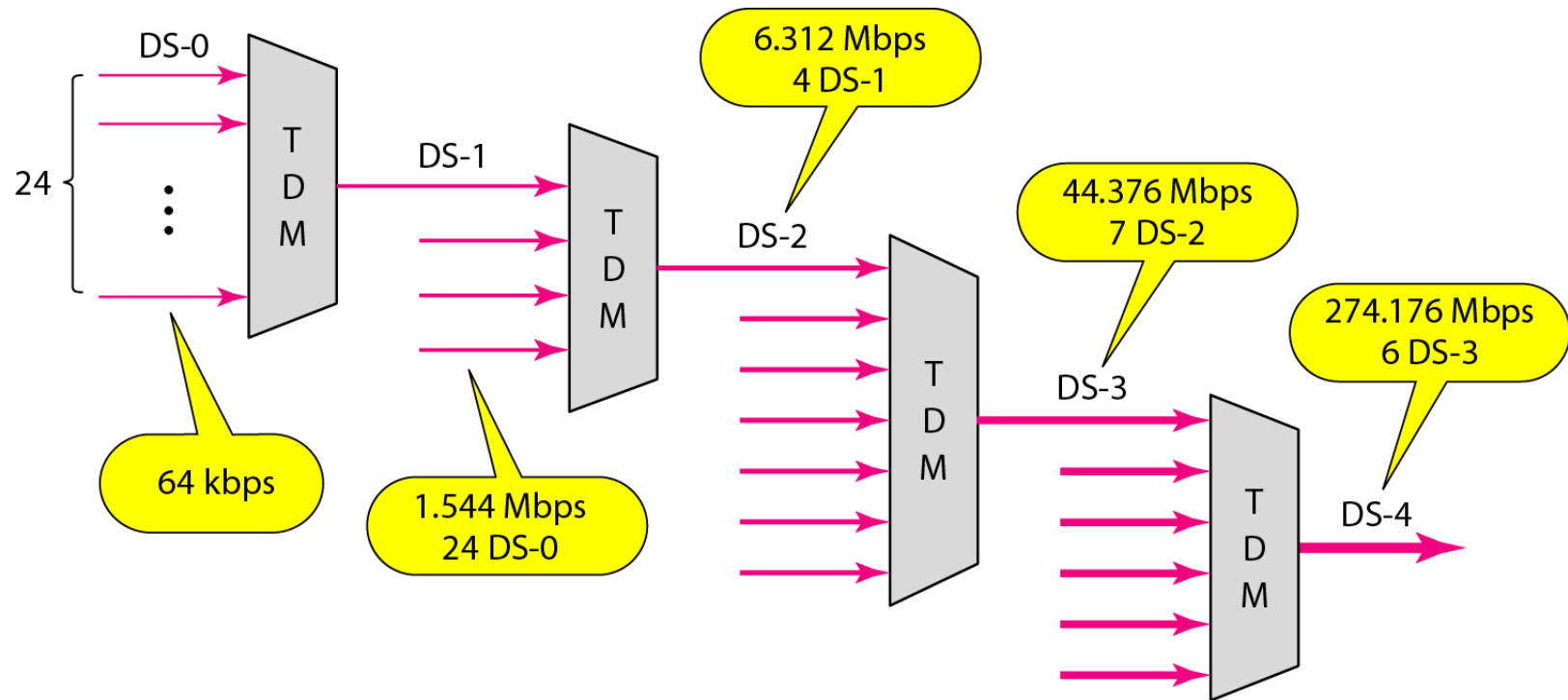
시분할 다중화

❖ 디지털 신호 서비스

- 전화회사들은 디지털 신호 서비스(digital signal service) 또는 디지털 계층 구조(digital hierarchy)를 통해 TDM을 구현한다.
- 그림 6.23에 각 단계에서 지원되는 데이터 속도를 보여준다.
 - DS-0 64kbps
 - DS-1 24개의 DS-0 + 8kbps의 오버헤드 1.544Mbps
 - DS-2 96개의 DS-0 + 168kbps의 오버헤드 6.312Mbps
 - DS-3 672개의 DS-0 + 1.368Mbps의 오버헤드 44.376Mbps
 - DS-4 4032개의 DS-0 + 16.128Mbps의 오버헤드 274.176Mbps

시분할 다중화

그림 6.23 디지털 신호 서비스



시분할 다중화

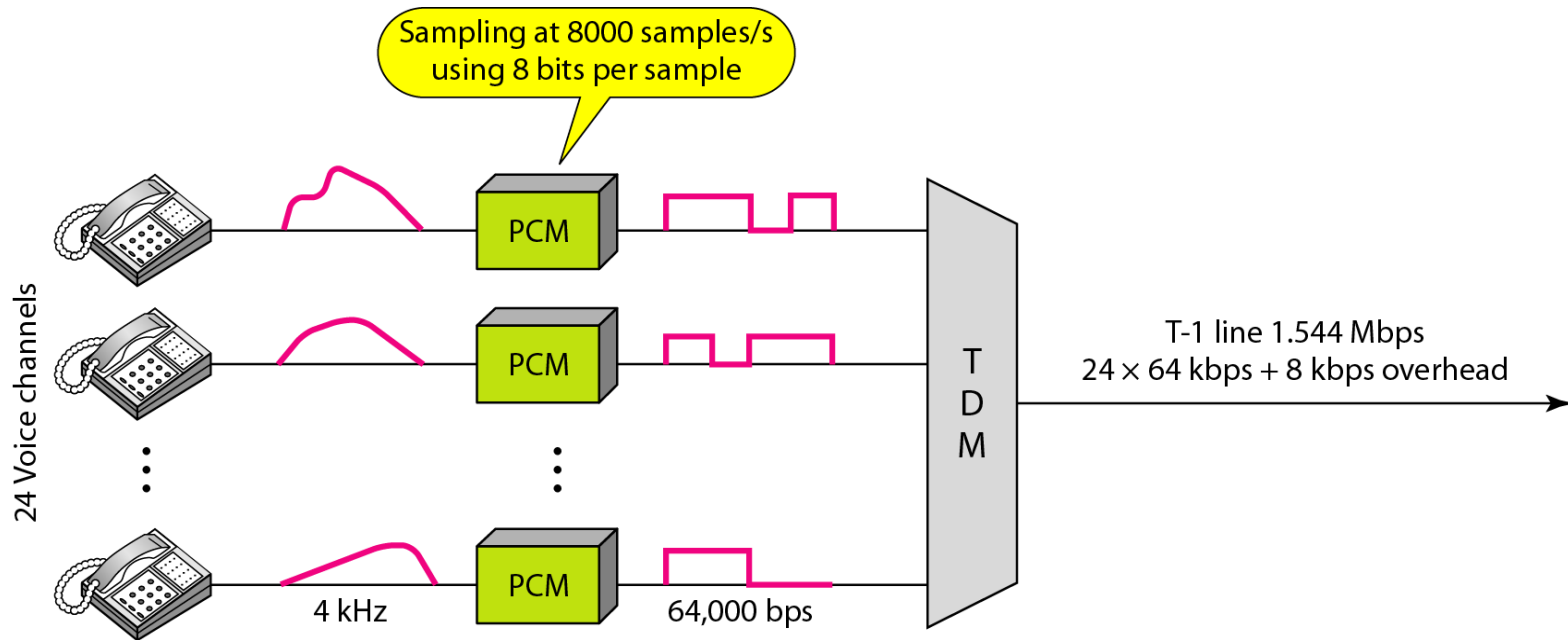
▶ T회선

- DS-0는 서비스 이름이다. 이같은 서비스를 구축하기 위해 전화회사는 T 회선(T line)을 사용한다.
- 표 6.1 DS 및 T회선 전송속도를 보여준다.

<i>Service</i>	<i>Line</i>	<i>Rate (Mbps)</i>	<i>Voice Channels</i>
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

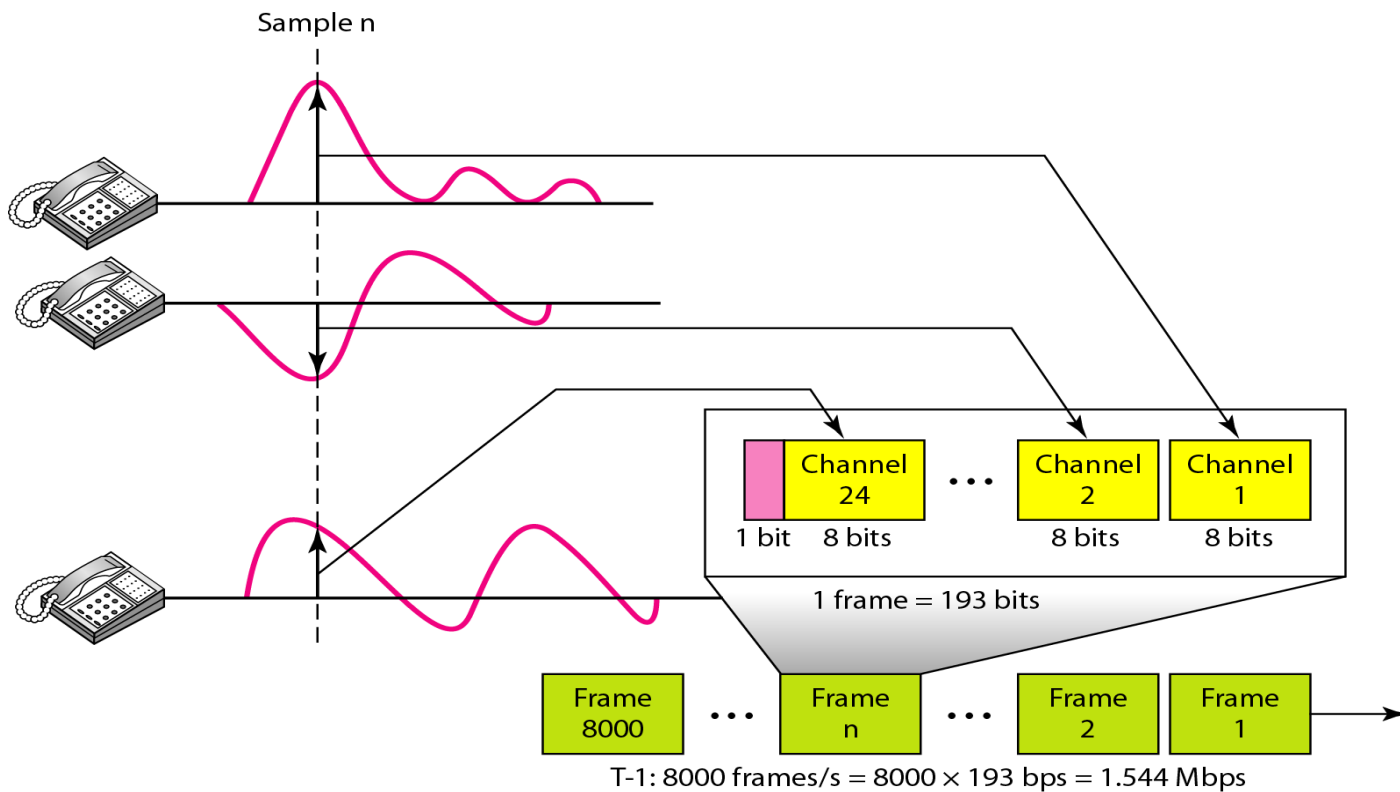
시분할 다중화

▶ 아날로그 전송용 T회선



시분할 다중화

▶ T-1 프레임 구조



시분할 다중화

❖ E 회선

- 유럽은 E 라인으로 불리는 T-라인 버전을 사용

<i>Line</i>	<i>Rate (Mbps)</i>	<i>Voice Channels</i>
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1920

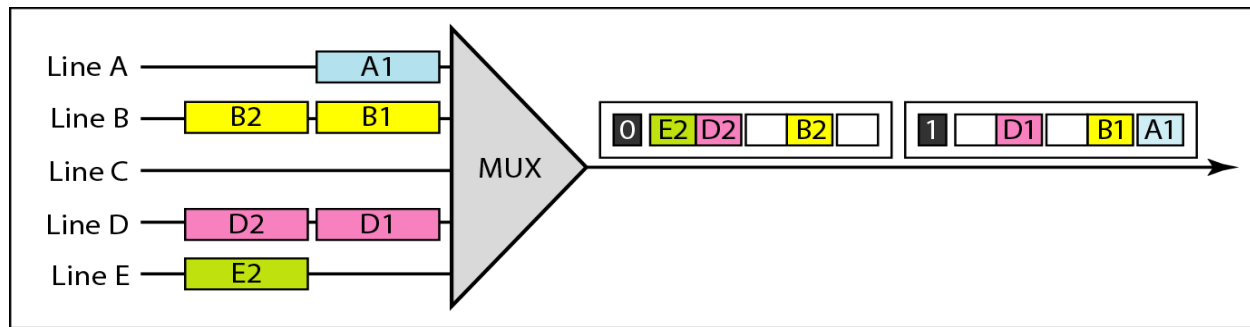
시분할 다중화

❖ 통계적 시분할 다중화

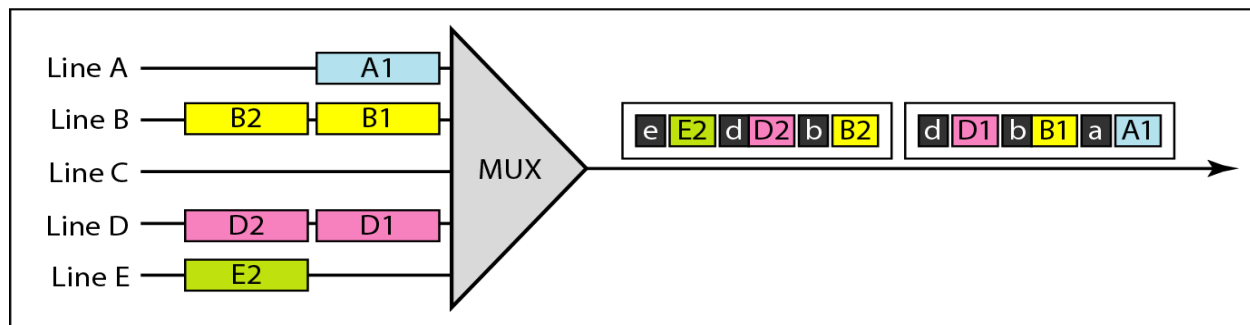
- 대역폭의 효율을 높이기 위해 슬롯은 동적으로 할당된다.
- 그림 6.26은 동기 및 통계적 TDM의 예를 보여준다.
- 주소 지정
 - 목적지 주소를 각 슬롯에 넣어 주어야 한다.
 - 주소 지정하는 방법으로는 N개의 서로 다른 출력 회선을 $n = \log_2 N$ 개의 비트를 주소로 사용한다.
- 슬롯 크기
 - 통계적 TDM에서 통상 데이터의 크기는 여러 바이트이고 주소는 몇 바이트 정도가 된다.
- 동기화 비트의 부재
- 대역폭
 - 통계적 TDM에서는 링크의 용량은 각 채널의 용량의 합보다 작다.

시분할 다중화

그림 6.26 TDM 슬롯 비교



a. Synchronous TDM



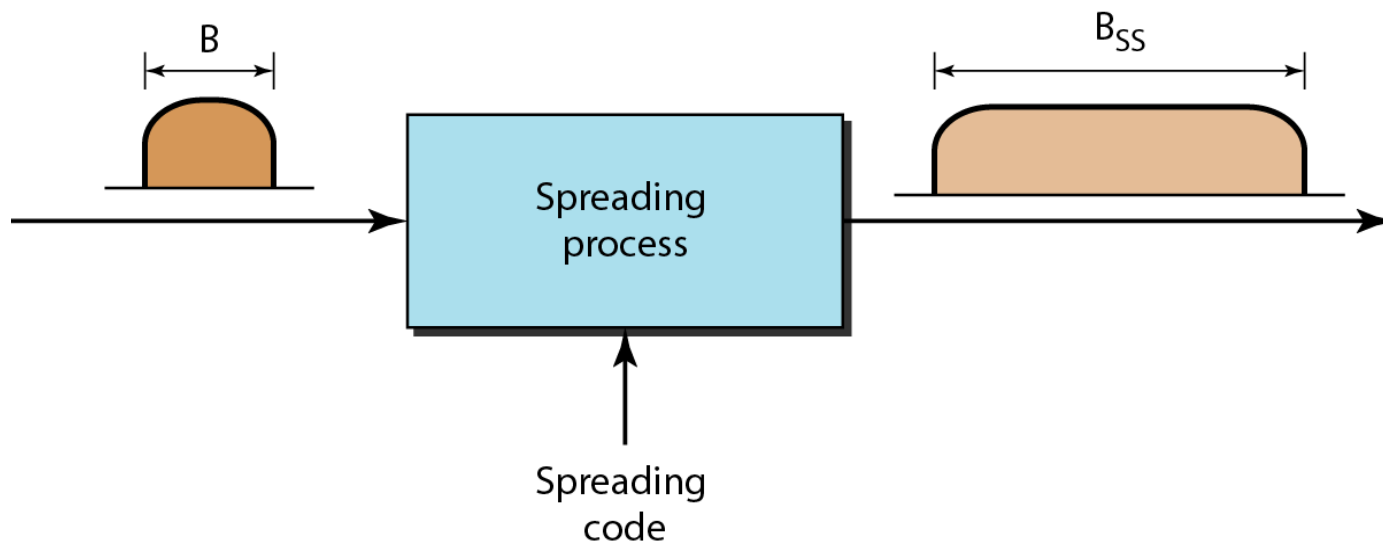
b. Statistical TDM

2. 대역 확산 방식

- ▶ 대역 확산 방식(SS, Spread Spectrum)은 서로 다른 발신지로부터의 신호를 합하여 더 큰 대역으로 만들지만 그 목적은 도청, 전파 교란 방지를 위해 사용한다.
- ▶ 그림 6.27은 대역 확산 방식이다.
 1. 각 지국의 할당된 대역폭은 필요 대역폭보다 더 크다. 추가 정보를 보낼 수 있다.
 2. 확산 과정은 발신지로부터 신호가 생성된 이후에 진행된다.
- 대역폭을 확산하는 기법
 1. 주파수 뛰기 대역 확산(FHSS, Frequency Hopping Spread Spectrum)
 2. 직접 순열 대역 확산(DSSS, Direct Sequence Spread Spectrum)

대역 확산 방식

그림 6.27 대역 확산 방식



주파수 뛰기 대역 확산

- ❖ 주파수 뛰기 대역 확산 방식(FHSS)은 발신지 M 개의 신호를 서로 다른 반송파를 사용하여 변조한다.
- ❖ 그림 6.28은 일반적인 FHSS의 도면이다. 가임의 잡음(PN, pseudorandom noise)이라고 부르는 가임의 코드 생성기(pseudorandom code generator)가 매 뛰기 주기(hopping period) T_h 동안 k 비트의 패턴을 만들어낸다.
- ❖ 뛰기 주기에 사용될 주파수를 주파수 테이블에서 알아내어 주파수 합성기로 보낸다.
- ❖ 주파수 합성기는 해당 주파수의 반송파를 생성하여 발신지 신호를 변조한다. (그림 6.29)

주파수 뛰기 대역 확산

그림 6.28 주파수 뛰기 대역 확산(FHSS)

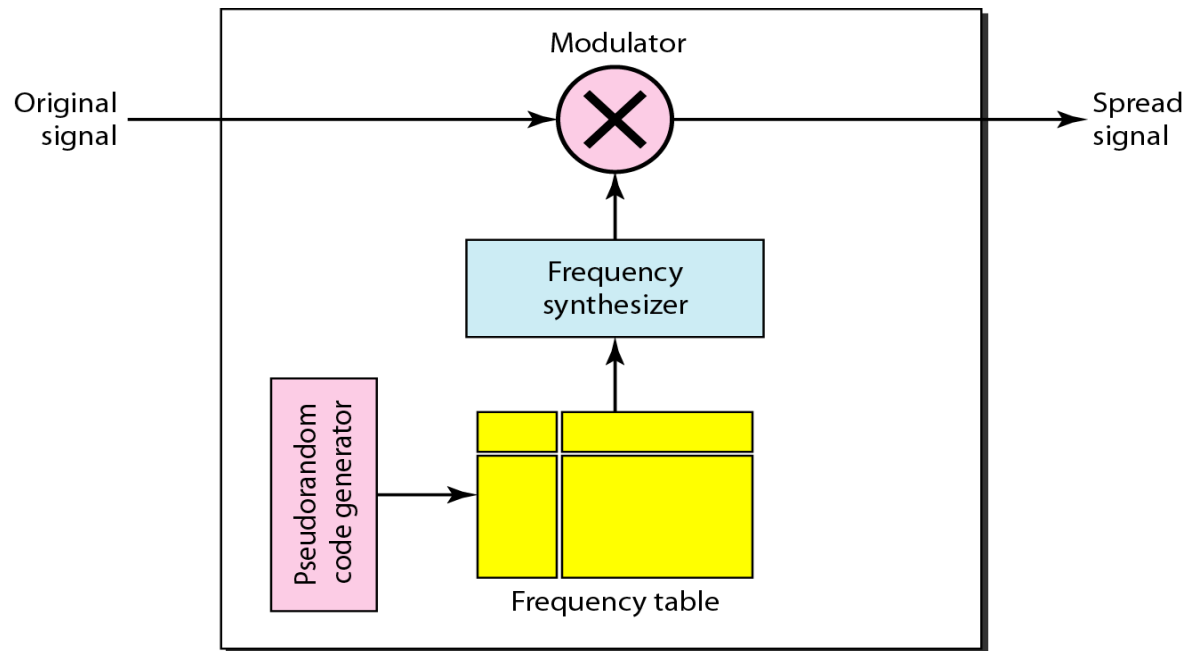
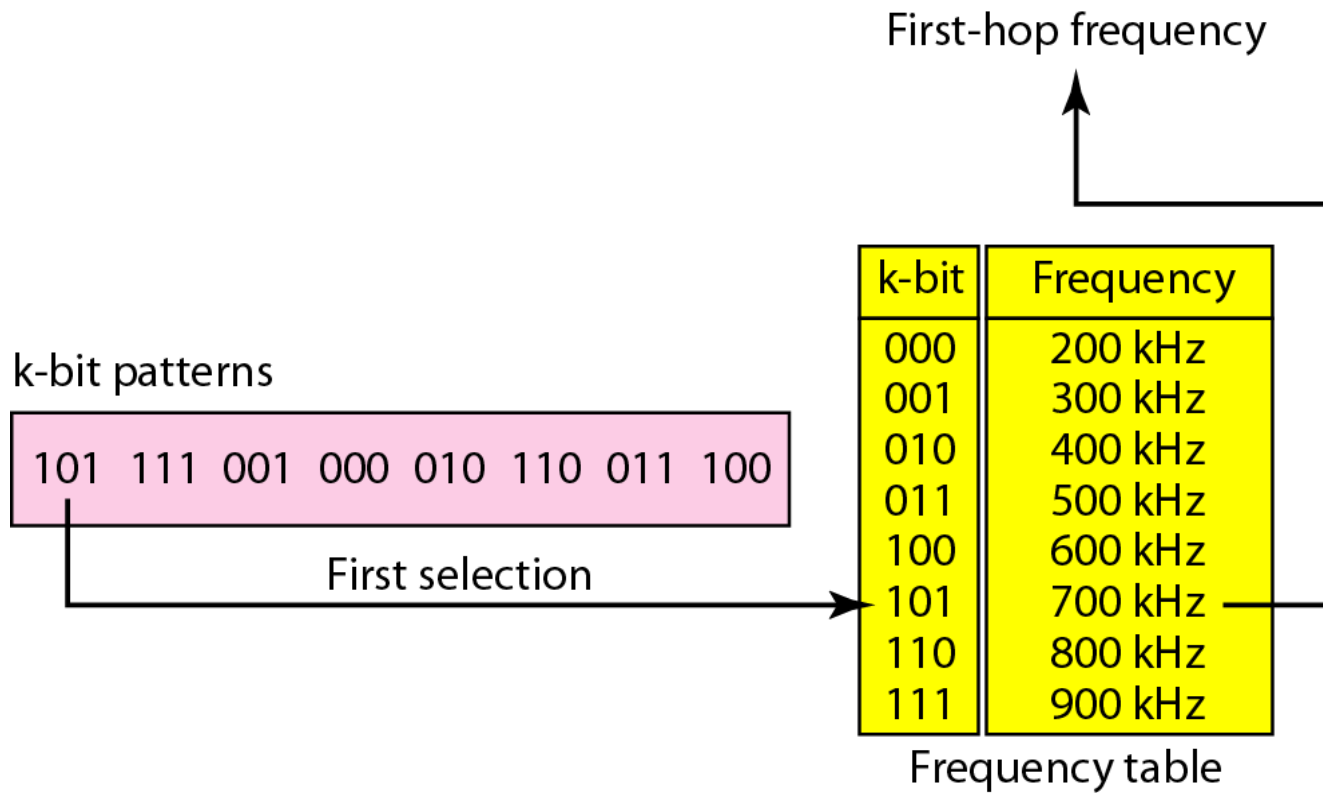


그림 6.29 FHSS에서 주파수 선정



주파수 뛰기 대역 확산

- ❖ 그림 6.30은 FHSS 사이클이 반복되는 것을 보여준다. 여기서 원래 신호를 위한 대역폭은 100kHz라고 가정한다.
- ❖ 많은 k비트 패킷이 있고 뛰기 주기가 짧다면 송수신자 사이에 비밀을 유지할 수 있다. 또한 모든 주기에 전파 교란은 할 수 없다.
- ❖ 대역폭 공유
 - ❖ 그림 6.31 대역폭 공유를 보여준다.

그림 6.30 FHSS 사이클

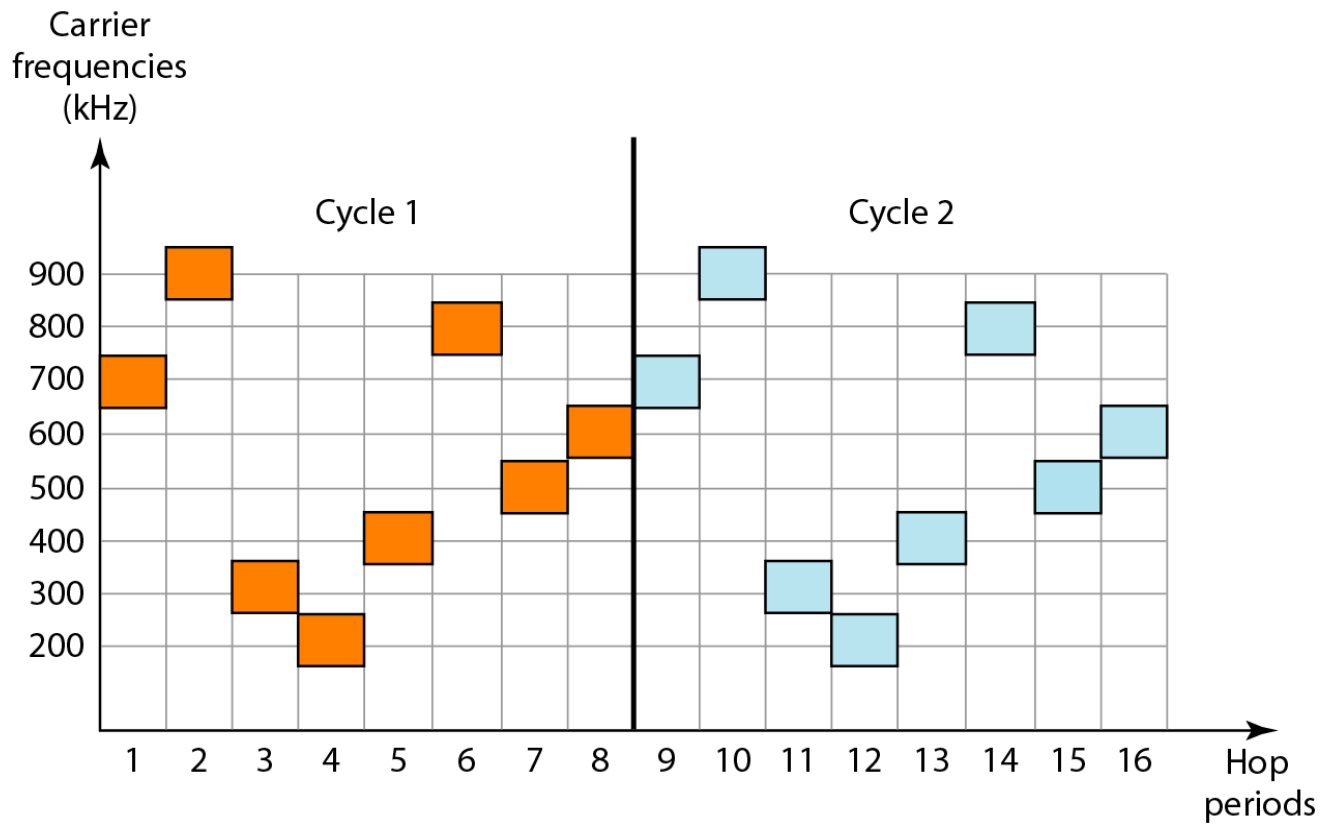
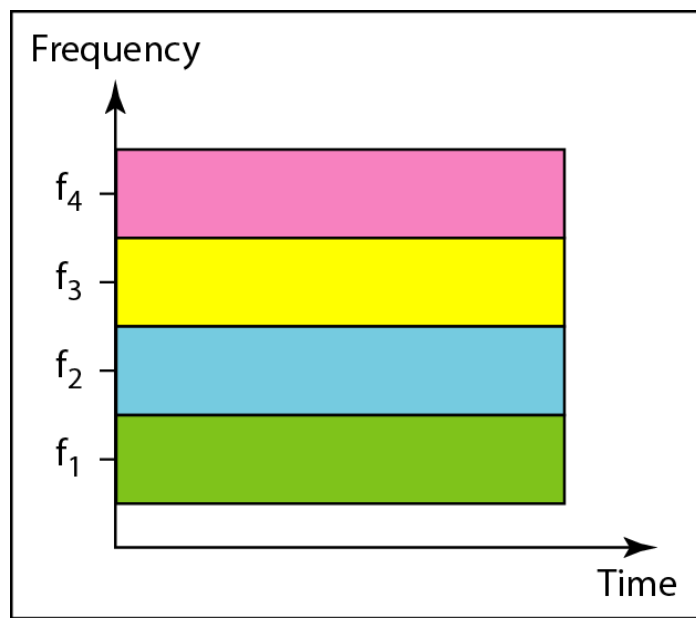
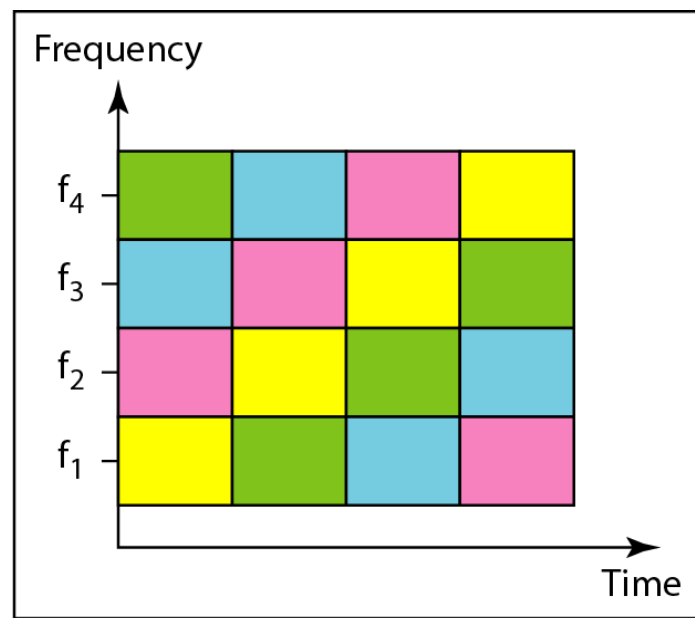


그림 6.31 대역폭 공유



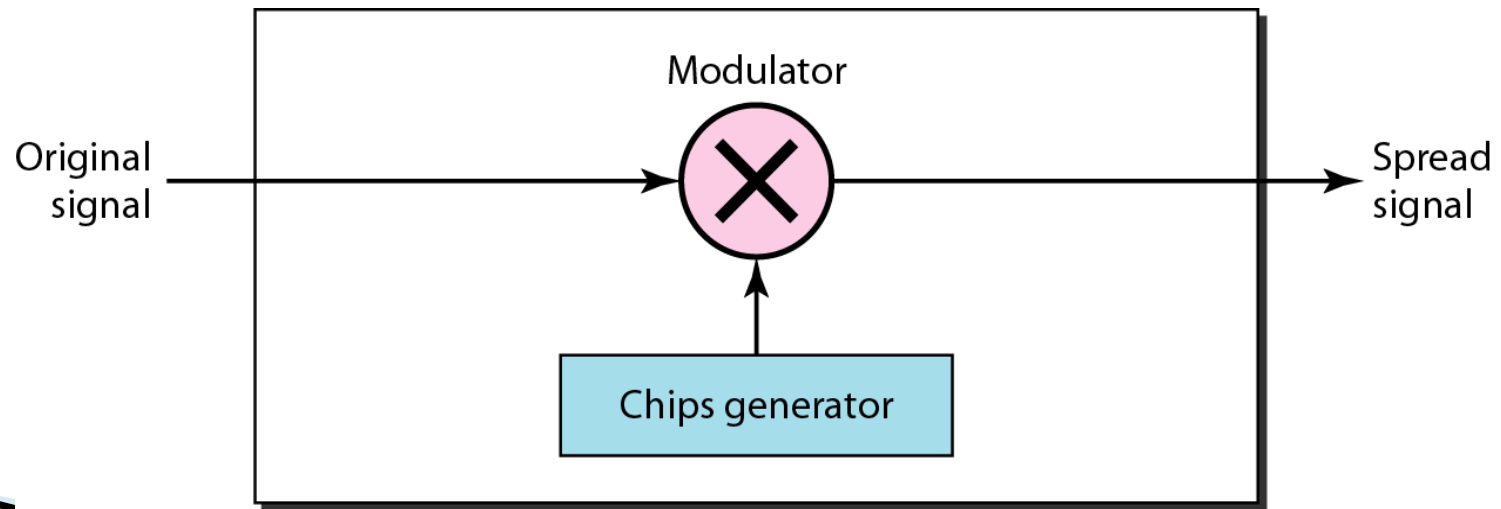
a. FDM



b. FHSS

직접 순열 확산 방식

- ❖ 직접 순열 확산 방식(DSSS, direct sequence spread spectrum)에서는 각 데이터 비트를 확산코드를 사용하여 n 개의 bit로 대체한다.
- ❖ 각 비트에 칩(chip)이라고 불리는 n 비트의 코드를 지정
- ❖ 그림 6.32는 DSSS 개념을 보여준다.



직접 순열 확산 방식

- ▶ 그림 6.33에서는 11개의 칩 코드를 보여준다.

