

5G通訊演算法工程師 培訓課程

Course Overview

2021

Outlines

- * **FPGA/Verilog 數位邏輯電路設計**
 - ◆ Introduction to FPGA
 - ◆ Introduction to Verilog
- * 通道編碼技術(Channel Coding)
- * **OFDM 與多通道通訊系統**

遠古通訊

烽火臺



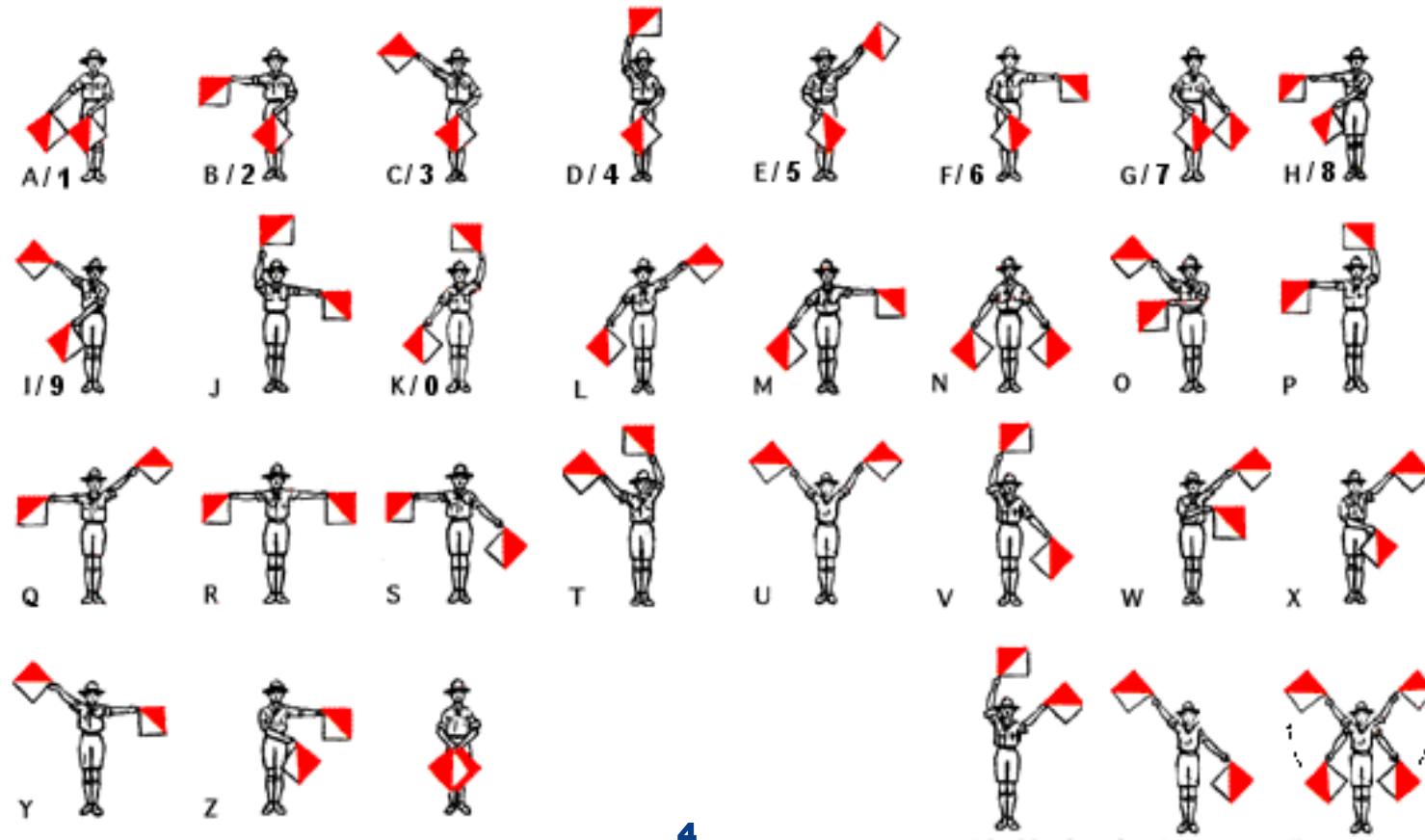
烽火台，又稱烽燧，中國古代的戰略性建築，一般建在險要處或交通要道上。一旦發現敵情，便發出警報：

- 白天以燃燒摻有糞便的柴草，釋放濃煙來表達訊息；
- 夜裡則燃燒加有硫磺和硝石的乾柴，使火光通明，以傳遞緊急軍情；

遠古通訊

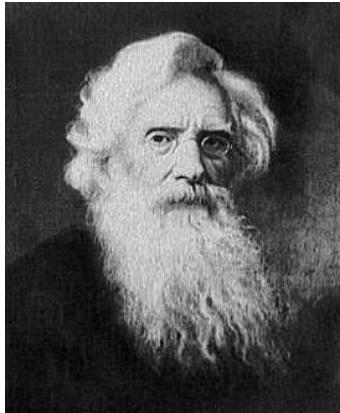
旗語

1684年英國人羅伯特·虎克利用懸掛數種明顯的符號來通訊。1793年法國人克嚕·夏薔利用十字架左右木臂上下移動所呈現出的位置和角度來表示各個字母，叫做「信號標」（原文Semaphore）。

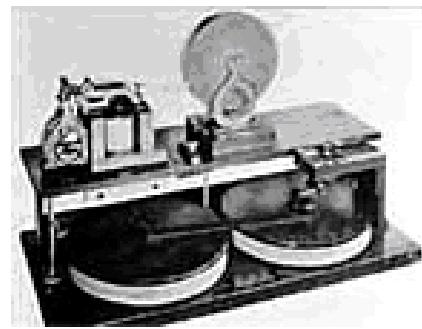


“上帝創造了何等奇蹟！” — 電報的發明

- * 1844年5月24日在美國國會，摩斯操縱著他傾十餘年心血研製成功的電報機，向巴爾的摩發出了人類歷史上的第一份電報：“*What hath God wrought*(上帝創造了何等奇蹟)！”
- * 電報的發明，拉開了電信時代的序幕



摩斯
(美國人，1791~1872)



摩斯發出第一封
電報的設備

5

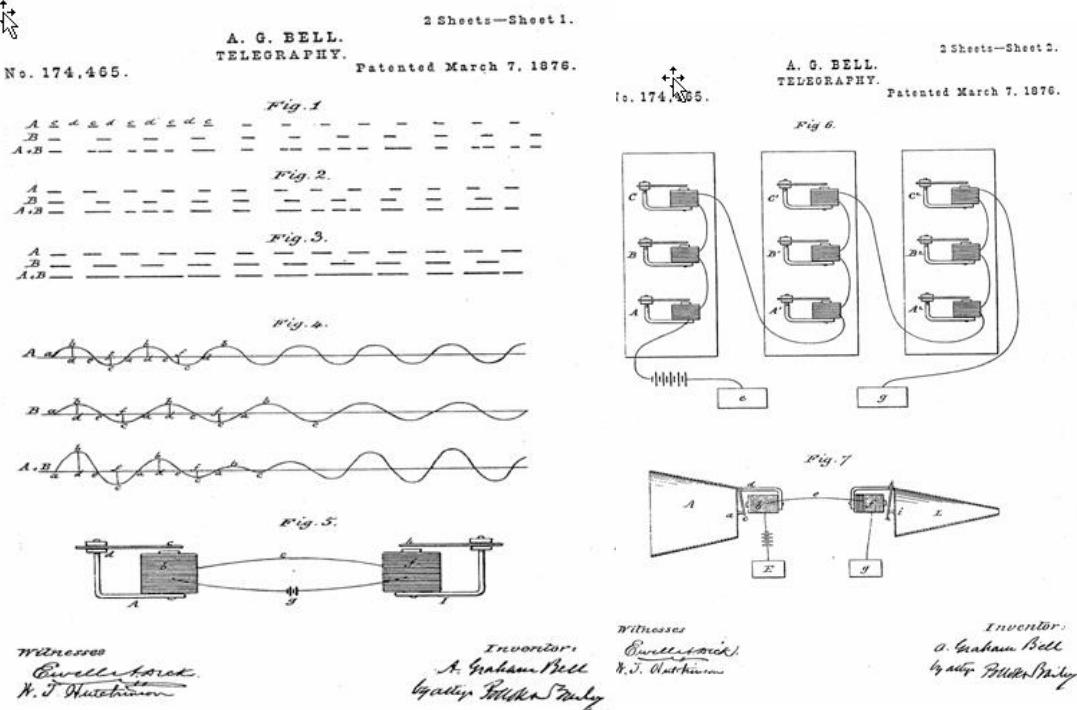
International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.

A	—	U	••—
B	—••	V	•—•—
C	—•—	W	•—•—•
D	—••	X	—•—
E	•	Y	•—•—•
F	••—	Z	—•—•—
G	——		
H	•••		
I	••		
J	———		
K	—•—	1	———
L	•—•—	2	—•——
M	——	3	—•—
N	—••	4	——•
O	———	5	—•——
P	—•——	6	—•———
Q	—•———	7	—•————
R	—•————	8	—•—————
S	—•—————	9	—•——————
T	—	0	—•———————

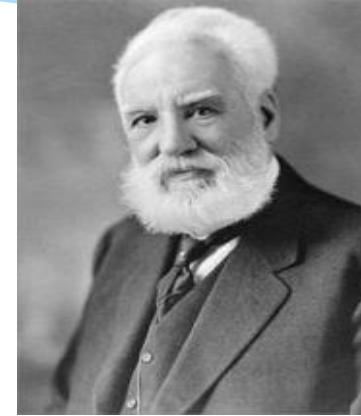
“沃森特先生，快來幫我啊！”——電話的發明

- * 1875年6月2日，貝爾和他的助手沃森特正在進行電話模型最後測試，貝爾在操作時不小心把硫酸濺到自己的腿上並叫了起來：“沃森特先生，快來幫我啊！”沒有想到，這句話通過他實驗中的電話傳到了在另一個房間工作的沃森特先生的耳朵裏，成為人類歷史上的一刻



貝爾的電話專利(No. 174,465)

6



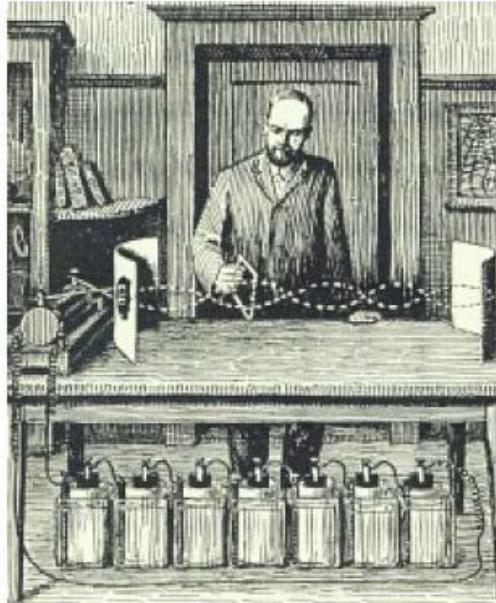
貝爾(英裔美籍，1847~1922)



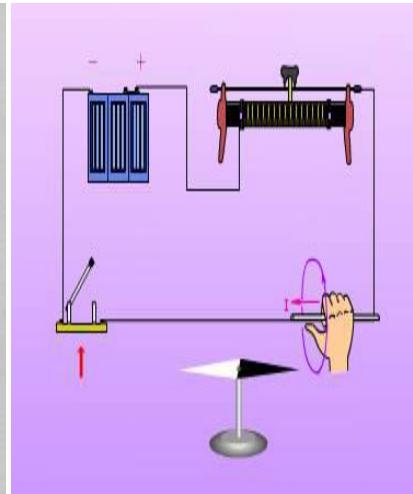
1892年貝爾開啟全球
首用商業電話服務

無形的信使—電磁波的發現

- * 1820年代法拉第以實驗發現了導線在磁場中運動時會有電流產生的現象，此即所謂的“電磁感應”現象
- * 1864年，馬克斯威爾發表了電磁場理論，成為人類歷史上預言電磁波存在的第一人
- * 1887年赫茲以實驗證實電磁波的存在



赫茲(德國人，1857~1894)與赫茲實驗



法拉第(英國人，1791~1867)與電磁感應



$$\oint_S \mathbf{D} \cdot d\mathbf{A} = \int_V \rho dV$$
$$\oint_S \mathbf{B} \cdot d\mathbf{A} = 0$$
$$\oint_C \mathbf{E} \cdot d\mathbf{l} = - \frac{d}{dt} \int_S \mathbf{B} \cdot d\mathbf{A}$$
$$\oint_C \mathbf{H} \cdot d\mathbf{l} = \int_S \mathbf{J} \cdot d\mathbf{A} + \frac{d}{dt} \int_S \mathbf{D} \cdot d\mathbf{A}$$

James Clark Maxwell

馬克斯威爾(英國人，1831~1879)與電磁方程式

飛越整個世界的電磁波—無線電報的發明

- * 1896年，波波夫成功地用無線電進行莫爾斯電碼的傳送，距離為250米，電文內容為“海因裏斯•赫茲”
- * 在1897年5月18日，另一位研究無線電的年輕人—馬可尼，改進了無線電傳送和接收設備，在布里斯托爾海峽進行無線電通信取得成功，把資訊傳播了12公里
- * 1909年馬可尼獲得諾貝爾物理獎，被尊稱為無線通信之父



波波夫(俄國人，1859~1905)與俄國發行之紀念郵票



馬可尼(義大利人，1874~1937)



載著聲音飛翔的電波—無線廣播的發明

- * 1902年特波菲德在穆雷廣場放好話筒，由他的兒子在話筒前說話、吹奏口琴，他在附近的樹林裏放置了5台礦石收音機，均能清晰地聽到說話和口琴聲
- * 1906年費森登主持人類歷史上第一次無線電廣播，播放亨德爾的《舒緩曲》，讓聽慣嘀滴答答爾斯電碼聲的電報員首次聽到無線音樂廣播
- * 費森登被尊稱為無線廣播之父

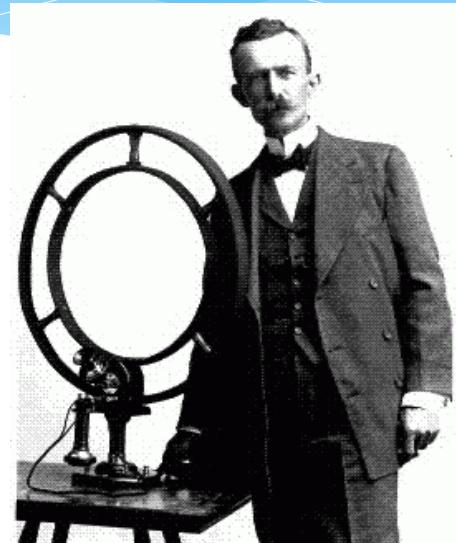


1920年代的收音機



1930年代的收音機

9



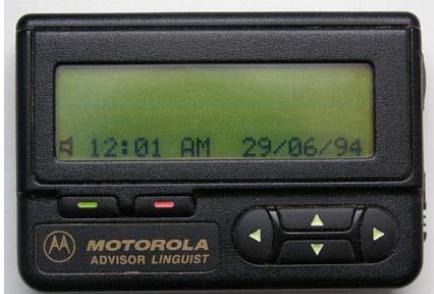
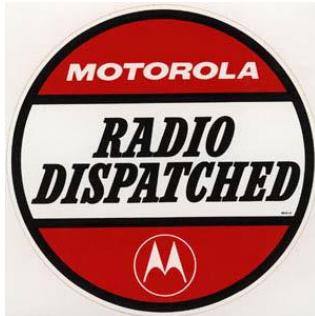
史特波菲德(美國人，1860~1928)



費森登(美國人，1866~1932)

個人通信的發源地—傳呼的誕生

- * 1941年，摩托羅拉生產出了美軍參戰時唯一的可攜式無線電通訊工具SCR-300型高頻率調頻背負式通話機
- * 1956年，全球第一個無線電傳呼機也在摩托羅拉問世了，大部分多應用在無線電派遣



摩托羅拉無線電傳呼機



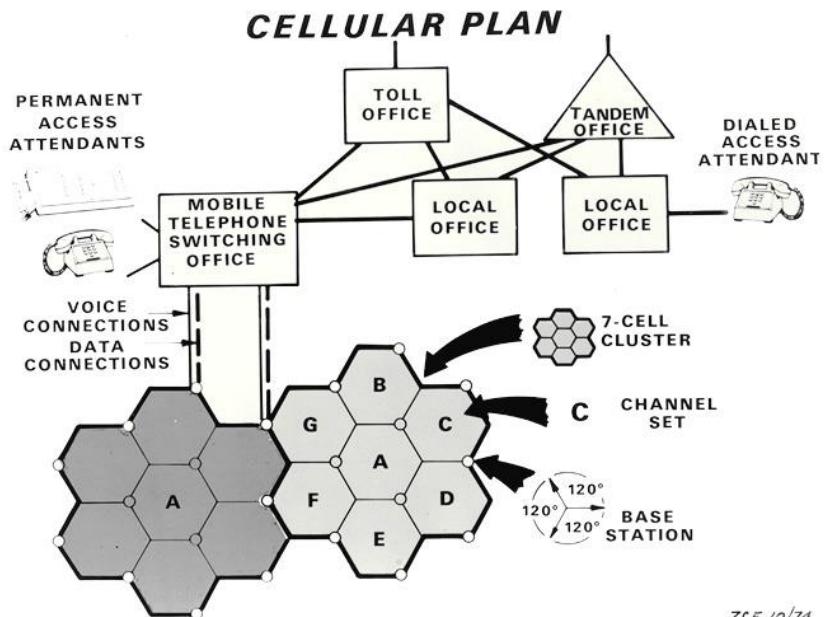
SCR-300 FM 背負式通話機



現代各式樣對講機

實現個人電話的夢想—蜂窩式行動電話的誕生

- * 1970初，貝爾實驗室提出蜂窩系統的覆蓋社區的概念和相關的理論，有效地避免了頻率衝突，又可讓同一頻率多次使用，節省了頻率資源
- * 1979年，AMPS類比蜂窩式移動電話系統在美國芝加哥試驗後，終於在1983年12月在美國投入商用



全球第一支商用手機製造者Dr. Cooper



11 全球第一支商用手機
Motorola DynaTAC8000X



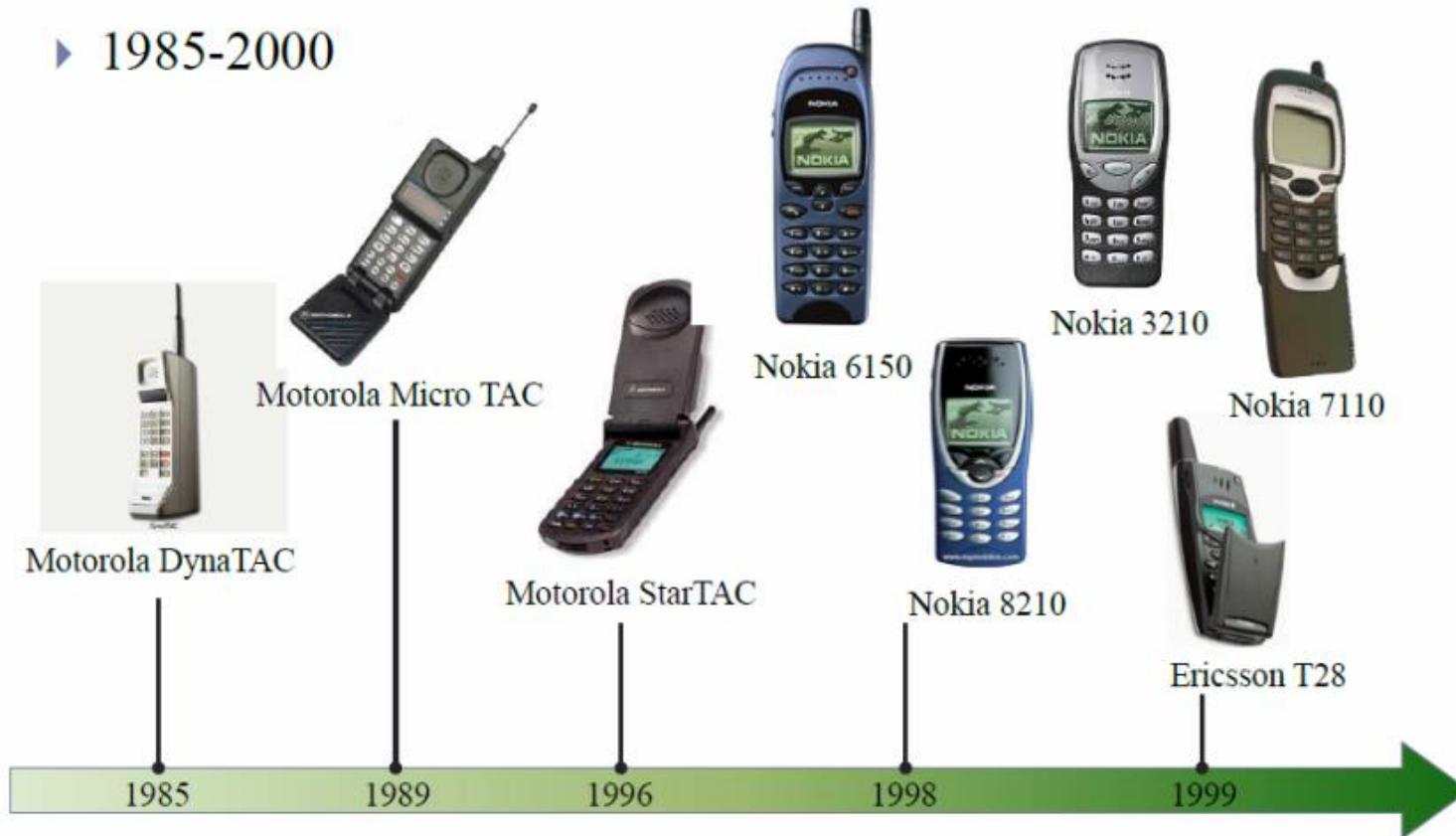
AMPS 黑金剛手機

讓手機走近每一個人—GSM手機的出現

- * 2G手機以數位訊號傳送語音-以數位訊號傳送語音(歐規的GSM、美規的CDMA、日規的PDC)

手機的發展史

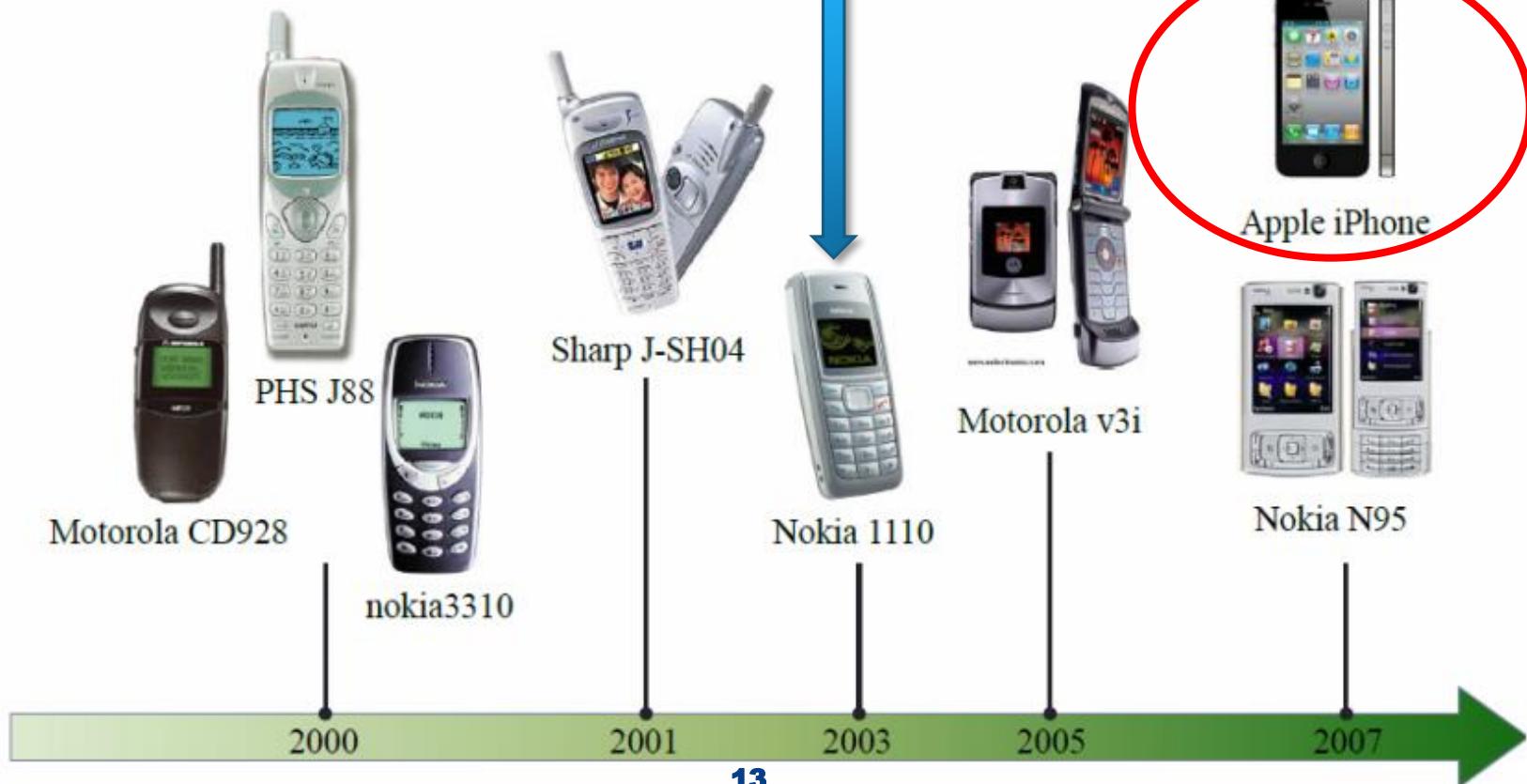
► 1985-2000



讓手機走近每一個人—GSM手機的出現

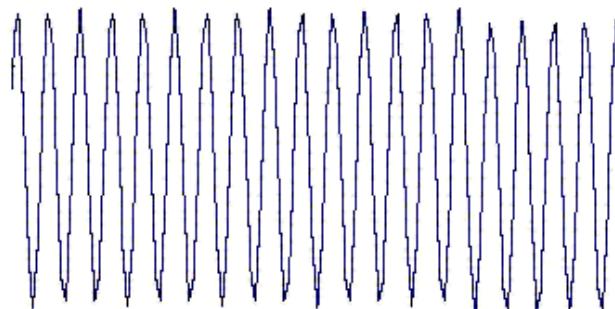
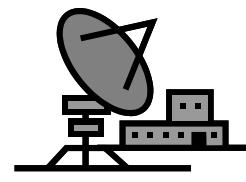
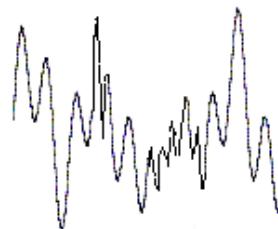
- * 全世界的第一支GSM手機，又是哪支呢？就是當時的手機巨頭-Nokia所推出的1011，當初1011這個型號，也正是為了紀念這款全世界第一支GSM手機的上市日期1992年11月10日而命名

► 2000-2011



Modulation

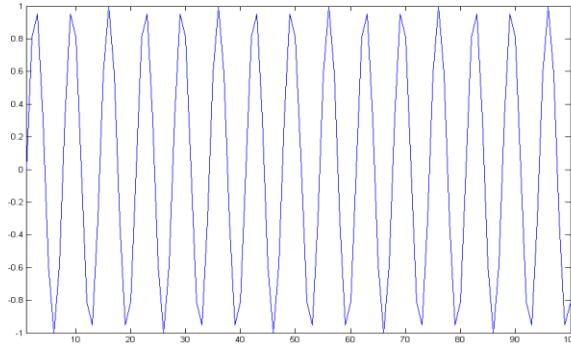
- * 語音轉變成的電氣信號並無法在空氣中做遠距離的傳輸，所以必須要將其轉換成高頻無線電波信號以傳送至遠處，這樣的動作稱為調變 (Modulation)。



調變 (modulation) (1/3)

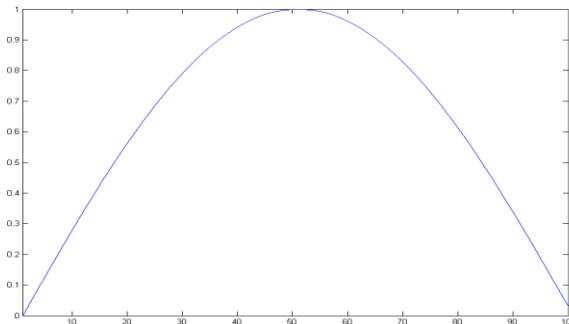
- * 在行動通訊系統中，若希望將語音等資料轉成電磁波在空氣中傳送，不管是轉成類比信號或轉成數位信號傳遞，都還需要更多的處理。
- * 將欲傳送的原始信號編碼放入一個載送的高頻電磁波中，而這個載送電磁波是適合在大氣自然環境中傳送的。
 - ◆ 這個高頻無線電波，稱為載波（carrier）。
 - ◆ 原始信號被稱為進行調變的信號（modulating signal）或基頻的信號。
 - ◆ 轉換成的高頻信號被稱為已調變的信號（modulated signal），適合無線電波傳送。

調變過程

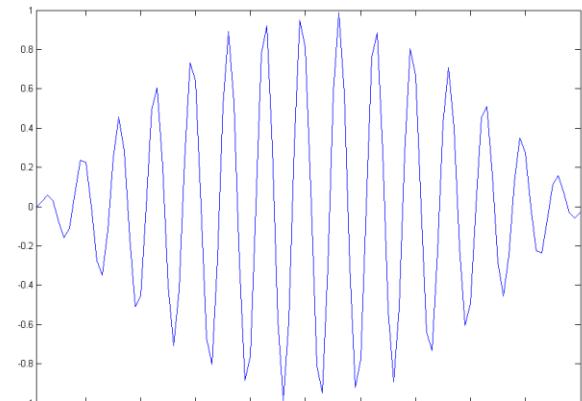


載波的信號

兩個信號相乘



原始的信號



最終傳送的信號

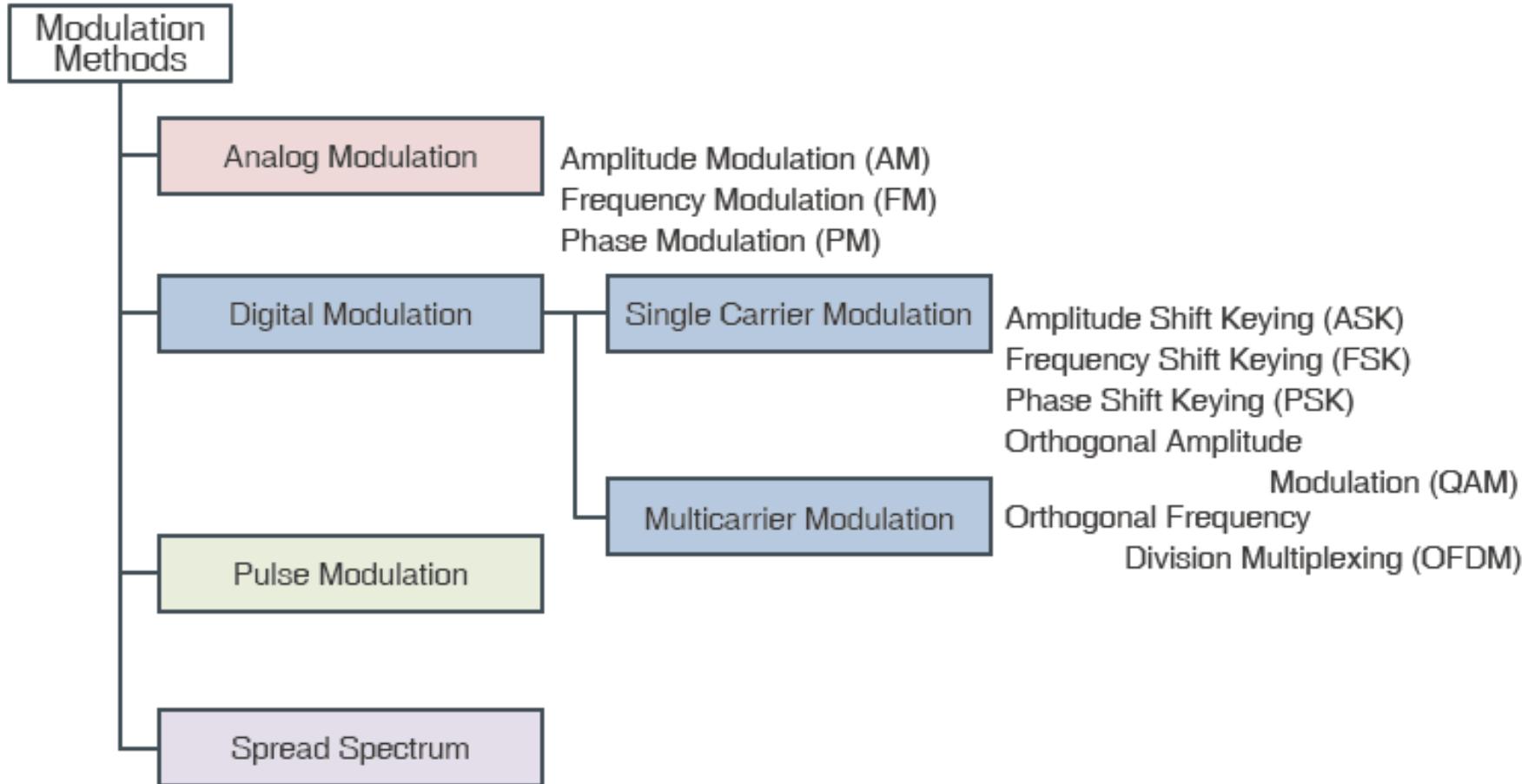
調變 (modulation) (2/3)

- * 已調變的信號從發射機發送出來後，透過無線電介面傳送到接收機。接收機接到信號後，執行調變相反的動作，把它還原成原始信號，稱為解調變 (demodulation) 。
- * 調變技術是將原始信號型態轉換成適合於傳輸介質的傳輸信號型態，以提高傳輸效能之技術。

調變 (modulation) (3/3)

- * 如果原始信號是類比信號，就需要類比調變 (analog modulation & demodulation) 與解調變的技術。
- * 如果原始信號是數位信號，一樣需要將數位信號轉成適合的特定通道傳送，這一類被稱為數位調變與解調變 (digital modulation & demodulation) 。

調變技術

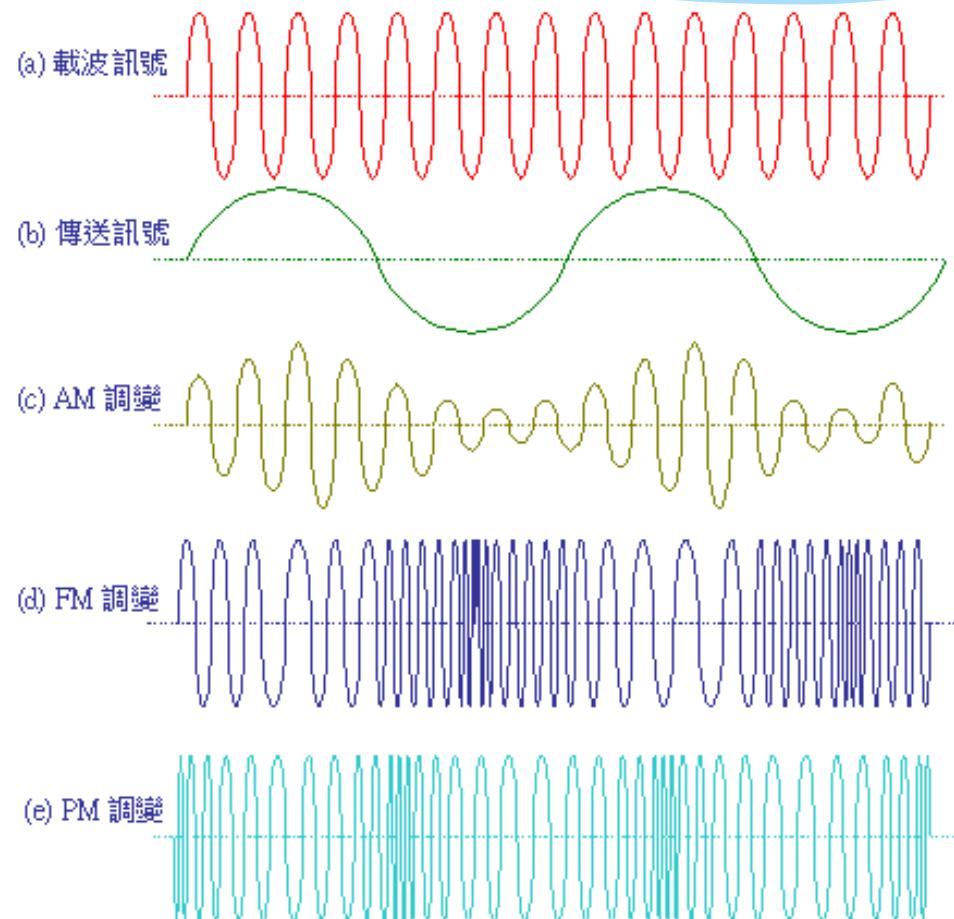


類比調變 (Analog Modulation)

- * 若要以無線電傳送類比信號，將類比信號載入無線電波傳送有多種方式，其原理都是嘗試改變載波某些特徵來表現原始的信號。
- * 載波有三個重要參數可做為調變的因素，即利用載波的振幅、頻率、或相位的變化來表示原始信號。
 - ◆ 振幅調變 (Amplitude Modulation, AM)
 - ◆ 頻率調變 (Frequency Modulation, FM)
 - ◆ 相位調變 (Phase Modulation, PM)

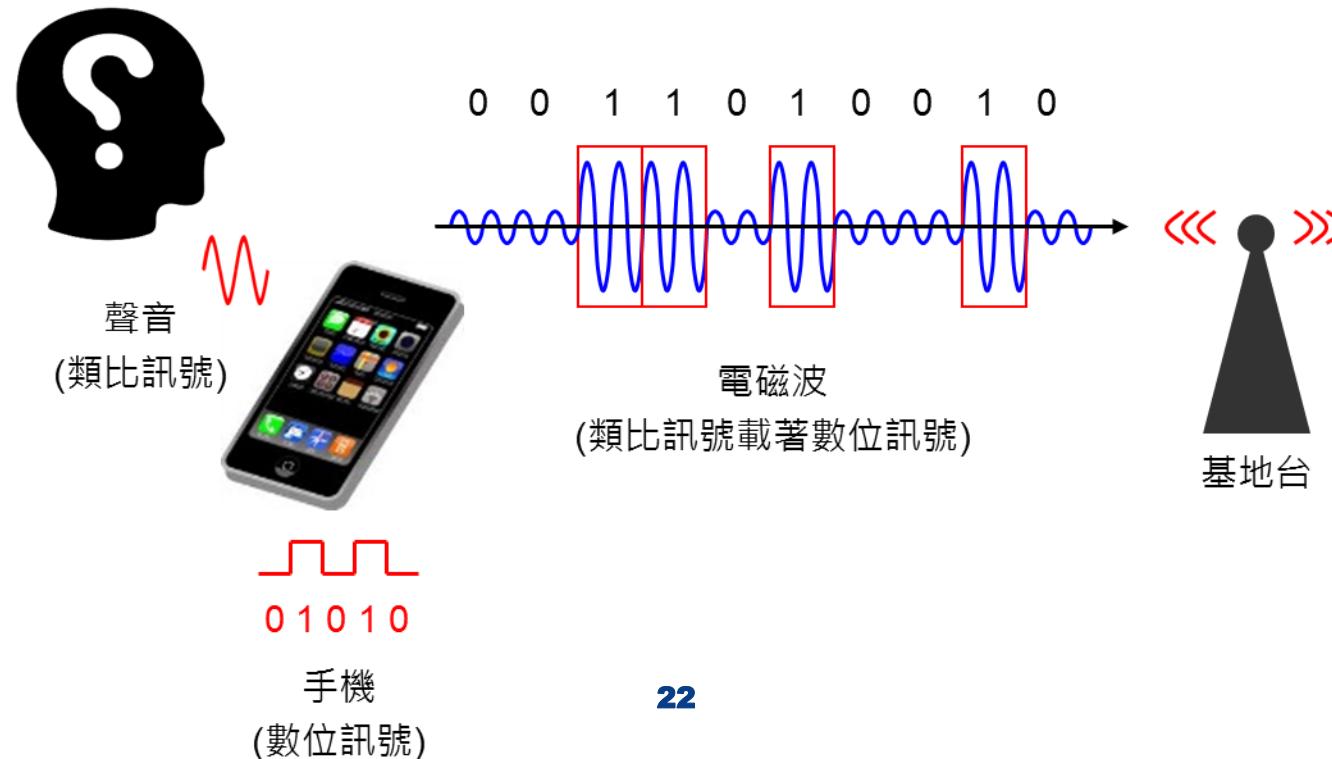
類比調變 (Analog Modulation)

- ◆ 振幅調變 (Amplitude Modulation, AM)
- ◆ 頻率調變 (Frequency Modulation, FM)
- ◆ 相位調變 (Phase Modulation, PM)



數位調變 (Digital Modulation)

- * 現在的手機是屬於「數位通訊」，也就是我們講話的聲音（連續的類比訊號），先由手機轉換成不連續的0與1兩種數位訊號，再經由數位調變轉換成電磁波（類比訊號載著數位訊號），最後從天線傳送出去



數位調變 (1/2)

- * 直接將 0 或 1 的數位信號以電脈衝 (pulse) 利用實體線路傳送，這是屬於基頻調變的範疇。
- * 就行動通訊系統而言，為了無線電遠距離的傳送，就需要將數位信號轉換為類比高頻載波來傳送。
- * 在數位調變中亦可以使用載波振幅、頻率、或相位的三種變化方式將 0 與 1 表現出來。

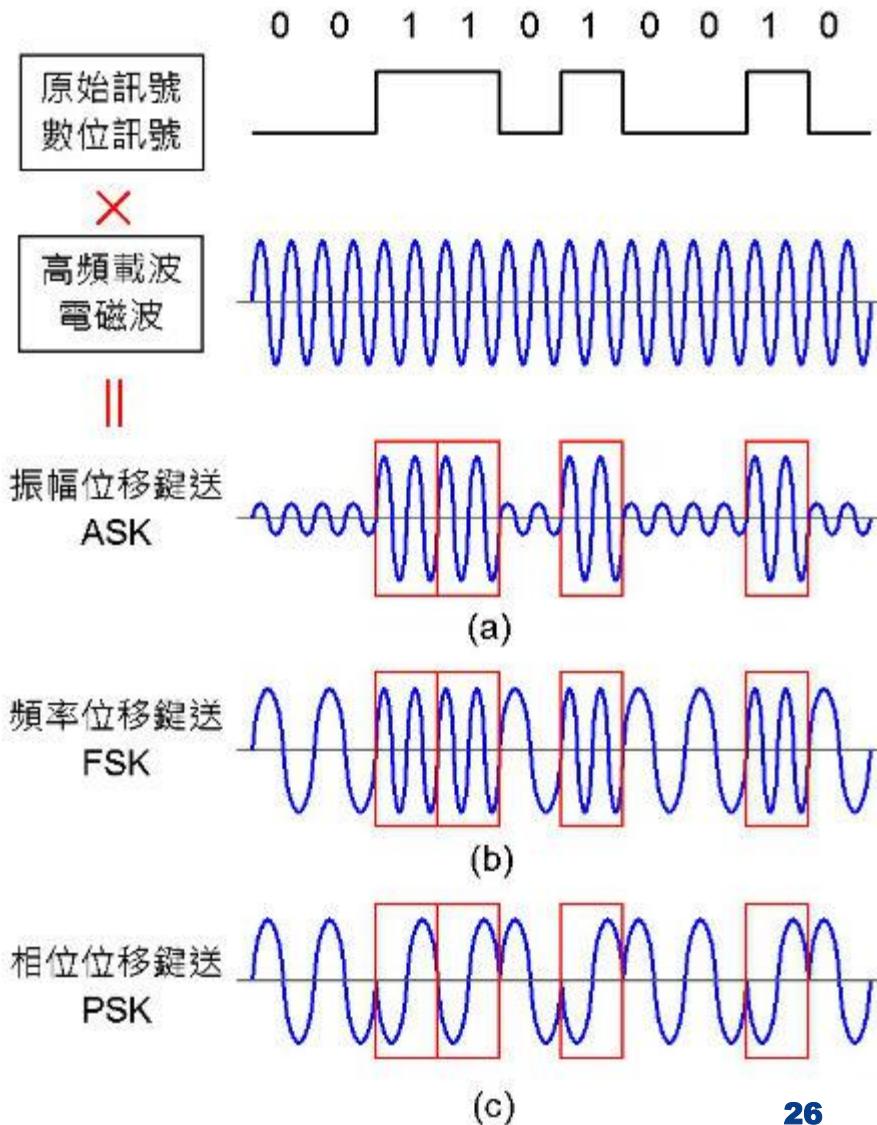
數位調變 (2/2)

- * 可以進行錯誤的偵測及錯誤更正
- * 可以進行加解密的工作，確保通訊安全
- * 希望能夠以簡單不貴的調變技術，提供低位元錯誤率 (bit error rate)，避免行動通訊無線電傳送的不友善環境（如多重路徑和信號衰減）的影響。
- * 至於在效能運用上，數位調變技術可朝能量有效利用與頻寬有效利用兩個方向。

數位調變技術（Digital Modulation）

- * 電磁波是連續的能量，如何利用電磁波替我們傳送這些0與1的數位訊號呢？There are **three** major classes of **digital modulation** techniques used for transmission of digitally represented data：
 - ◆ 振幅位移鍵送（ASK）：利用電磁波的「振幅大小」載著數位訊號（0與1）傳送出去，振幅小代表0，振幅大代表1。
 - ◆ 頻率位移鍵送（FSK）：利用電磁波的「頻率高低」載著數位訊號（0與1）傳送出去，頻率低代表0，頻率高代表1。
 - ◆ 相位位移鍵送（PSK）：利用電磁波的「相位不同（波形不同）」載著數位訊號（0與1）傳送出去，相位 0° 代表0，相位 180° 代表1。

數位調變技術 (Digital modulation)



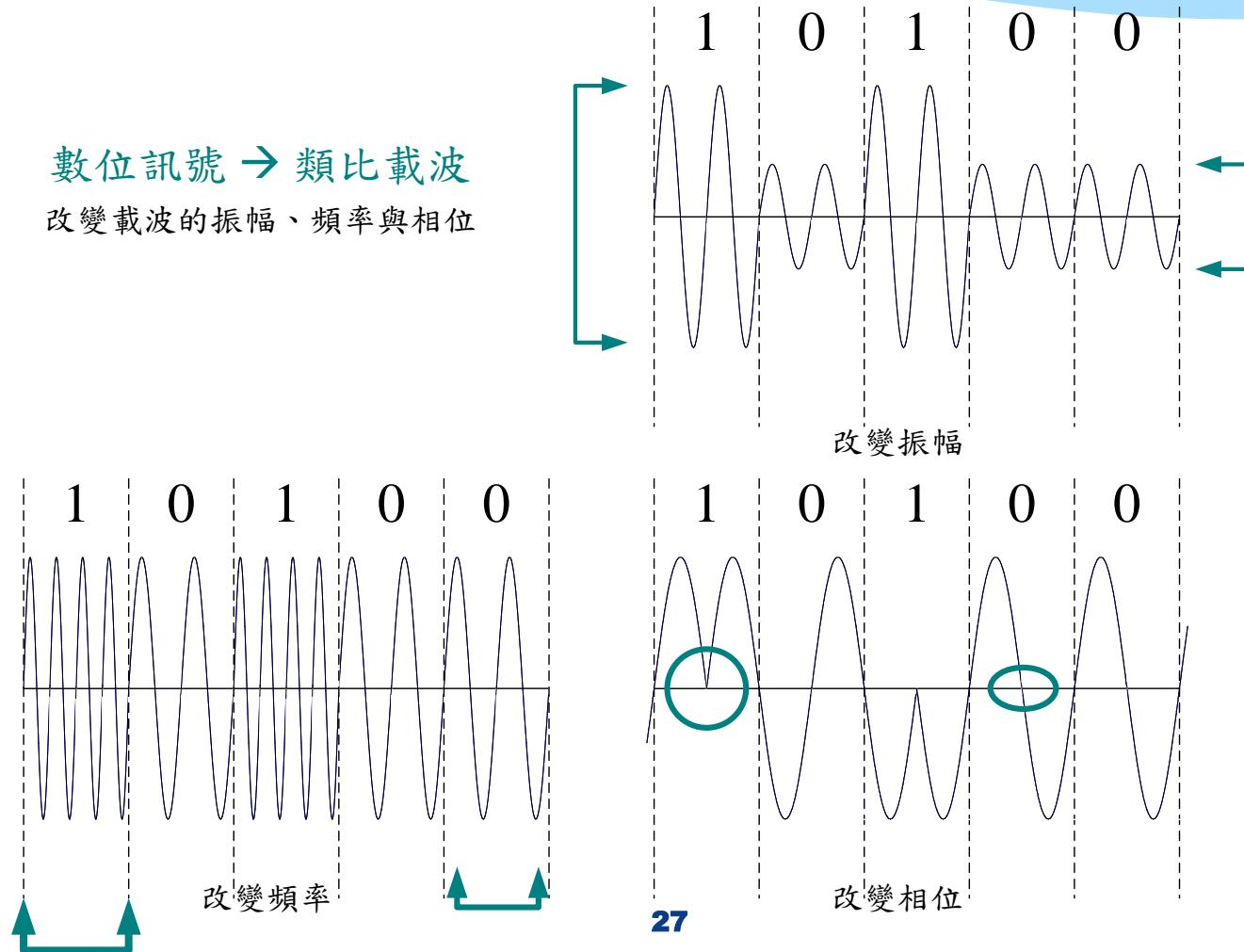
振幅小代表 0，振幅大代表 1

頻率低代表 0，頻率高代表 1

相位 0° 代表 0，相位 180° 代表 1

數位調變技術 (Digital modulation)

數位訊號 → 類比載波
改變載波的振幅、頻率與相位



數位調變技術的分類

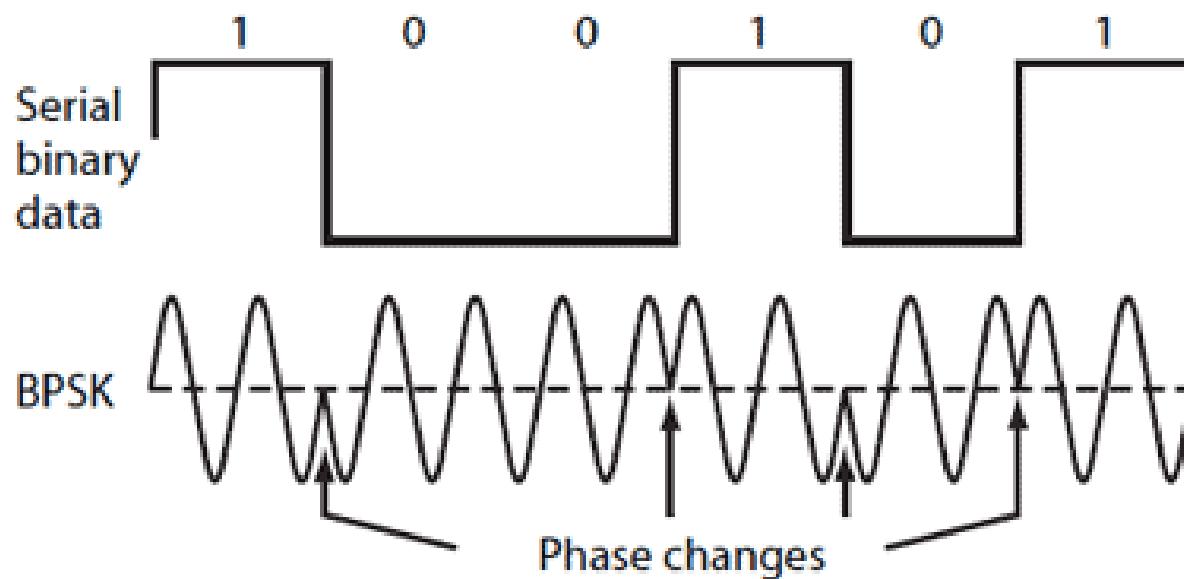
- * 線性振幅數位調變 (linear modulation)
- * 固定振幅數位調變 (constant envelope modulation)
- * 整合線性及固定振幅之數位調變 (combined linear and constant envelop modulation)
- * 展頻 (spread spectrum modulation)

線性振幅數位調變

- * 調整信號的相位角以表現 0 與 1。
- * 頻寬的使用顯得有效率，但輸出功率較差。
 - ◆ 二階相移鍵控（Binary Phase Shift Keying，BPSK）
 - ◆ 差動相移鍵控（Differential Phase Shift Keying，DPSK）
 - ◆ 四階相移鍵控（QuadriPhase Shift Keying，QPSK）

線性振幅數位調變技術 BPSK

- * 利用兩信號（0與1）間相位差180度（ π ）做調變。



線性振幅數位調變技術 BPSK

- * 利用兩信號（0與1）間相位差180度（ π ）做調變。

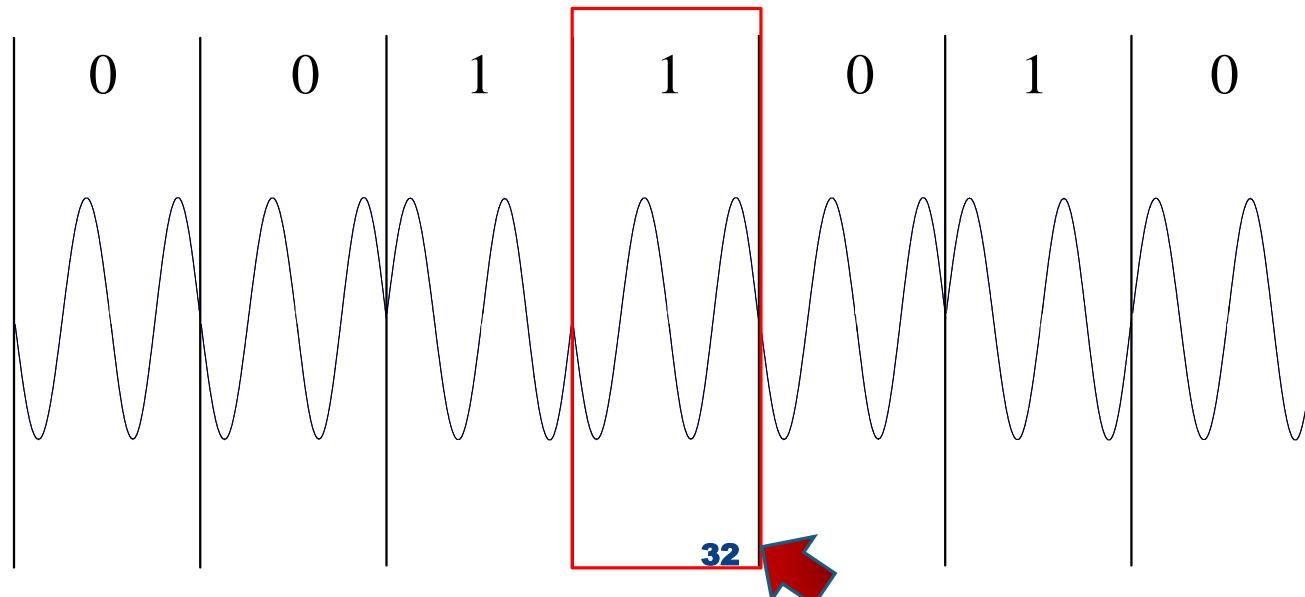
$$S_{BPSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \theta_c) \quad 0 \leq t \leq T_b$$

$$S_{BPSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t + \pi + \theta_c) \quad 0 \leq t \leq T_b$$

- ♦ f_c 表載波信號、 θ_c 表相位角
 - ♦ E_b 是每個位元的傳輸能量、 T_b 是每個位元的傳輸時間
- * 因為接收端要偵測信號的相位的變化，所以傳收雙方須要同步。

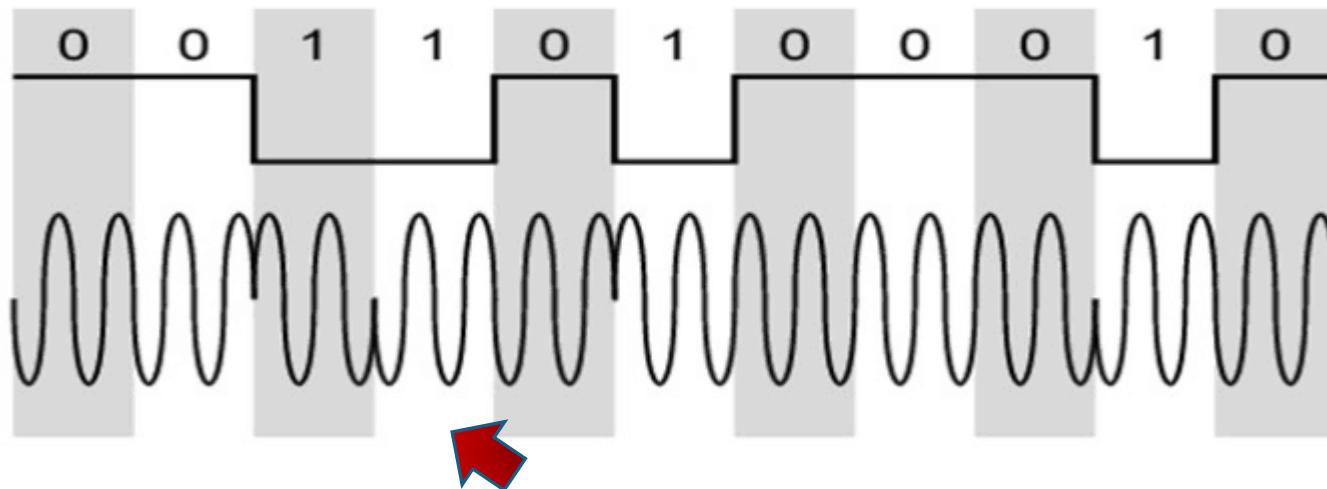
線性振幅數位調變技術 DPSK

- * 利用相位是否變化表示 0 與 1。
- * DPSK**不需要同步信號**以及一些相關的電路，所以它可以降低信號接收端電路的複雜性。
 - When the data is a High i.e., '1', the phase of the signal is reversed.



線性振幅數位調變技術 DPSK

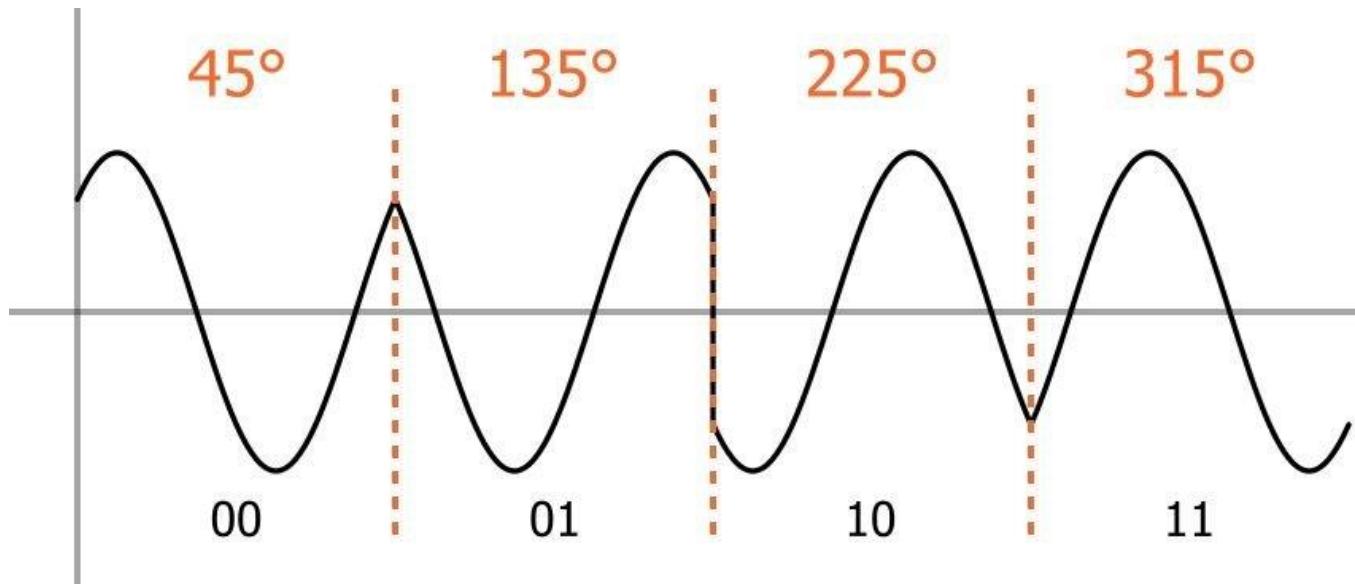
- When the data is a High i.e., '1', the phase of the signal is reversed.



<https://electronicspost.com/differential-phase-shift-keyingdpsk/>

線性振幅數位調變技術 QPSK

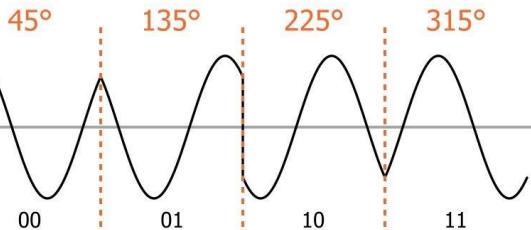
- We have 360° of phase to work with and four phase states, and thus the separation should be $360^\circ / 4 = 90^\circ$.
- So our four QPSK phase shifts are 45° , 135° , 225° , and 315° .



<https://www.allaboutcircuits.com/technical-articles/quadrature-phase-shift-keying-qpsk-modulation/>

線性振幅數位調變技術 QPSK

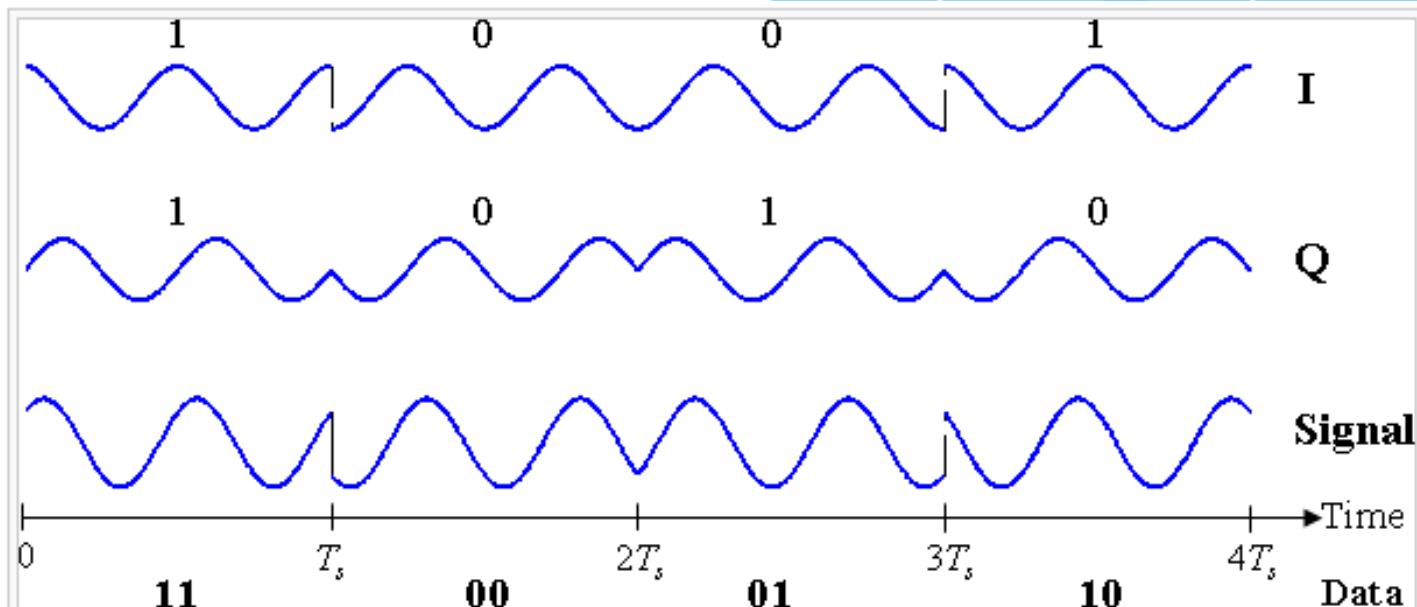
- 45° , 135° , 225° , and 315° : they are easily generated using **I/Q modulation** techniques because summing **I** and **Q** signals that are either **inverted** or **noninverted** results in these four phase shifts. The following table should clarify this:



I	Q	phase shift of I+Q
noninverted	noninverted	45°
inverted	noninverted	135°
inverted	inverted	225°
noninverted	inverted	315°

<https://www.allaboutcircuits.com/technical-articles/quadrature-phase-shift-keying-qpsk-modulation/>

線性振幅數位調變技術 QPSK



Timing diagram for QPSK. The binary data stream is shown beneath the time axis. The two signal components with their bit assignments are shown at the top, and the total combined signal at the bottom. Note the abrupt changes in phase at some of the bit-period boundaries.

The binary data that is conveyed by this waveform is: 11 0 0 0 11 0.

- The **odd** bits, highlighted here, contribute to the **in-phase** component: 1 1 0 0 0 1 1 0
- The **even** bits, highlighted here, contribute to the **quadrature-phase** component: 1 1 0 0 0 1 1 0

線性振幅數位調變技術 QPSK

- * 用 4 種不同相位的類比信號來表示長度為 2 個位元的數位信號。
 - ◆ 依序可以表示數位信號 00、01、10 及 11。

$$S_{QPSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left[2\pi f_c t + (i-1)\frac{\pi}{2}\right]$$

$$0 \leq t \leq T_b \quad i = 1, 2, 3, 4$$

- ◆ E_b 跟 T_b 分別代表信號的能量及傳輸位元的時間， i 的值由 1 依序到 4，代表傳輸信號有這 4 種可能。

固定振幅數位調變

- * 載波信號的振幅是一個常數。
- * 利用改變載波信號頻率來區分位元 0 與 1：
 - ◆ 二階頻移鍵控（Binary Frequency Shift Keying，BFSK）
 - ◆ 非連續性BFSK（Discontinuous Binary Frequency Shift Keying，D-BFSK）
 - ◆ 連續性BFSK（Continuous Binary Frequency Shift Keying，C-BFSK）。

二階頻移鍵控 BFSK (binary FSK)

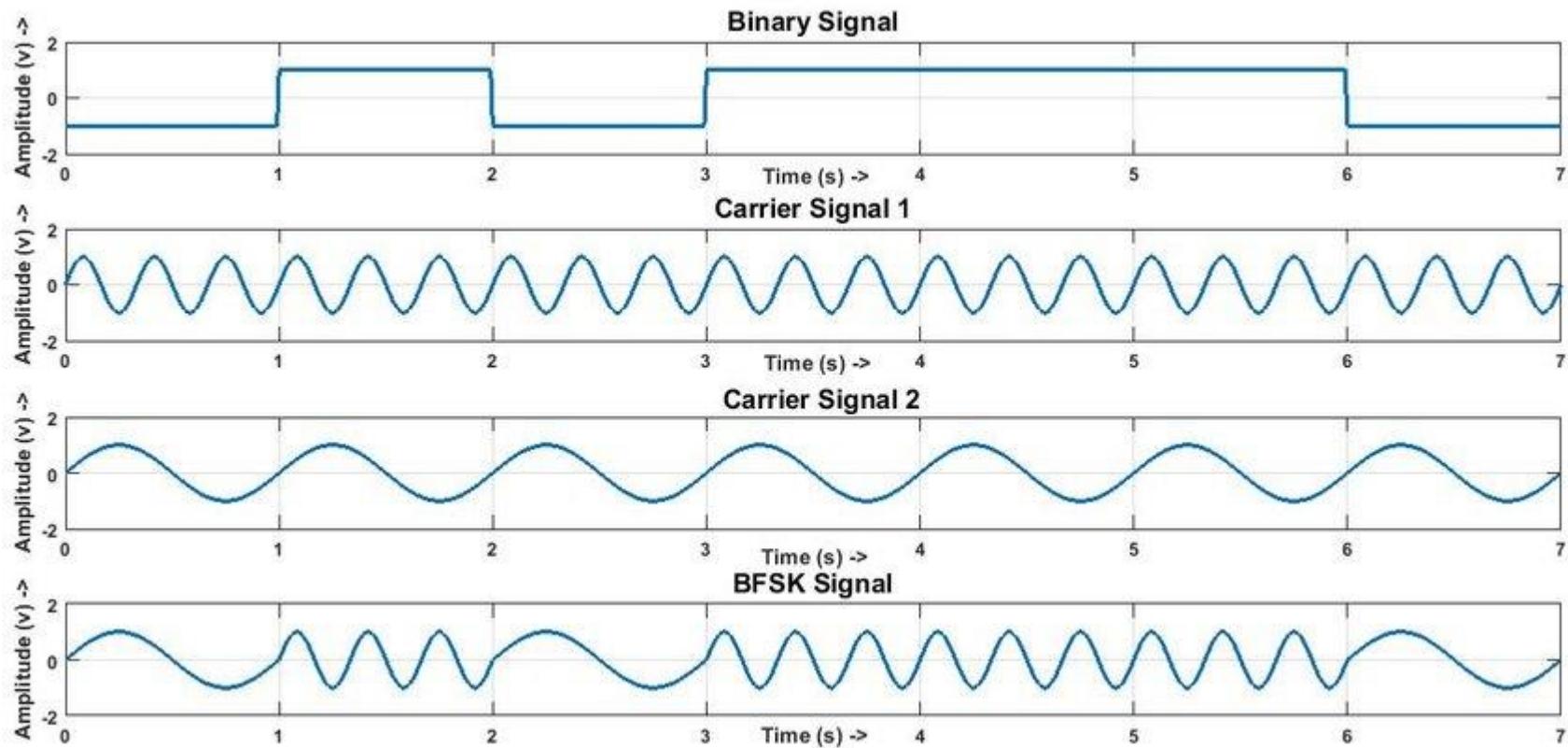
* 根據這個輸入信號0或1的值，在兩個頻率之間作交替變換的動作。

$$S_1 = \sqrt{\frac{2E}{T}} \cos(2\pi f_1 t) \rightarrow \text{represents '0'}$$

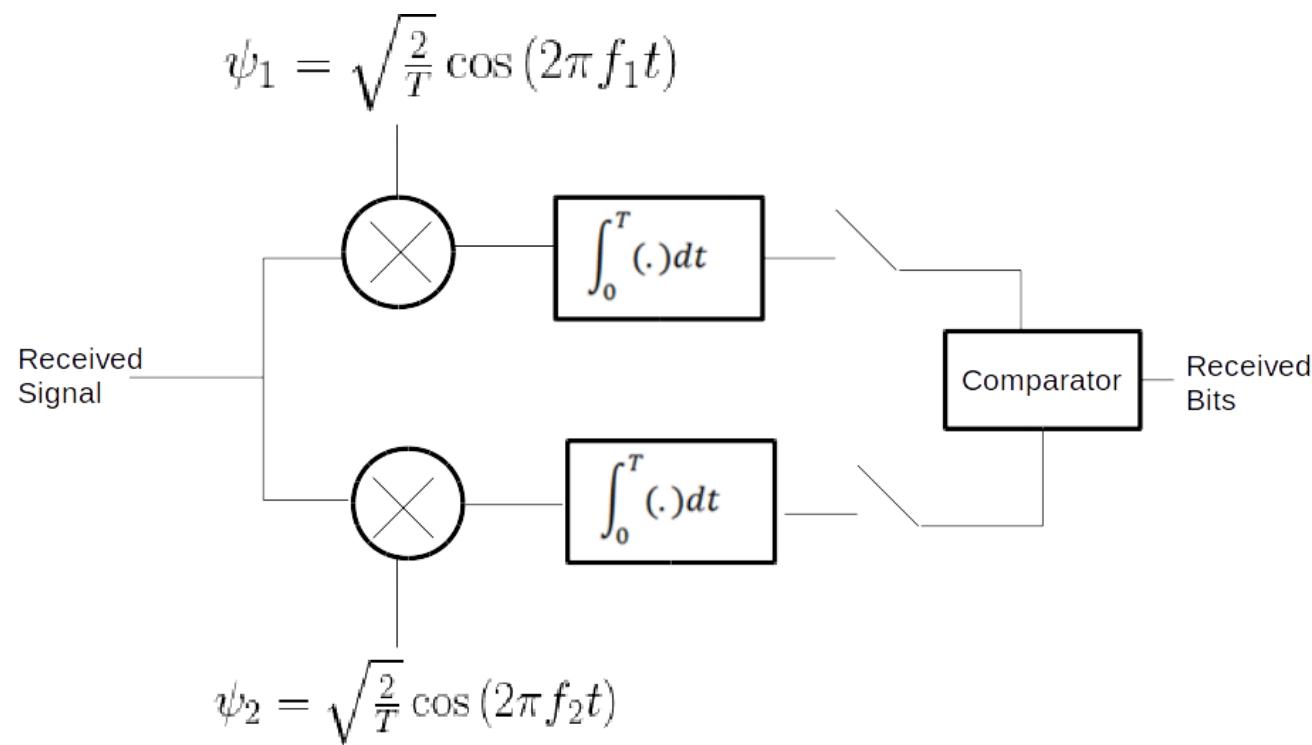
$$S_2 = \sqrt{\frac{2E}{T}} \cos(2\pi f_2 t) \rightarrow \text{represents '1'}$$

- ◆ f_1, f_2 表載波信號
- ◆ We represent '0' by carrier frequency 'f1' and '1' by carrier frequency 'f2'.
- ◆ E是每個位元的傳輸能量、T是每個位元的傳輸時間

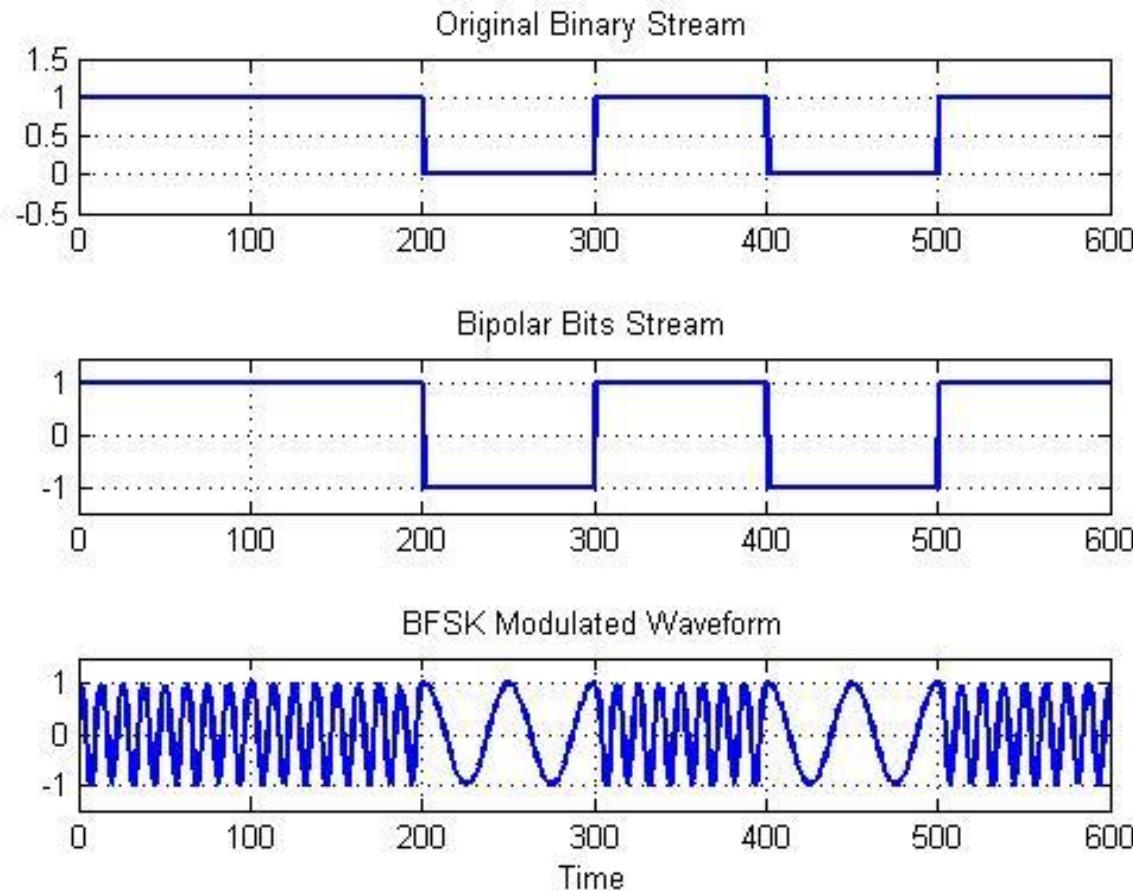
二階頻移鍵控 BFSK



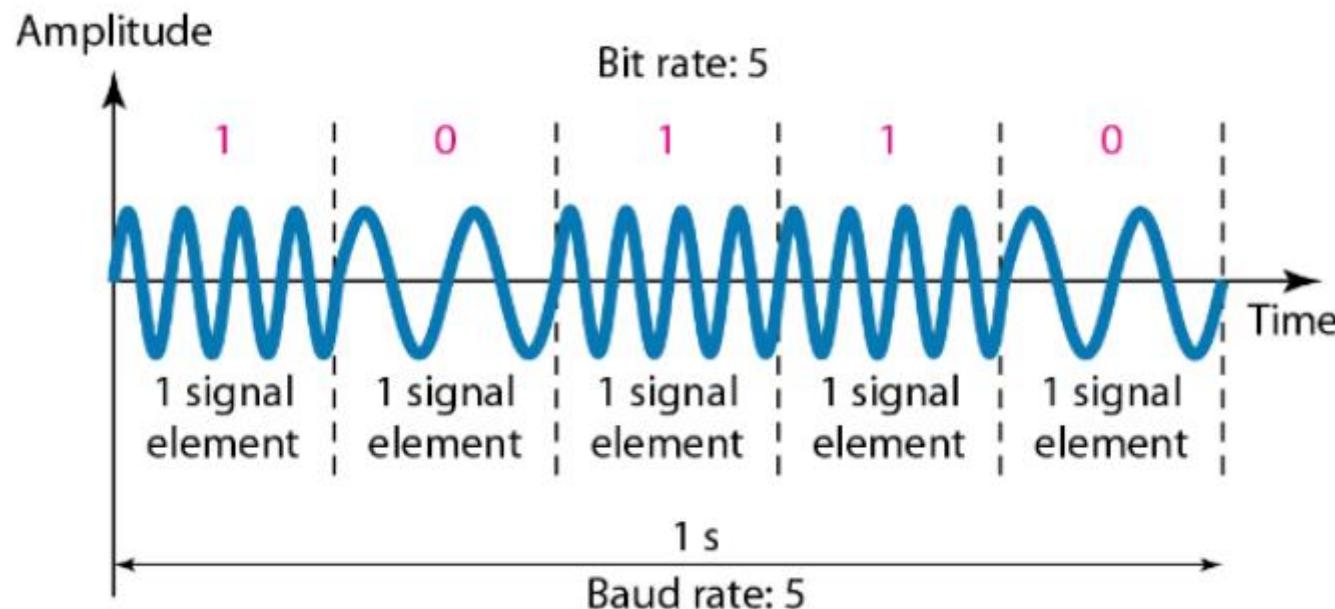
二階頻移鍵控 BFSK



二階頻移鍵控 BFSK



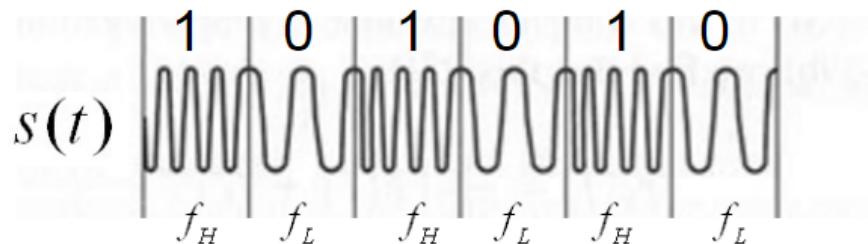
二階頻移鍵控 BFSK



http://www.uobabylon.edu.iq/eprints/publication_1_6905_1562.pdf

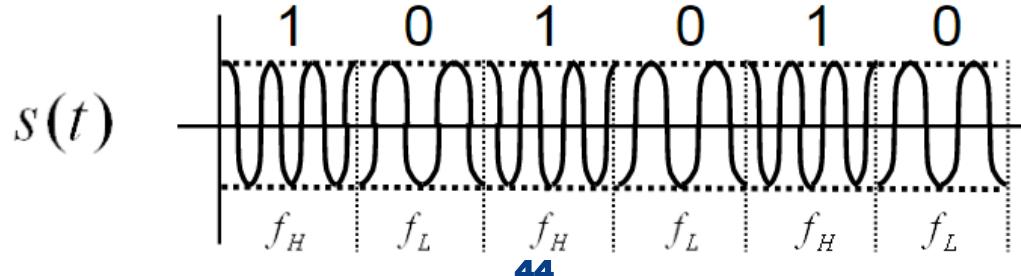
Frequency Shift Keying

- ◆ Frequency Shift Keying = FSK
- ◆ Two Major FSK Categories
 - Continuous Phase
 - » Phase between bit transitions is continuous → no abrupt change
 - Discontinuous Phase
 - » Phase between bit transitions is discontinuous → abrupt change



● Discontinuous Phase

» Phase between bit transitions is discontinuous → abrupt change



非連續性 (D-BFSK)

- * D-BFSK是利用兩個獨立、產生不同頻率的振盪器，而在這兩個振盪器中選定所代表的信號為0或為1。
- * 在信號交替的過程中，會造成不連續現象。

$$S_{FSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_H t + \theta_1) \quad 0 \leq t \leq T_b$$

$$S_{FSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_L t + \theta_2) \quad 0 \leq t \leq T_b$$

連續性 BFSK (C-BFSK)

- * C-BFSK技術能夠使得即使所欲表達的原始信號在不連續的情況下，而載波信號的相位，依然是連續的。
- * 技術上是以對信號進行積分的動作，如同下面的數學式。

$$S_{FSK}(t) = \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \theta(t)] = \sqrt{\frac{2E_b}{T_b}} \cos \left[2\pi f_c t + 2\pi \kappa_f \int_{-\infty}^t m(\eta) d\eta \right]$$

$$0 \leq t \leq T_b$$

整合線性及固定振幅之數位調變

- * 整合相位角、頻率改變兩種方式。
- * 同時利用載波信號振幅及相位的改變，就能夠代表更多種信號的狀態。
 - ◆ M-ary相移鍵控（M-ary Phase Shift Keying，MPSK）
 - ◆ M-ary正交振幅調變（M-ary Quadrature Amplitude Modulation，QAM）
 - ◆ M-ary頻移鍵控（M-ary Frequency Shift Keying，MFSK）

整合線性及固定振幅之數位調變技術

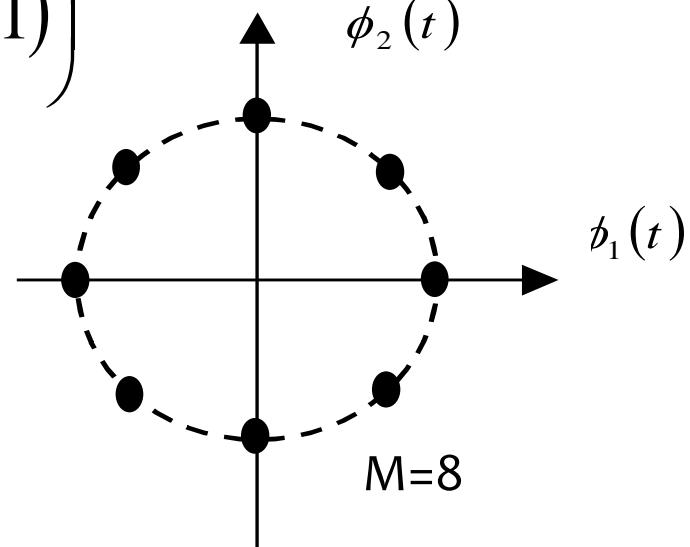
MPSK

- * 用一個固定的一個相位角 $2\pi/M$ ，來區隔所要代表的信號狀態。

$$S_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left(2\pi f_c t + \frac{2\pi}{M}(i-1)\right)$$

$$0 \leq t \leq T_b$$

$$i = 1, 2, \dots, M$$



整合線性及固定振幅之數位調變技術

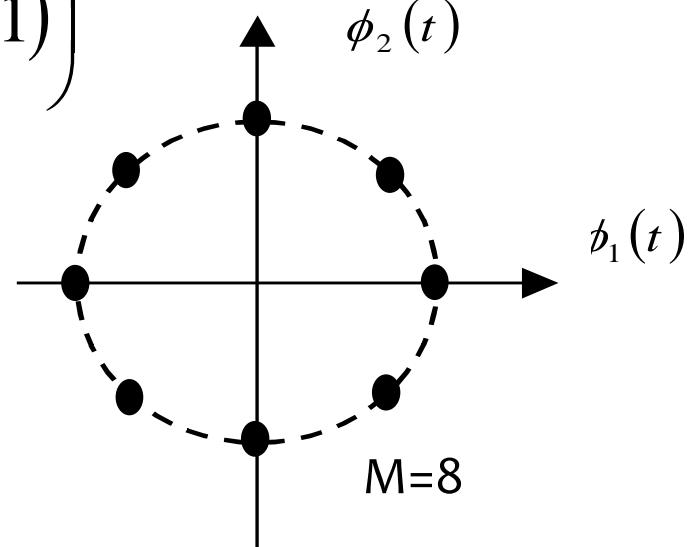
MPSK

- * 用一個固定的一個相位角 $2\pi/M$ ，來區隔所要代表的信號狀態。

$$S_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left(2\pi f_c t + \frac{2\pi}{M}(i-1)\right)$$

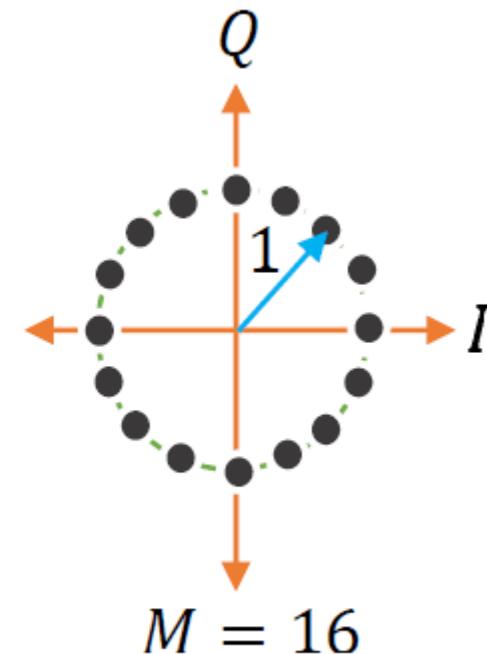
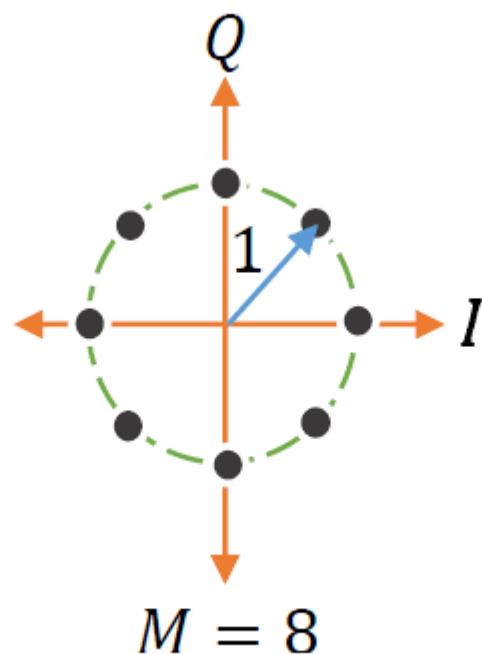
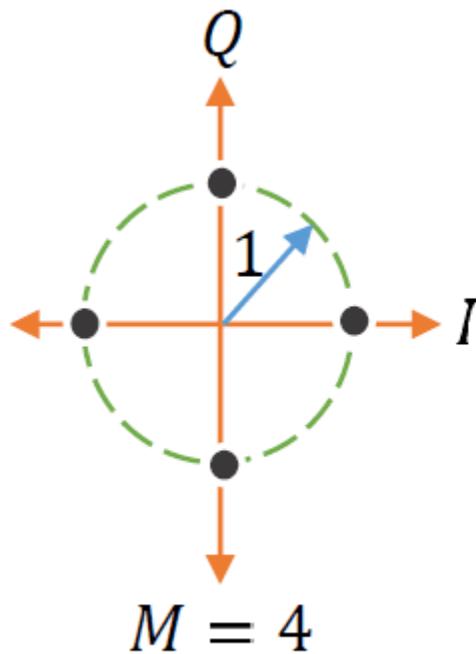
$$0 \leq t \leq T_b$$

$$i = 1, 2, \dots, M$$



整合線性及固定振幅之數位調變技術 MPSK

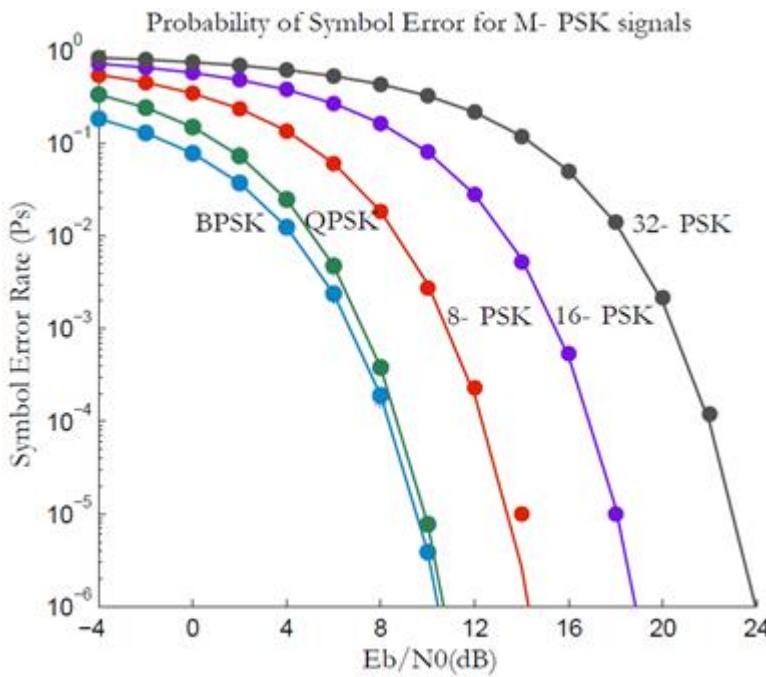
- * *Signal space constellations for various MPSK modulations*



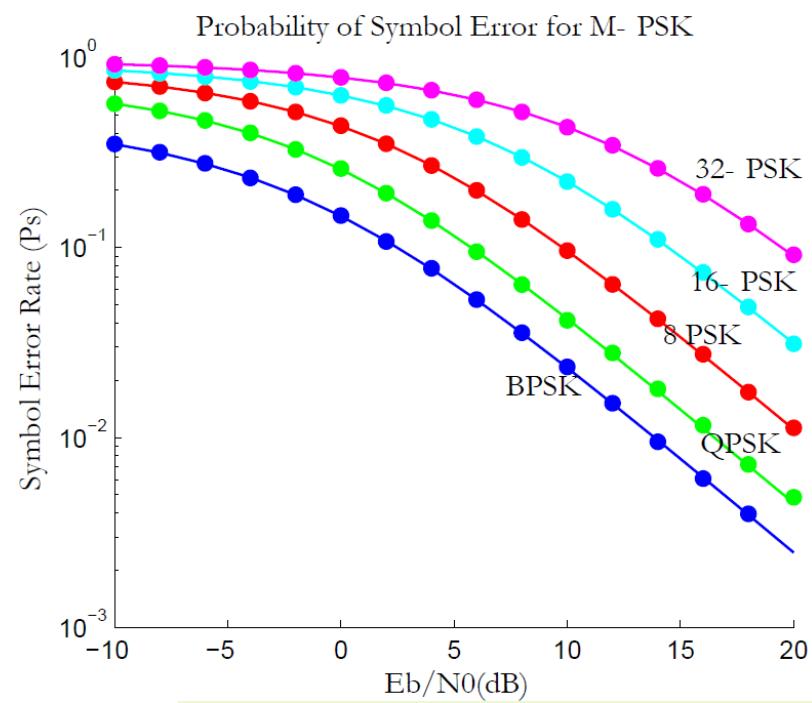
整合線性及固定振幅之數位調變技術

MPSK

- * The simulation results for error rate performance of M-PSK modulations over AWGN channel and Rayleigh flat-fading channel is given in the following figures.



Error rate performance of MPSK modulations in AWGN channel

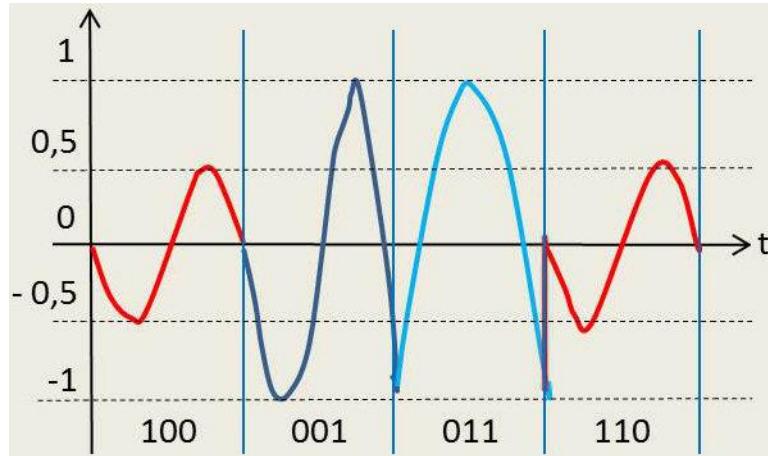


Error rate performance of MPSK modulations in Rayleigh flat-fading channel

整合線性及固定振幅之數位調變技術 (QAM)

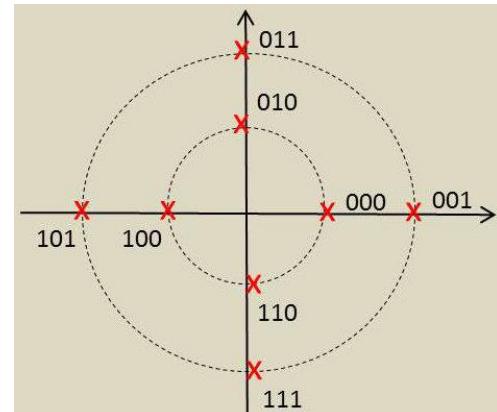
8QAM

Group bits	Amplitude	Phase shift
000	0.5	0
001	1	0
010	0.5	$\pi/2$
011	1	$\pi/2$
100	0.5	π
101	1	π
110	0.5	$3\pi/2$
111	1.	$3\pi/2$



正交振幅調變 (QAM, Quadrature Amplitude Modulation) 是一種在兩個正交 (Orthogonal) 載波 (\sin & \cos) 上進行振幅調變的調變方式。這兩個載波通常是相位差為 90 度 ($\pi/2$) 的正弦波，因此被稱作正交載波。

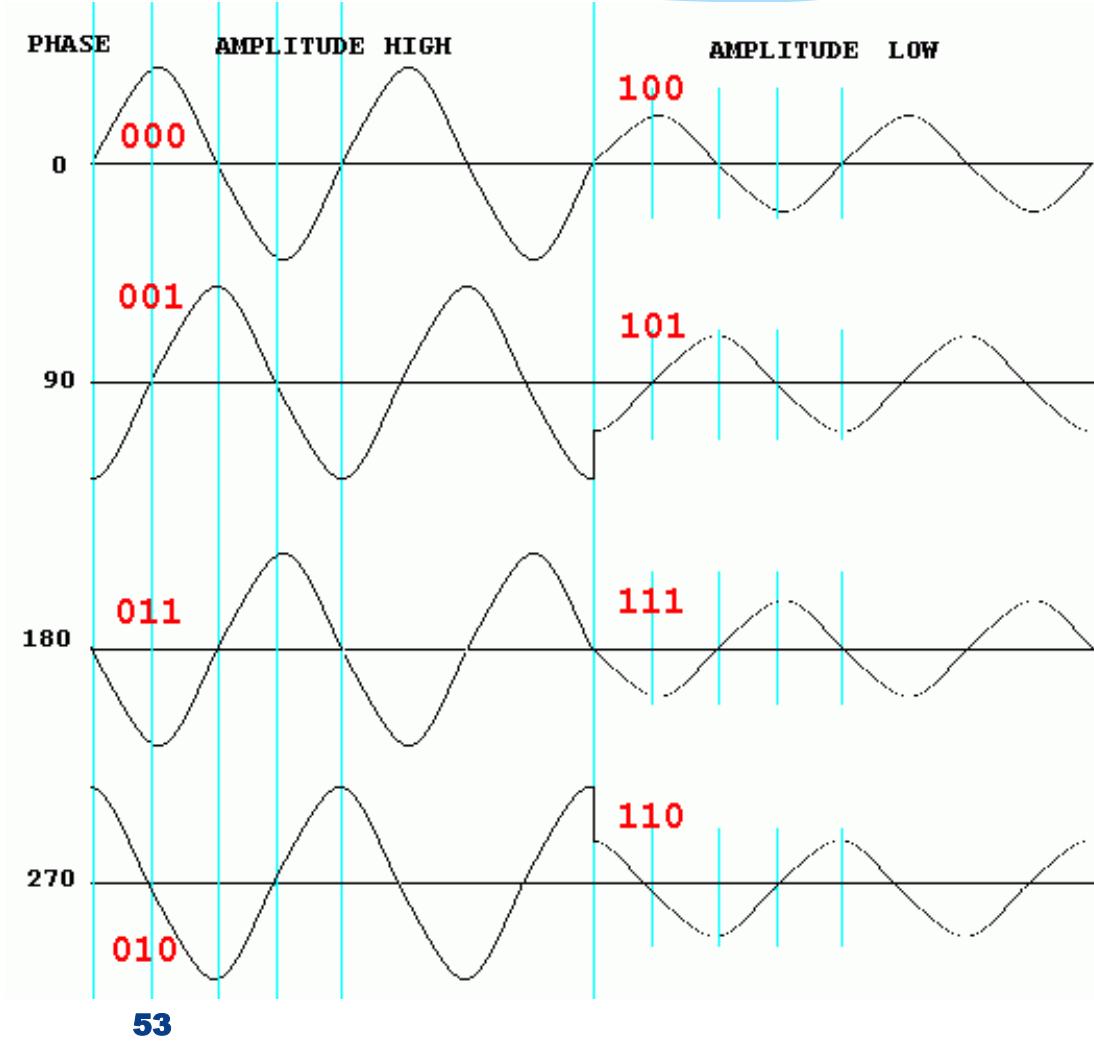
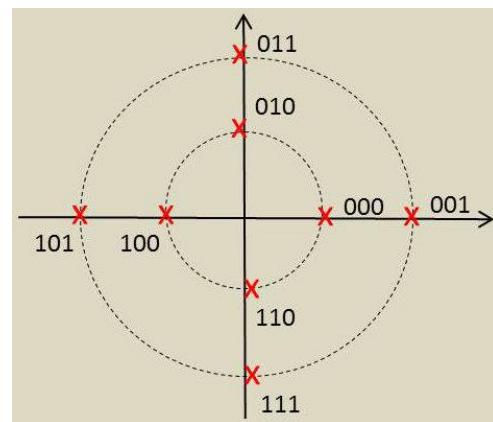
正交振幅調變 (Quadrature Amplitude Modulation, QAM)，是一種結合 PSK 和 ASK 的調變技術，利用兩個正交的載波進行振幅調變，檢測相位和振幅的量值。



整合線性及固定振幅之數位調變技術 (QAM)

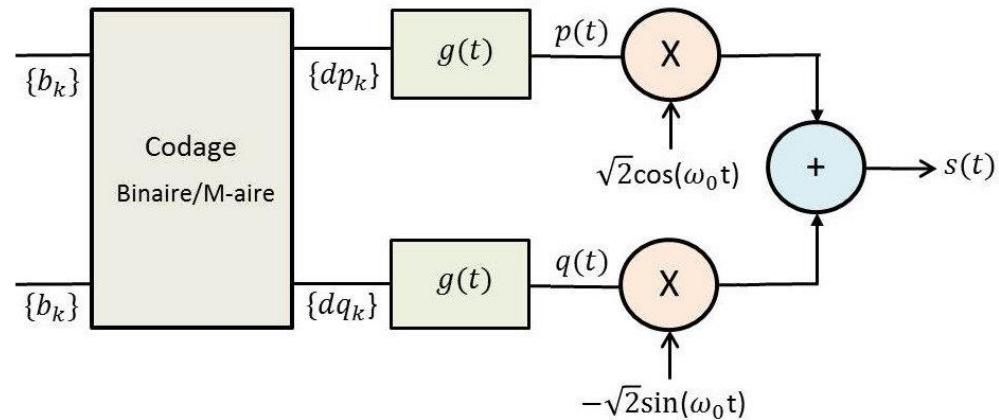
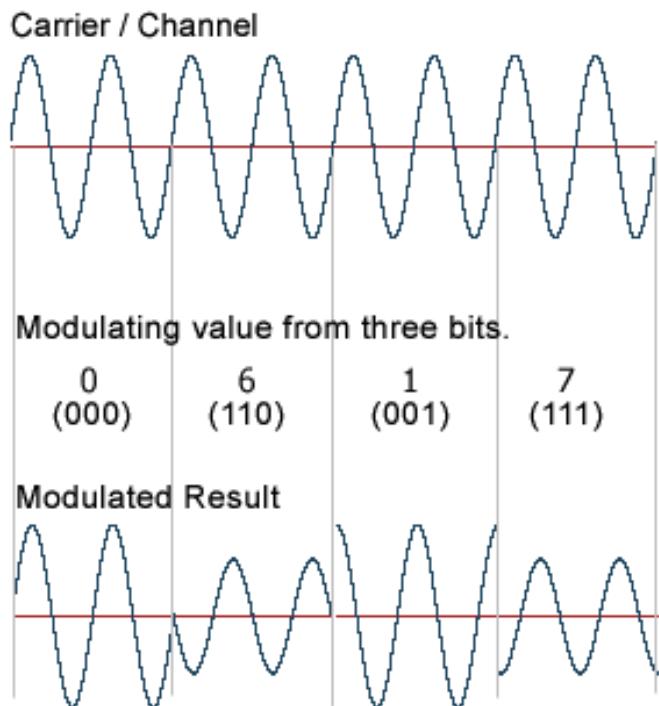
8QAM

Group bits	Amplitude	Phase shift
000	0.5	0
001	1	0
010	0.5	$\pi/2$
011	1	$\pi/2$
100	0.5	π
101	1	π
110	0.5	$3\pi/2$
111	1.	$3\pi/2$



整合線性及固定振幅之數位調變技術 (QAM)

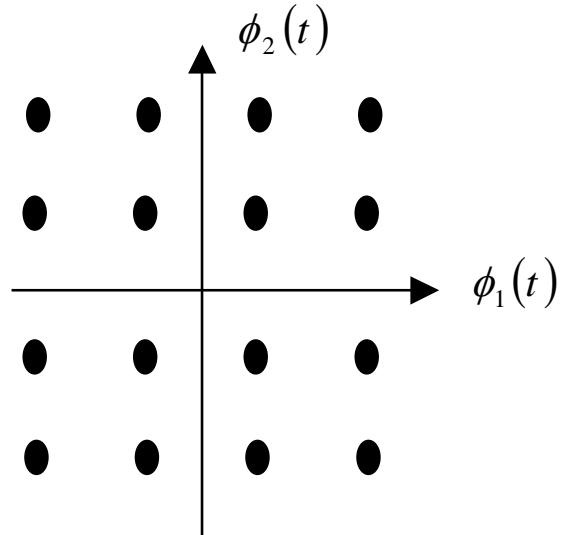
DIGITAL QAM (8QAM)



Note: Only four (0, 6, 1 and 7) out of the eight possible modulation states (0-7) are shown in this illustration.

整合線性及固定振幅之數位調變技術 (16-QAM)

- * 利用三角餘弦與正弦的函數的組合，分別代表我們所要代表的信號狀態。
- * 各個信號之間的**振幅**或者是**距離**並不是相等的



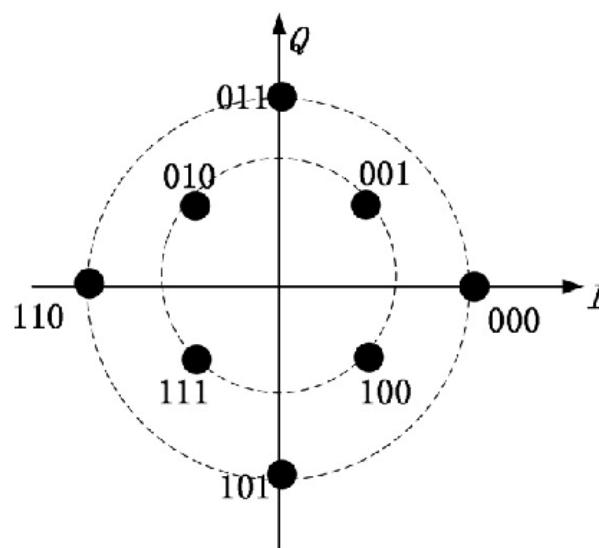
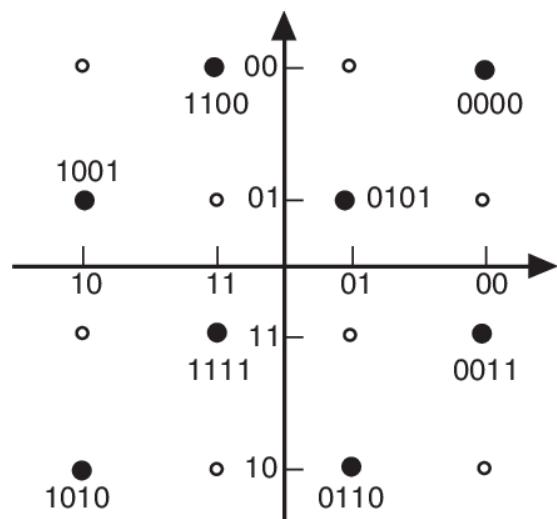
$$S_i(t) = \sqrt{\frac{2E_b}{T_b}}a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_b}{T_b}}b_i \sin(2\pi f_c t)$$

$$0 \leq t \leq T_b$$

$$M=16$$

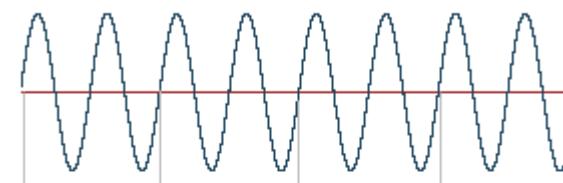
$$i = 1, 2, \dots, M$$

整合線性及固定振幅之數位調變技術 (16-QAM)



DIGITAL QAM (8QAM)

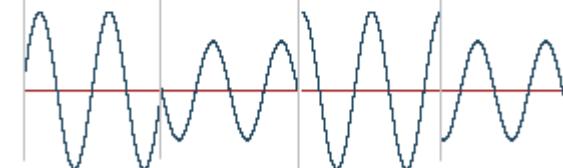
Carrier / Channel



Modulating value from three bits.

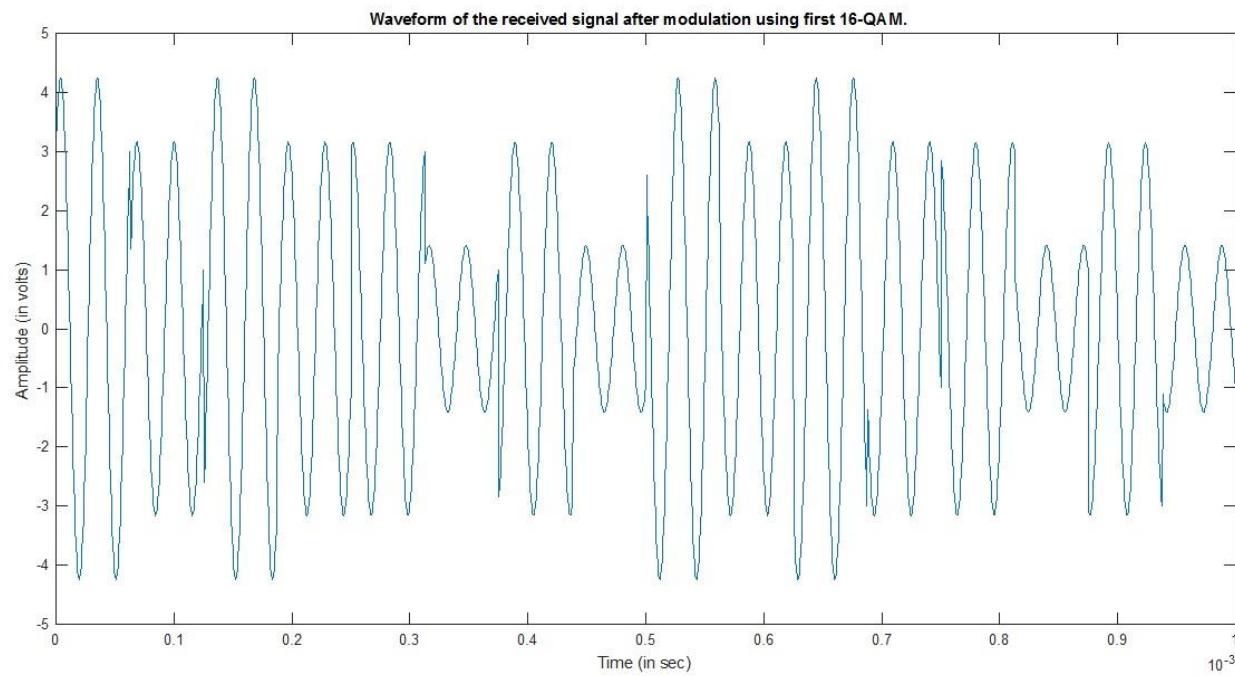
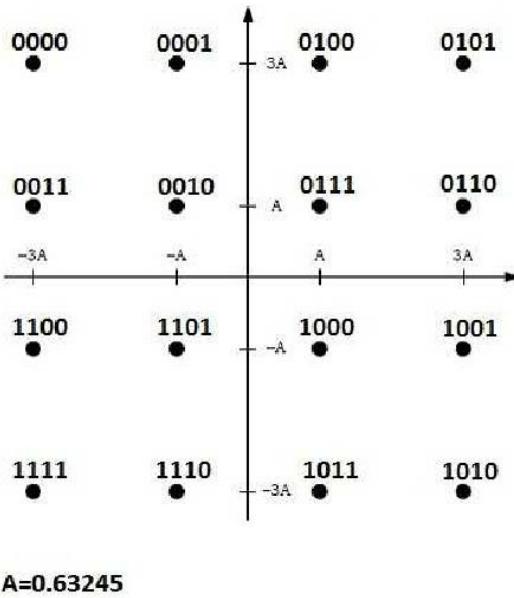
0 (000)	6 (110)	1 (001)	7 (111)
------------	------------	------------	------------

Modulated Result



Note: Only four (0, 6, 1 and 7) out of the eight possible modulation states (0-7) are shown in this illustration.

整合線性及固定振幅之數位調變技術 (16-QAM)



整合線性及固定振幅之數位調變技術 (QAM)

- * 下列哪一種調變技術，其載波的頻率及振幅都會隨著輸入訊號而變動？
(1) AM (Amplitude Modulation) (2) FM (Frequency Modulation)
(3) PSK(Phase Shift Keying) (4) QAM (Quadrature Amplitude Modulation)

Ans: 4

AM: 振幅變動。

FM: 頻率變動。

PSK: 相位角變動。

QAM: 可以為AM+PSK or AM+FSK。

M-ary 頻移鍵控 (MFSK)

- * 用載波頻率之間的一個區隔，來表示所欲傳送的數位信號。
- * 信號的頻率間隔，是二分之一的信號時間長短，同時彼此是互相垂直的（orthogonal）。

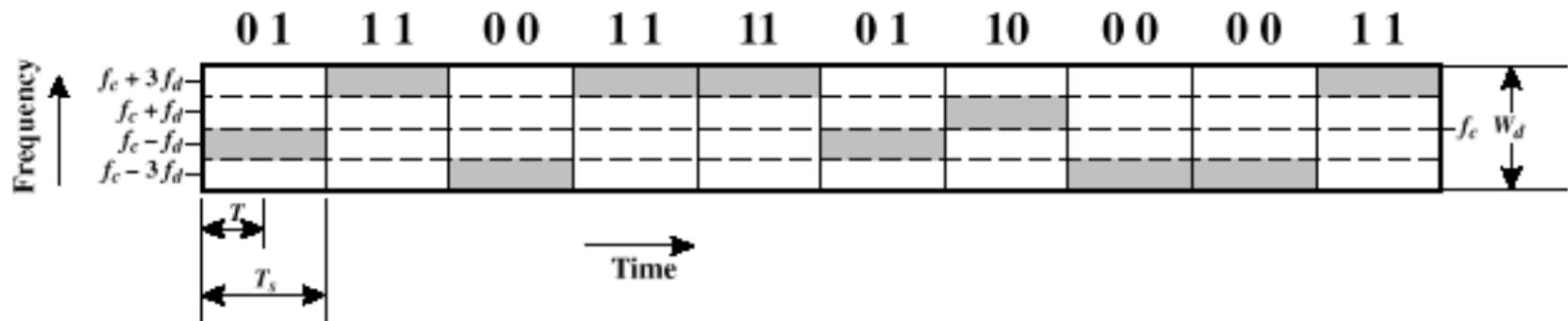
$$S_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left[\frac{\pi}{T_b}(n_c + i)t\right] \quad f_c = n_c/2T_b$$

$$0 \leq t \leq T_b$$

$$i = 1, 2, \dots, M$$

M-ary 頻移鍵控 (MFSK)

- * Assume M=4 → 4 frequencies
- * 20 bit stream: we send 2 bits per frequency
- * Note: $T_s = 2T_b = \text{Symbol period}$
- * Total BW = $2M \cdot f_d$



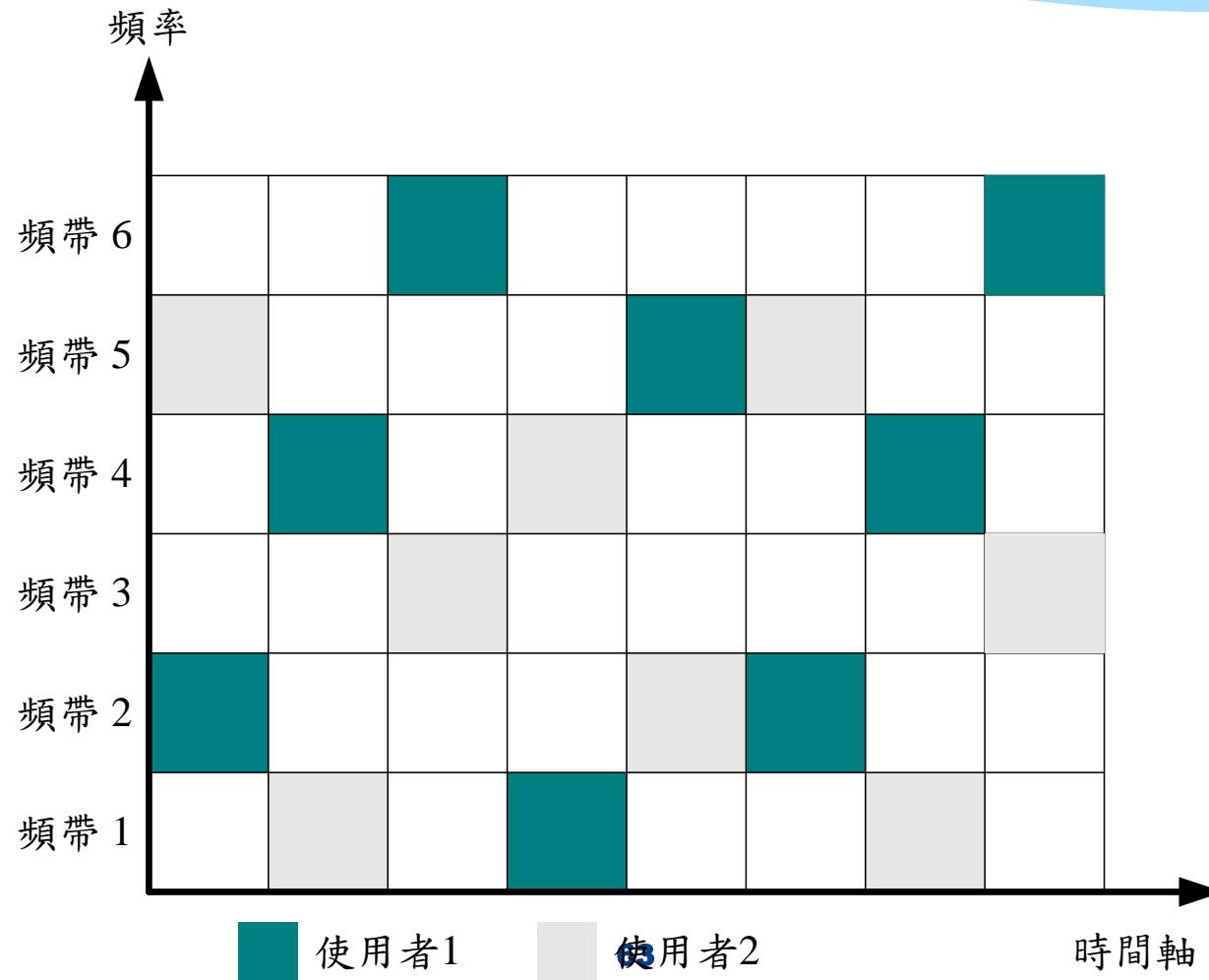
展頻

- * 前面所談的各種調變技術，是希望用於傳輸信號的頻寬，也就是**所使用通訊頻率的資源，能夠愈少愈好**，提供更多的使用者來分享無線電資源。
- * 相反的**展頻**顧名思義就是將所需要使用的頻帶的寬度，加以延展開來，使用更大的頻寬。
- * 展頻調變技術通常分為兩類技術：
 - ◆ 跳頻展頻（Frequency Hopping Spread Spectrum，FHSS）
 - ◆ 直接序列展頻（Direct Sequence Spread Spectrum，DSSS）

跳頻展頻（FHSS）

- * 在信號傳輸的過程中，週期性地改變所使用的載波頻率，將一個原始信號使用的頻道，分散到一個很大的頻寬中。
- * 在行動通訊系統中，跳頻被用於減輕干擾源的影響，增加系統的容量，以便服務更多的使用者。

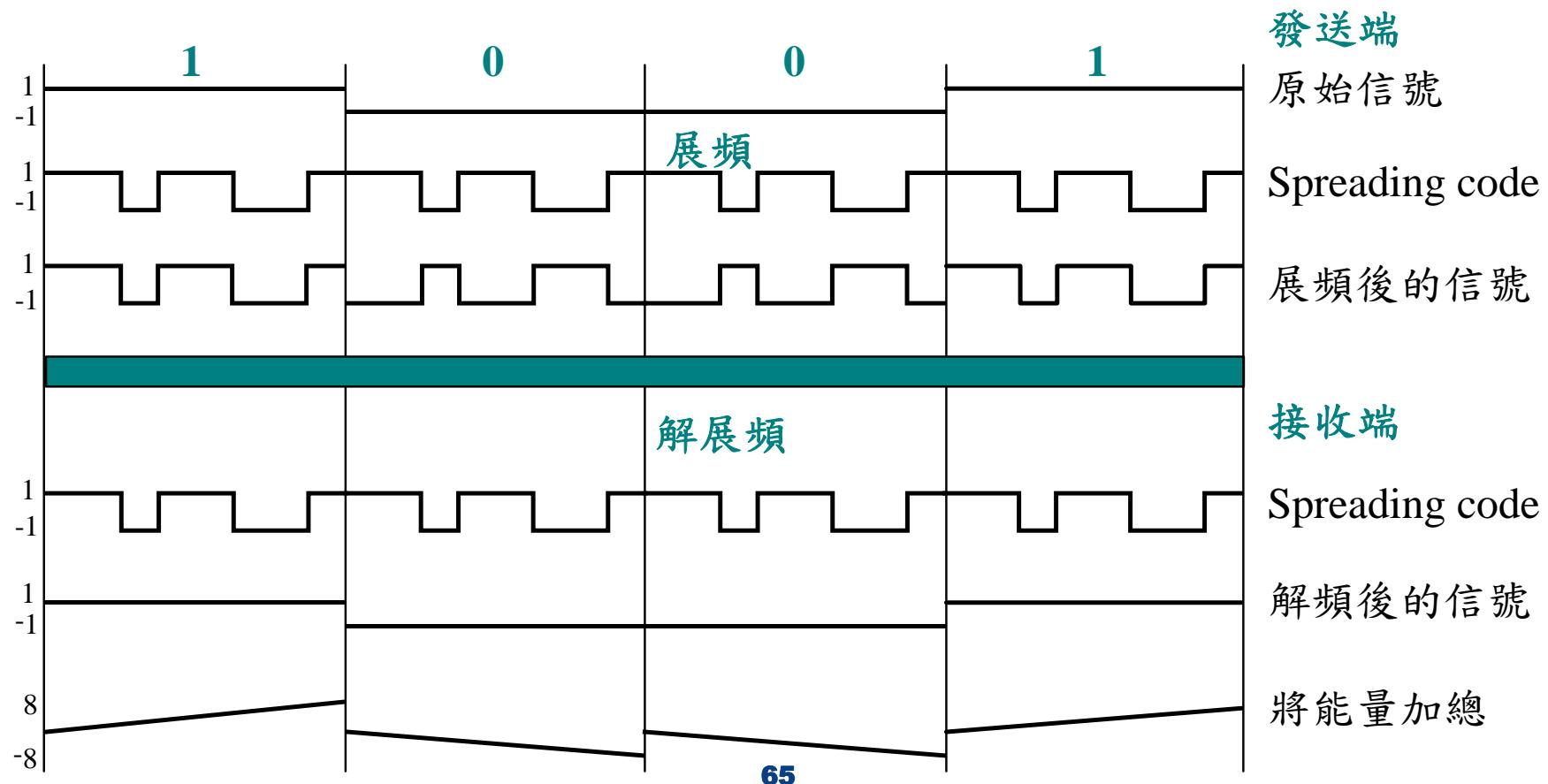
圖 2-13 跳頻展頻的示意圖



直接序列展頻 (DSSS)

- * 將要傳送的信號直接乘上一串很長的位元碼（像是一串128個位元的0101...0101，碼中每個位元被稱為chip）。
- * 一個0或1的原始信號，必須用許多的傳輸信號（chip）來表示，因此所需的頻帶會因此展開，成為一個寬頻的信號。
- * 由於展頻通訊是將信號延展開來使用很大的頻寬，因此傳送的能量也分散於整個頻寬中。
- * 較不怕受到其它無線電波的干擾，具有抗雜訊的優良能力。

圖 2-14 展頻與解展頻



DSSS vs., FHSS

■ 直接序列展頻 (DSSS)

❖ 足將原來窄頻的訊號，利用展頻碼(Spreadng Code)將原始訊號開展成數倍頻寬的訊號，這個展開的方法會將原來訊號的能量降低，甚至低於背景雜訊值(Background Noise)，這樣的機制讓有心截取資料的人不易判斷出真正的訊號，這也是軍方在通訊上採用這種技術的原因。

■ 跳頻展頻 (FHSS)

❖ 跳頻展頻則是使用頻寬十分狹窄的窄波(Narrowband)，利用窄波於不同頻道間跳躍的方式傳遞訊號，這種傳遞方式必須建立於訊號交換的二方其窄波跳躍模式(Pattern)一致，否則無法溝通。

傳送（調變）與接收（解調）

- * 數位調變技術的優點包括可以**偵錯與除錯**、可以**壓縮與解壓縮**、可以**加密與解密**、更好的**抗雜訊能力**等，我們所使用手機 2G 的 GSM / GPRS 、3G 的 UMTS 、4G 的 LTE / LTE-A 、無線區域網路（Wi-Fi ）、藍牙（Bluetooth ）等都是使用數位調變，顯然數位通訊是發展的趨勢。
- * 傳送端將數位訊號（0 與 1 ）轉變成不同的電磁波波形稱為「**調變（Modulation）**」；同理，接收端將不同的電磁波波形還原成數位訊號（0 與 1 ）稱為「**解調（Demodulation）**」，所有的通訊裝置一般都必須同時支援傳送（調變）與接收（解調），因此負責調變與解調的元件稱為「**調變解調器**」

多工技術（Multiplex）

- * 多人共同使用一條資訊通道的方法稱為「**多工技術**」（**Multiplex**），**Multicarrier Modulation** 簡單的說，天空只有一個，你的手機要去電磁波出去，我的手機也要去電磁波出去，兩種電磁波在天空中混在一起，接收端該如何區分那些是你的（和你通話的），那些是我的（和我通話的）呢？
- * 「調變技術」（Modulation）與「多工技術」（Multiplex）。
 - ◆ **調變技術**是將類比電磁波調變成不同的波形，來代表0與1兩種不同的數位訊號，這樣才能利用天線傳送到很遠的地方（這裡只談數位調變技術，不討論早期的AM、FM這種類比調變技術）。
 - ◆ **多工技術**則是將電磁波區分給不同的使用者使用，由於手機必須設計給所有的人使用，當每支手機都把電磁波丟到空中，該如何區分那個電磁波是誰的呢？

多工與多重存取

- * 為讓許多使用者同時間通訊，會使用到多工（duplex）技術與多重存取（multiple access）技術。
- * 多工與多重存取分屬兩個不同範疇的技術，但技術本身有許多相似性，故在此一併介紹。

多工接取技術

- * 多工接取（**multiple access**）是利用相同的實體通信媒介，讓很多的使用者同時傳送資料。
- * 如何區隔不同的使用者且避免彼此干擾，是多重存取的關鍵。
- * 中文翻譯裡不論是多工接取、多重存取、多重接進，還是多重接取，指的都是multiple access。

多工接取技術（Multiple Access）

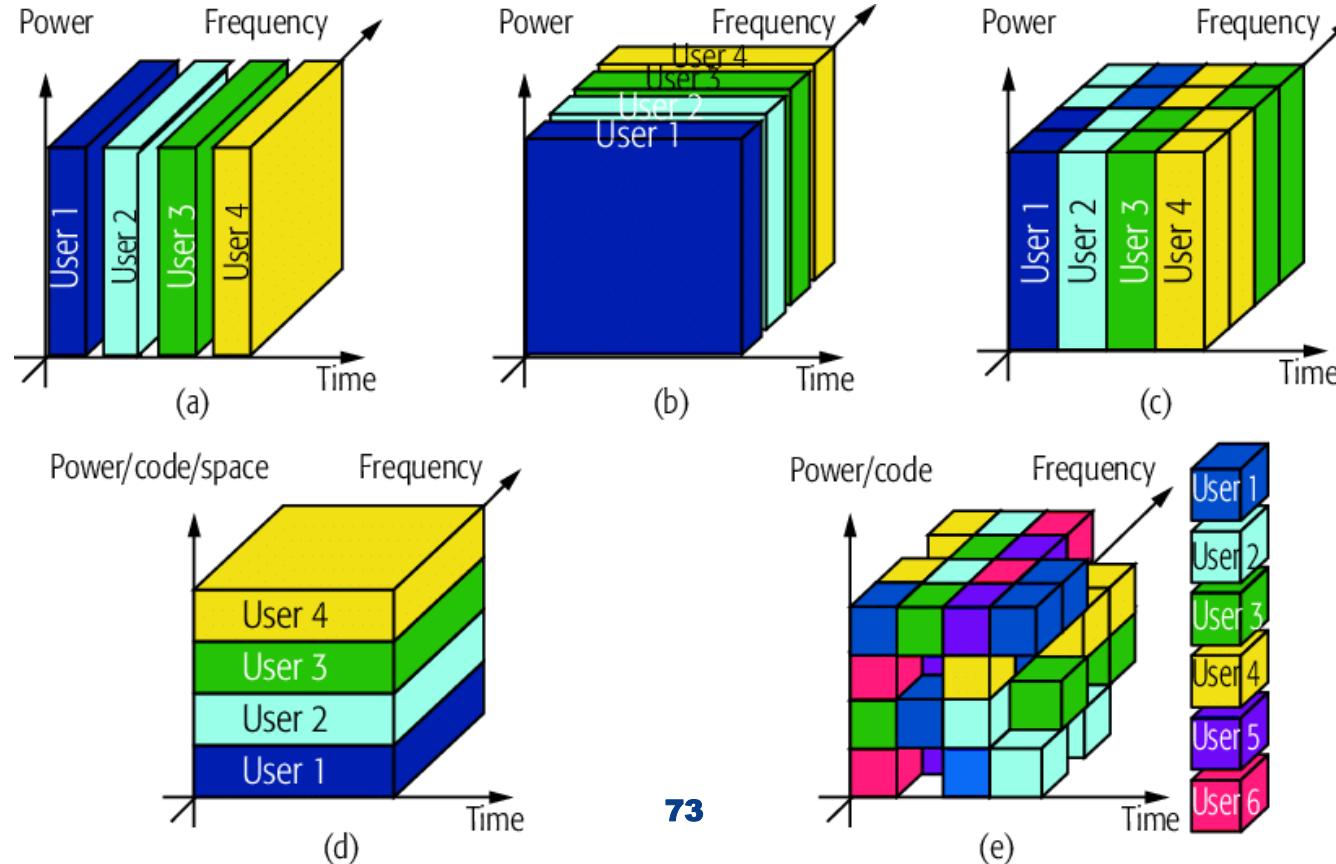
- * 多工技術的目的就是讓所有人使用，而且彼此不能互相干擾，為了增加資料傳輸率，可能必須同時使用兩種以上的多工技術，才能滿足每個人都要使用的需求。無線通訊常見的多工技術包括下列 4 種：
 - ◆ 分時多工接取（**TDMA**）：使用者依照「**時間先後**」輪流使用一條資訊通道，如圖三（a）所示，目前 2G 的 GSM / GPRS 系統有使用 TDMA。
 - ◆ 分頻多工接取（**FDMA**）：使用者依照「**頻率不同**」同時使用一條資訊通道，如圖三（b）所示，前面介紹每一對使用者必須使用「不同的頻率範圍」來通話，其實就是 FDMA，目前 2G 的 GSM / GPRS、3G 的 UMTS 有使用 FDMA。

多工接取技術（Multiple Access）

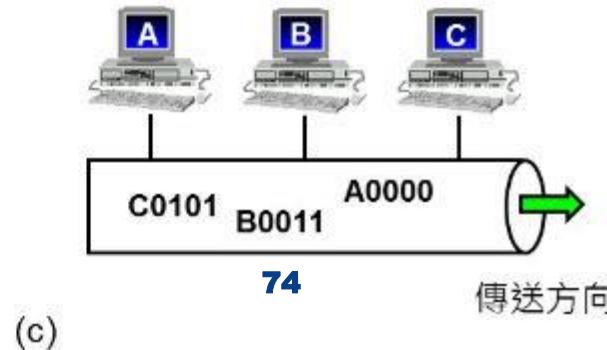
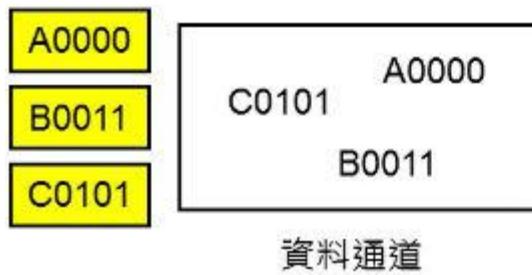
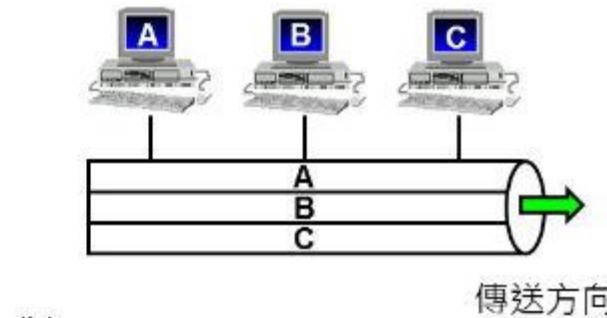
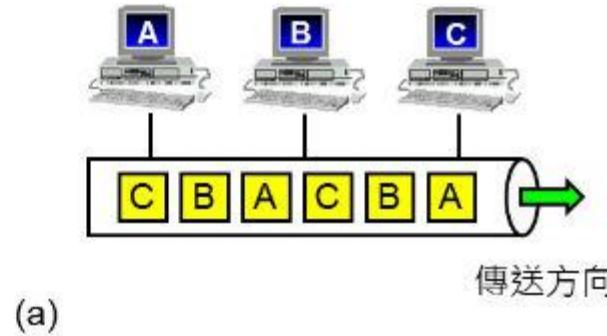
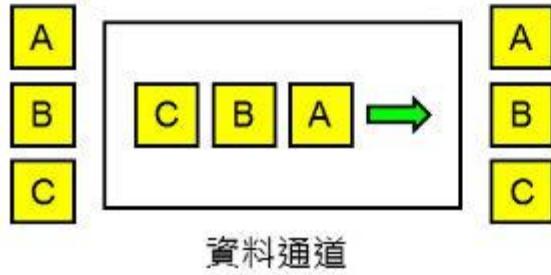
- ◆ **分碼多工接取（CDMA）**：將不同使用者的資料分別與特定的「密碼（Code）」運算以後，再傳送到資料通道，接收端以不同的密碼來分辨要接收的訊號，目前 3G 的 UMTS 有使用 CDMA。
- ◆ **正交分頻多工接取（OFDMA）**：前面介紹過分頻多工（FDMA）是使用者依照「頻率不同」同時傳送資料，而 OFDM 原理類似，唯一不同的是必須使用彼此「正交」的頻率，目前 4G 的 LTE / LTE-A、無線區域網路（IEEE802.11a/g/n）、數位電視（DTV）、數位音訊廣播（DAB）有使用 OFDM。
- ◆ **非正交分頻多工接取（NoMA）**：Non-orthogonal multiple access (NoMA)。

多工接取技術 (Multiple Access)

- * different multiple access schemes: a) TDMA; b) FDMA; c) OFDMA; d) CDMA/SDMA; e) possible NOMA solution.



TDMA / FDMA / CDMA

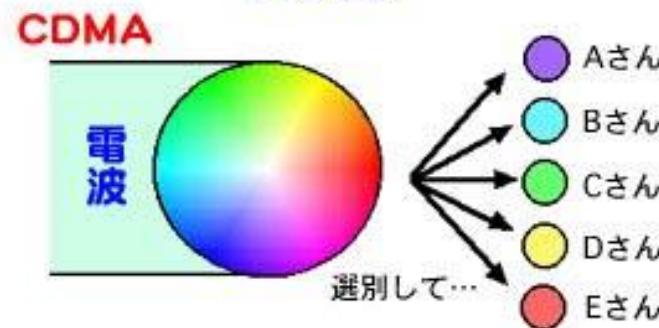
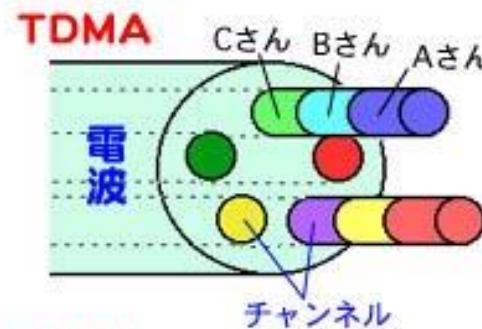
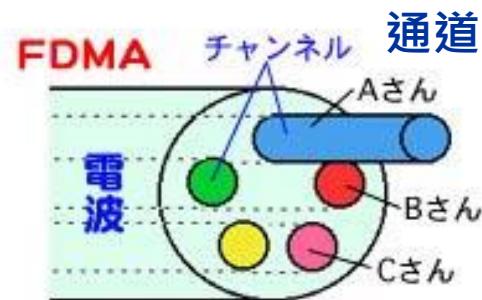
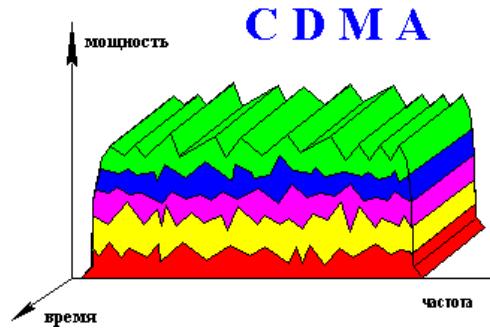
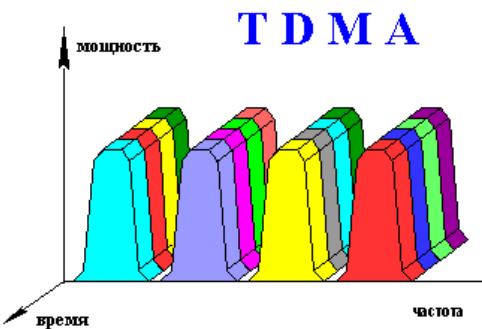
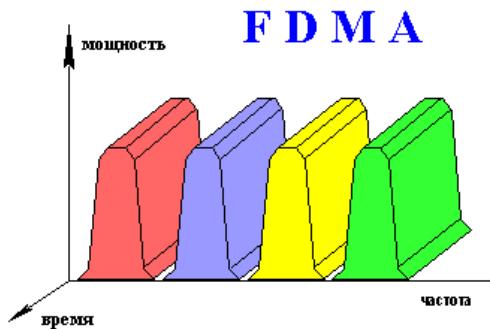


TDMA：依照時間先後輪流使用

FDMA：依照頻率不同同時使用

CDMA：將不同使用者的資料分別與特定的密碼運算

TDMA / FDMA /CDMA

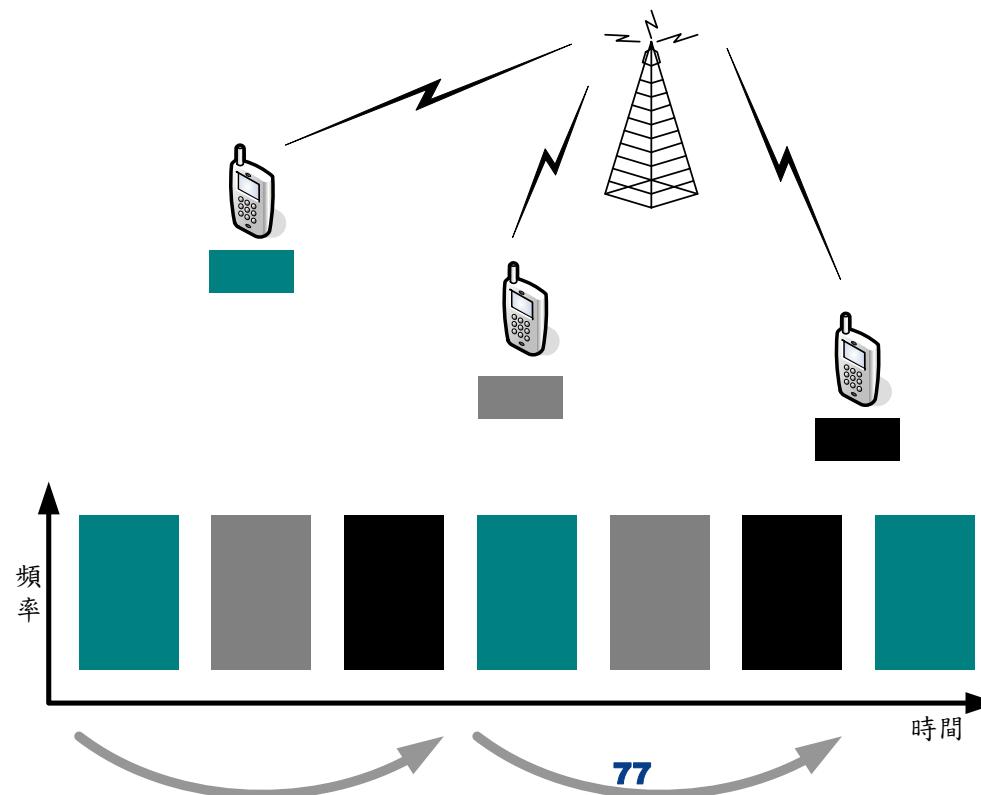


TDMA / FDMA /CDMA 的比喻

- * 多工技術比較複雜，我們可以想像在房子裡，甲與乙要講話，丙與丁要講話，戊與己要講話：
 - ◆ 分時多工接取（**TDMA**）：甲與乙先講一句，再換丙與丁講一句，再換戊與己講一句，依此類推，大家輪流（分時）講話彼此就不會互相干擾。
 - ◆ 分頻多工接取（**FDMA**）：甲與乙在客廳講話，丙與丁在書房講話，戊與己在臥室講話，大家在不同的房間（分頻）講話彼此就不會互相干擾。
 - ◆ 分碼多工接取（**CDMA**）：甲與乙用**中文**講話，丙與丁用**英文**講話，戊與己用**日文**講話，這樣雖然大家在同一個房子裡講話，各自仍然可以分辨出各自不同的語言，當甲與乙用中文講話時，丙與丁的英文以及戊與己的日文只是聲音干擾而已，不會造成甲與乙解讀中文的困擾；同理，當丙與丁用英文講話時，甲與乙的中文以及戊與己的日文只是聲音干擾而已，不會造成丙與丁解讀英文的困擾，在這個例子裡「用不同的語言講話」就好像「用不同的密碼加密」一樣。

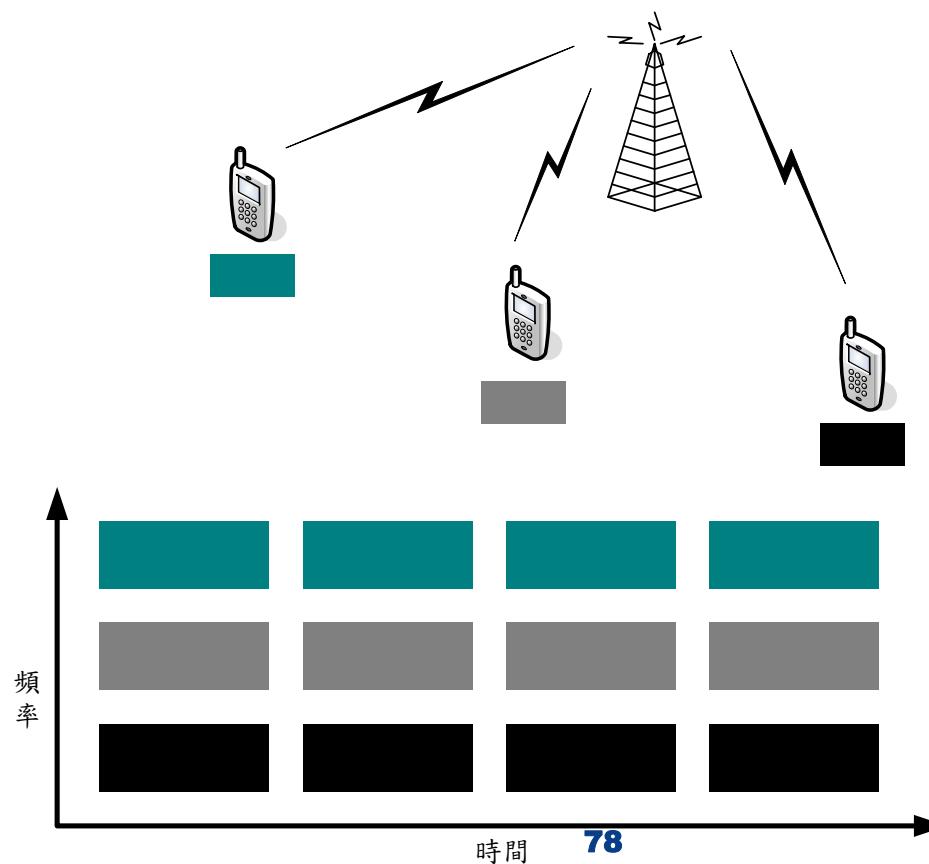
TDMA

- * TDMA用時間來區隔不同的使用者。分配每位使用者一個特定的時槽（time slot）輪流使用。



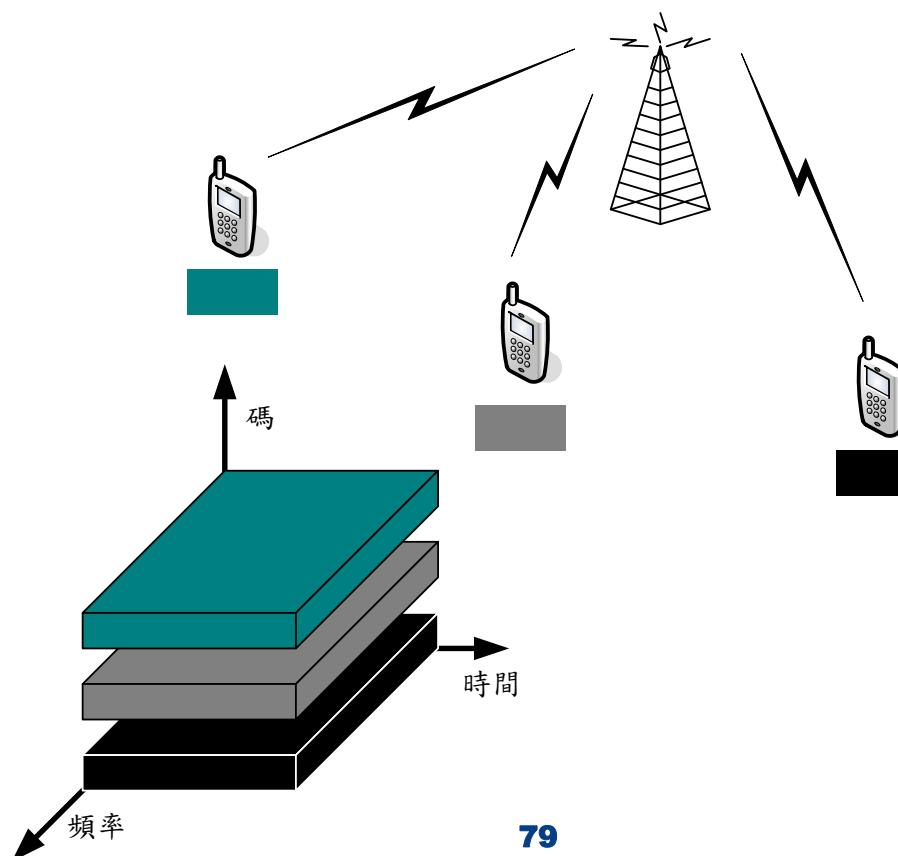
FDMA

- * FDMA 利用不同的傳送頻率，來區隔不同的使用者。



CDMA

- * CDMA用不同的碼來區隔不同的使用者。使用於展頻通訊系統。



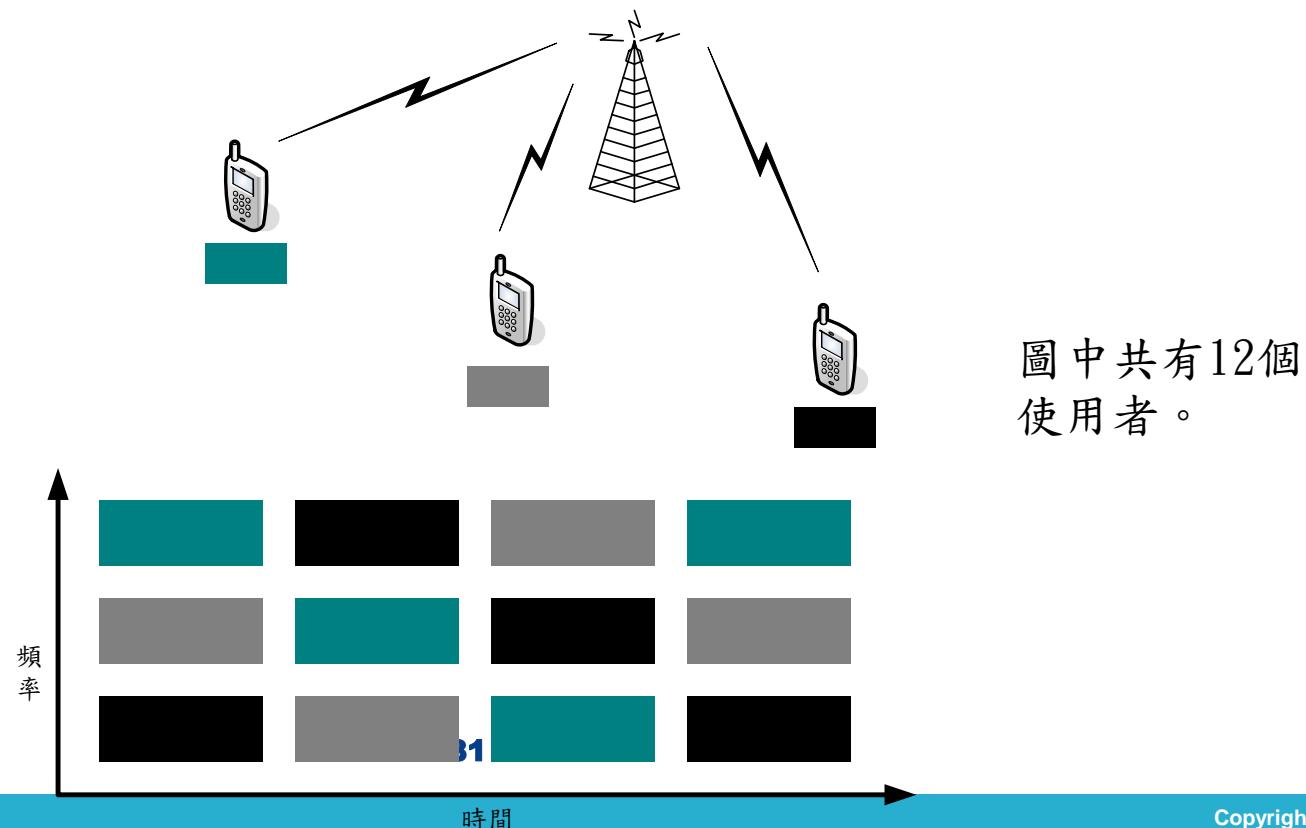
圖中共有3個使用者。

CDMA

- * 每一個使用者的原始資料在展頻的過程中，會與被分配到一個特殊的碼（code）與原始信號做運算。之後將所有使用者展開後的寬頻訊號混合，透過同一通道送出。接收端收到訊號後，再以同一特殊的碼做運算，便可萃取出原始信號。
- * 為了要區別各個使用者，這些特殊的碼之間彼此是兩兩正交（orthogonal）。
- * 屬於寬頻的通訊技術。

分時分頻多工存取技術

- * 在第二代的個人通訊服務系統中，常結合FDMA與TDMA使用。



OFDMA

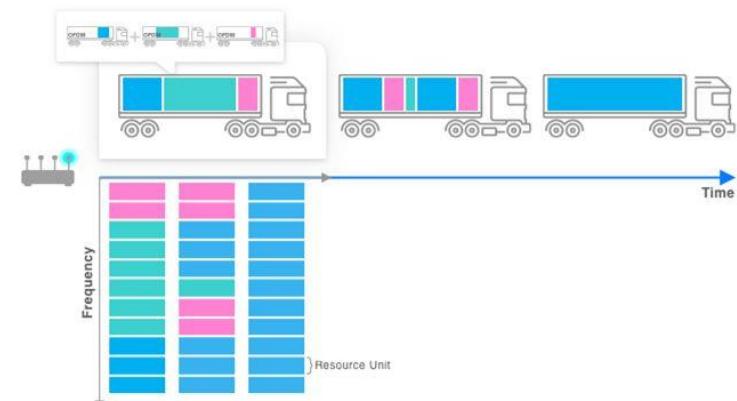
- * OFDMA是基於OFDM數位調變技術而生的multi-user演進版本
- * 在OFDM與OFDMA兩種技術中，當傳輸資料時，皆會將資料分割成數個較小的封包來進行傳送。
- * OFDMA更將通道再細分為數個subcarrier，讓不同用戶的數據可以被同時傳輸。



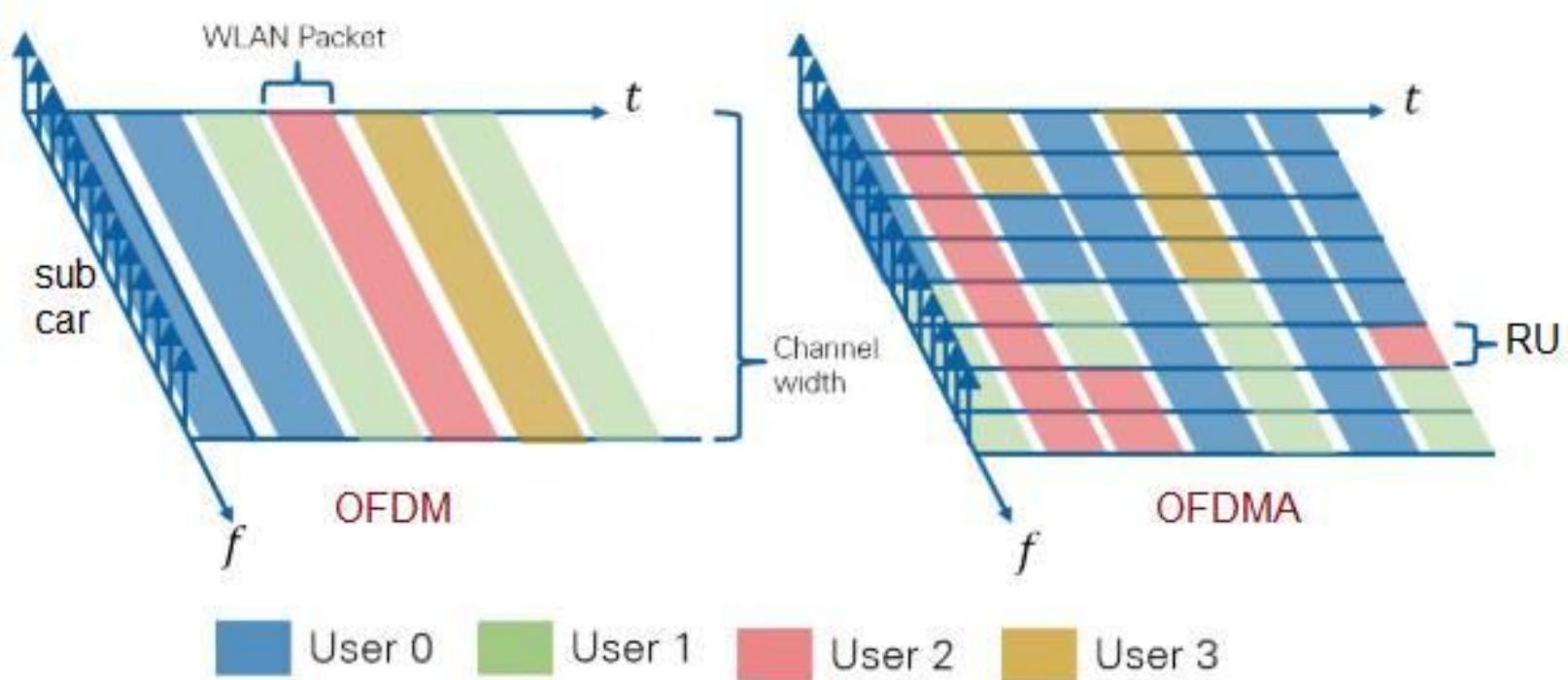
OFDM



OFDMA

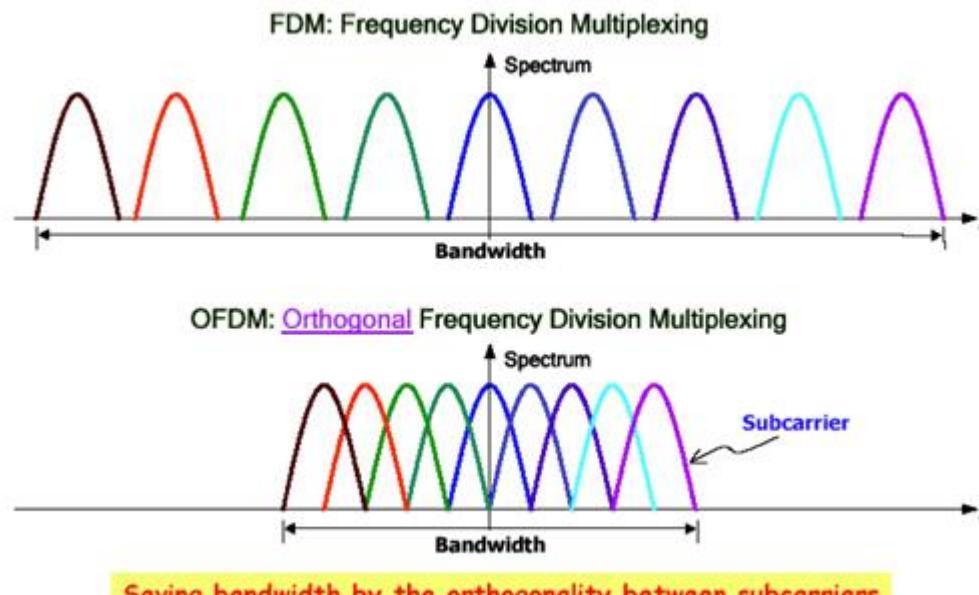


OFDM / OFDMA



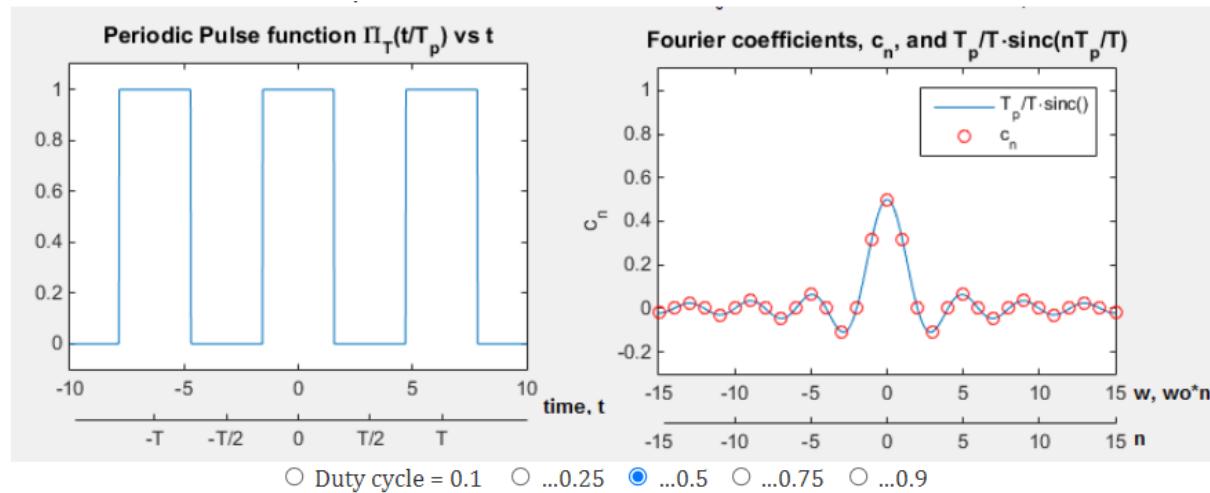
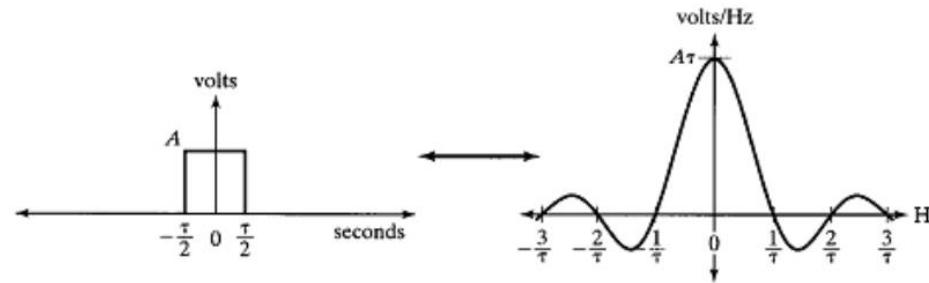
OFDM

- * 頻寬資源有限，若頻譜上載波可以重疊使用，那就可以提高頻譜效率(Spectrum Efficiency, η)，所以有學者提出正交分頻多工(Orthogonal Frequency Division Multiplexing, OFDM)的技術架構



Fourier Transform

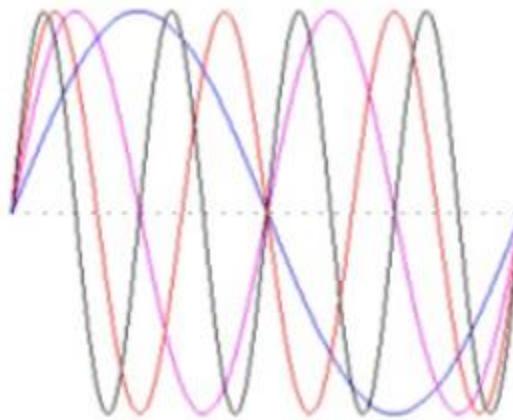
* Recall from Fourier Transform:



Fourier Transform

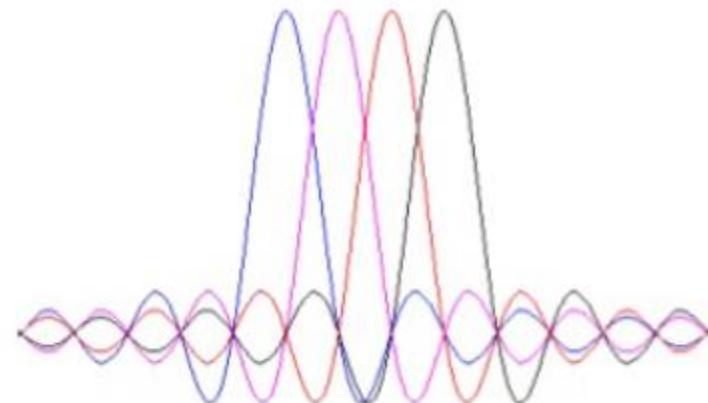
* What Does Orthogonality Mean?

Time domain



Example of four subcarriers within one OFDM symbol

Frequency domain

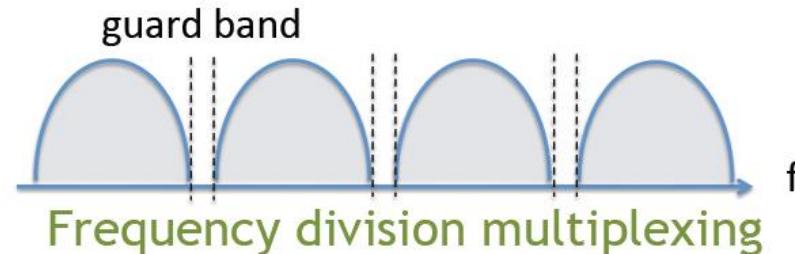


Spectra of individual subcarriers

Difference between FDM and OFDM

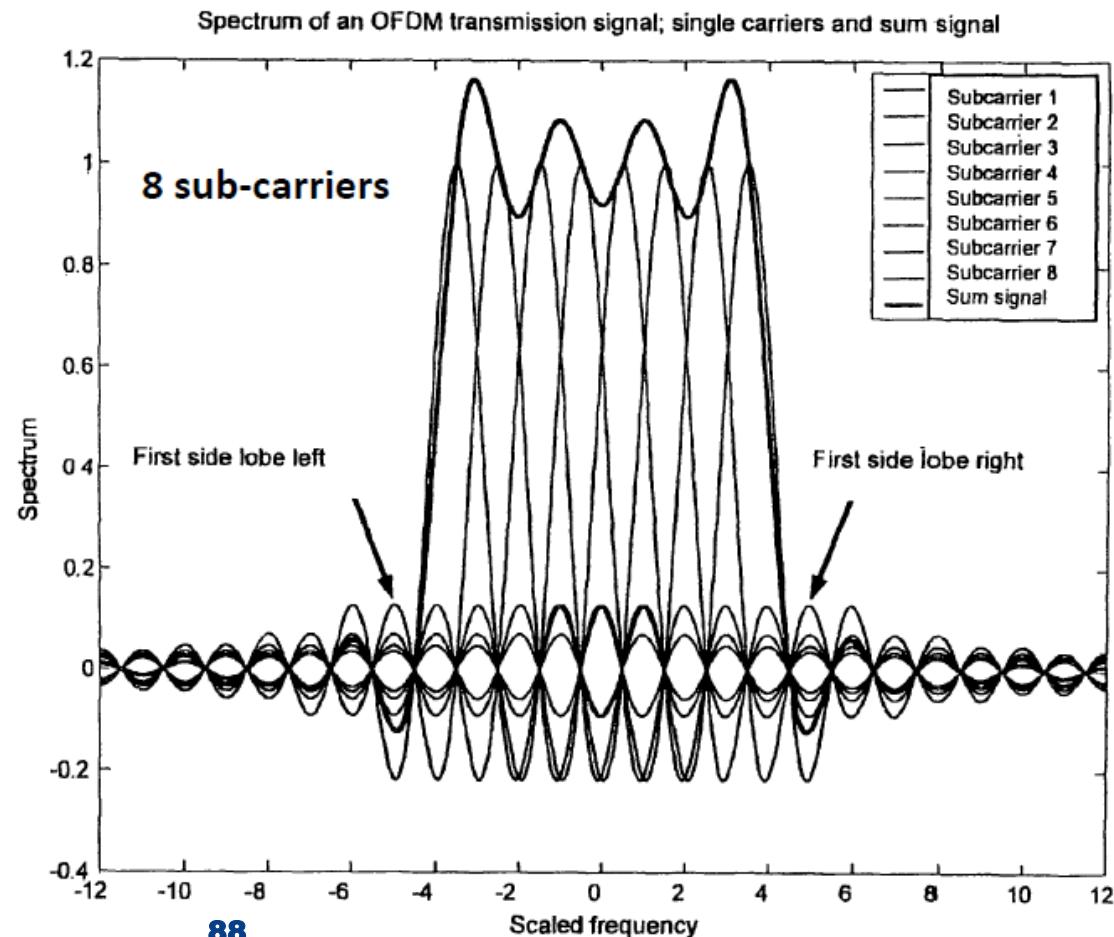
- * The spectral efficiency of **OFDM** is better than **FDM**

Need guard band between adjacent frequency bands → extra overhead and low throughput



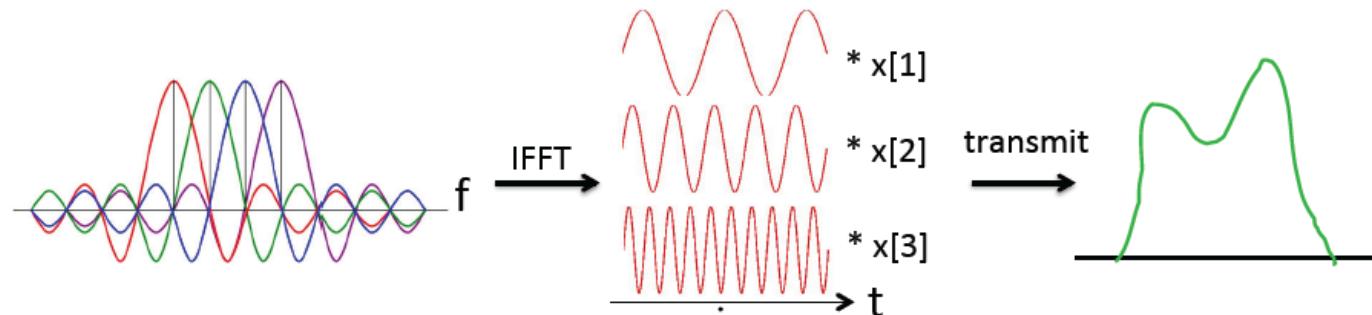
OFDM

* Spectrum of OFDM Signal Transmission



OFDM Transmitter/Receiver

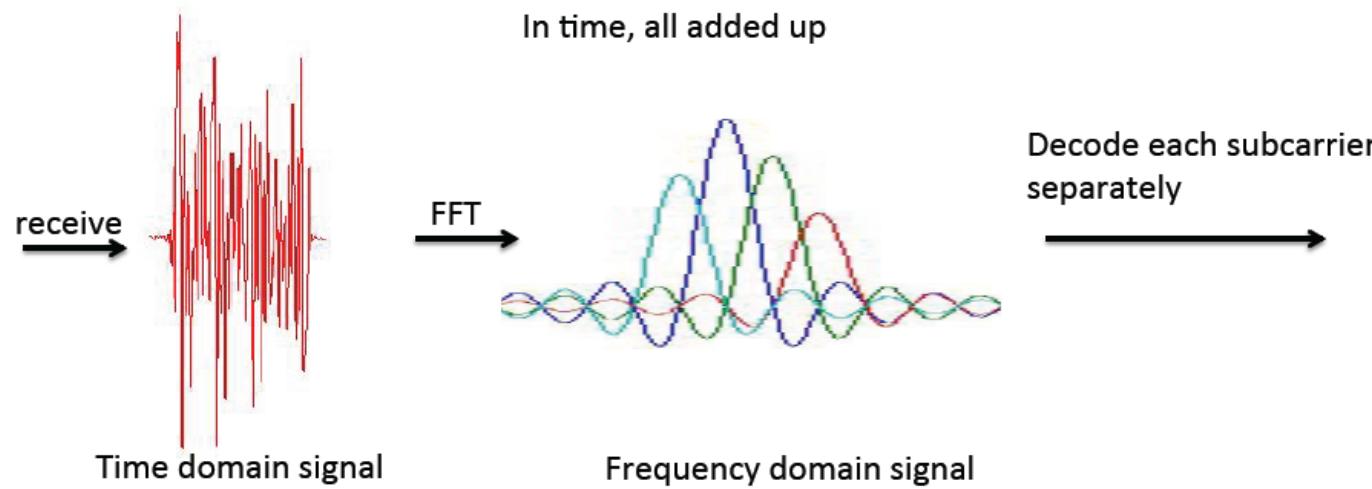
* Orthogonal Frequency Division Multiplexing



Data coded in frequency domain

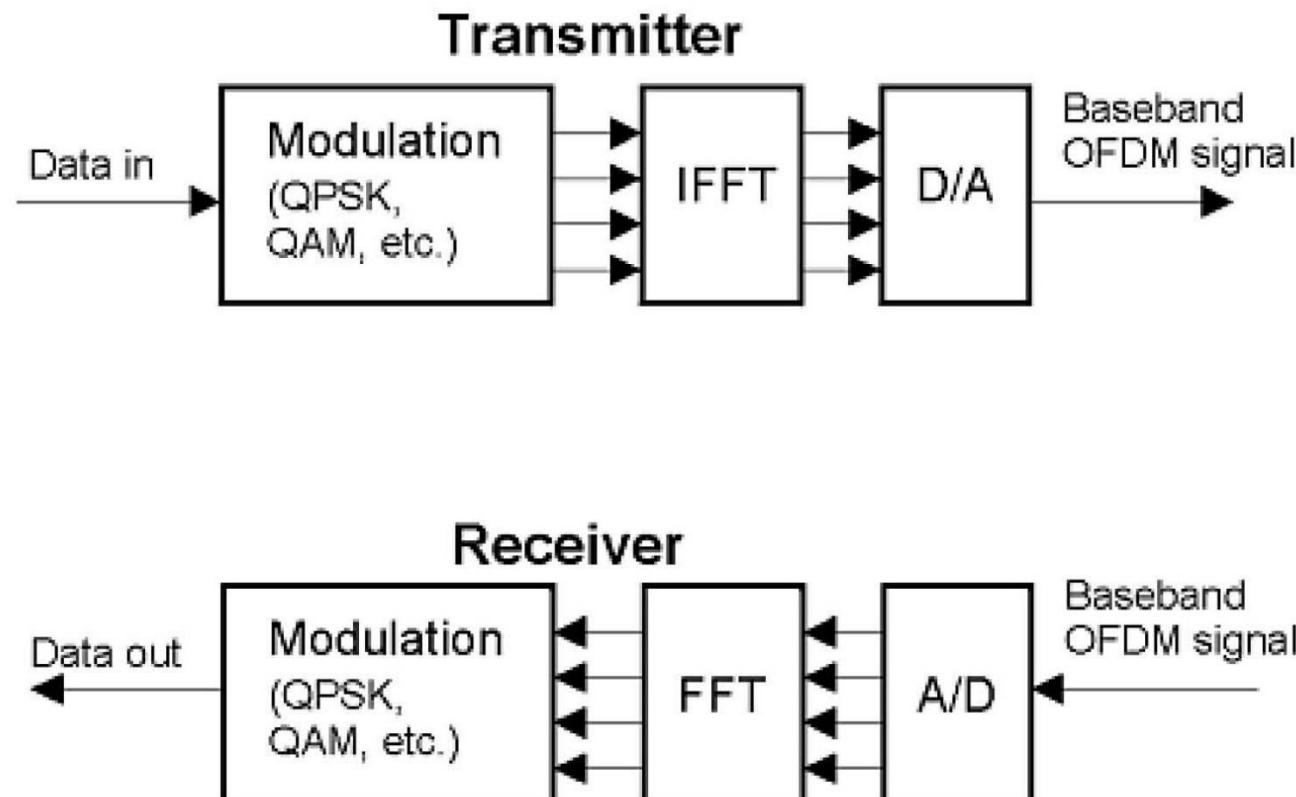
Transformation to time domain:
each frequency is a sine wave
In time, all added up

Channel frequency
response



OFDM Transmitter/Receiver

- * OFDM Transmitter and Receiver



Orthogonality of Sub-carriers

*

Encode: frequency-domain samples → time-domain sample IFFT

$$x(t) = \sum_{k=-N/2}^{N/2-1} X[k] e^{j2\pi kt/N}$$
$$X[k] = \frac{1}{N} \sum_{t=N/2}^{N/2-1} x(t) e^{-j2\pi kt/N}$$

FFT

*

Decode: time-domain samples → frequency-domain sample

Orthogonality of any two bins :

$$\sum_{t=N/2}^{N/2-1} e^{-j2\pi kt/N} e^{-j2\pi pt/N} = 0, \forall p \neq k$$

OFDM Example

- * Say we use BPSK and 4 sub-carriers to transmit a stream of samples

1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1, ...

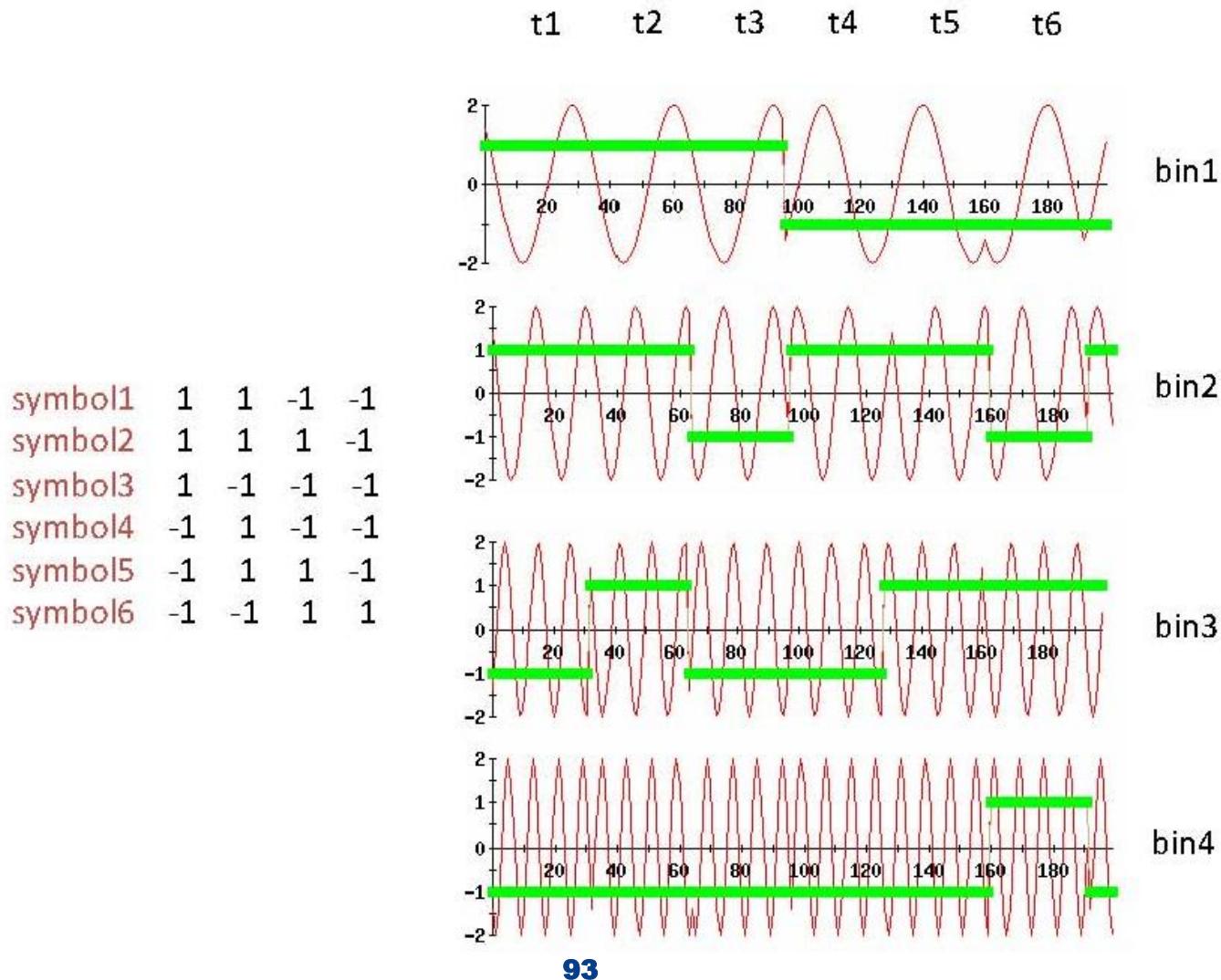
- * Serial to parallel conversion of samples

	Frequency-domain signal				Time-domain signal			
	c1	c2	c3	c4	IFFT			
symbol1	1	1	-1	-1	0	2 - 2i	0	2 + 2i
symbol2	1	1	1	-1	2	0 - 2i	2	0 + 2i
symbol3	1	-1	-1	-1	-2	2	2	2
symbol4	-1	1	-1	-1	-2	0 - 2i	-2	0 + 2i
symbol5	-1	1	1	-1	0	-2 - 2i	0	-2 + 2i
symbol6	-1	-1	1	1	0	-2 + 2i	0	-2 - 2i

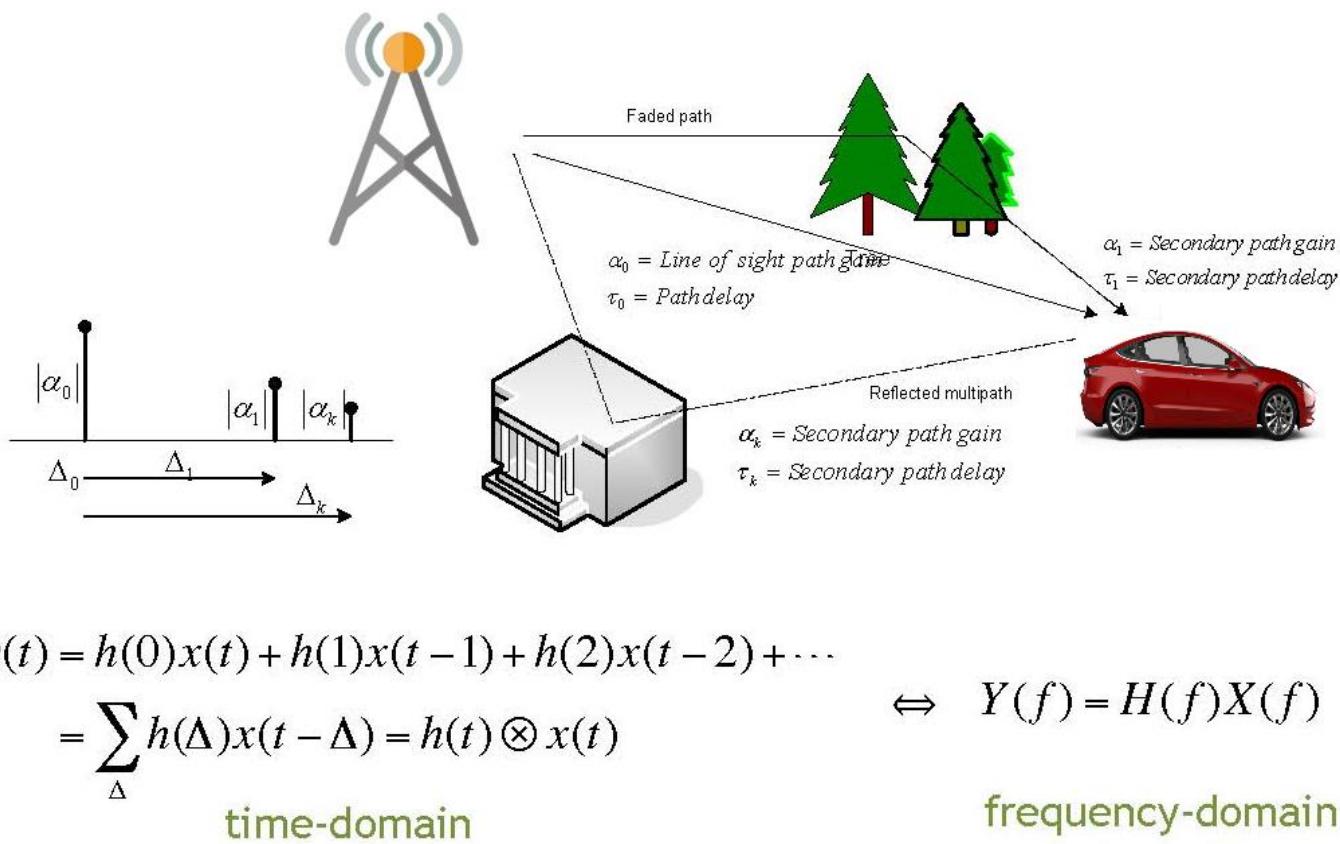
- * Parallel to serial conversion, and transmit time-domain samples

0, 2 - 2i, 0, 2 + 2i, 2, 0 - 2i, 2, 0 + 2i, -2, 2, 2, 2, -2, 0 - 2i, -2, 0 + 2i, 0, -2 - 2i, 0, -2 + 2i, 0, -2 + 2i, 0, -2 - 2i, ...

OFDM Example

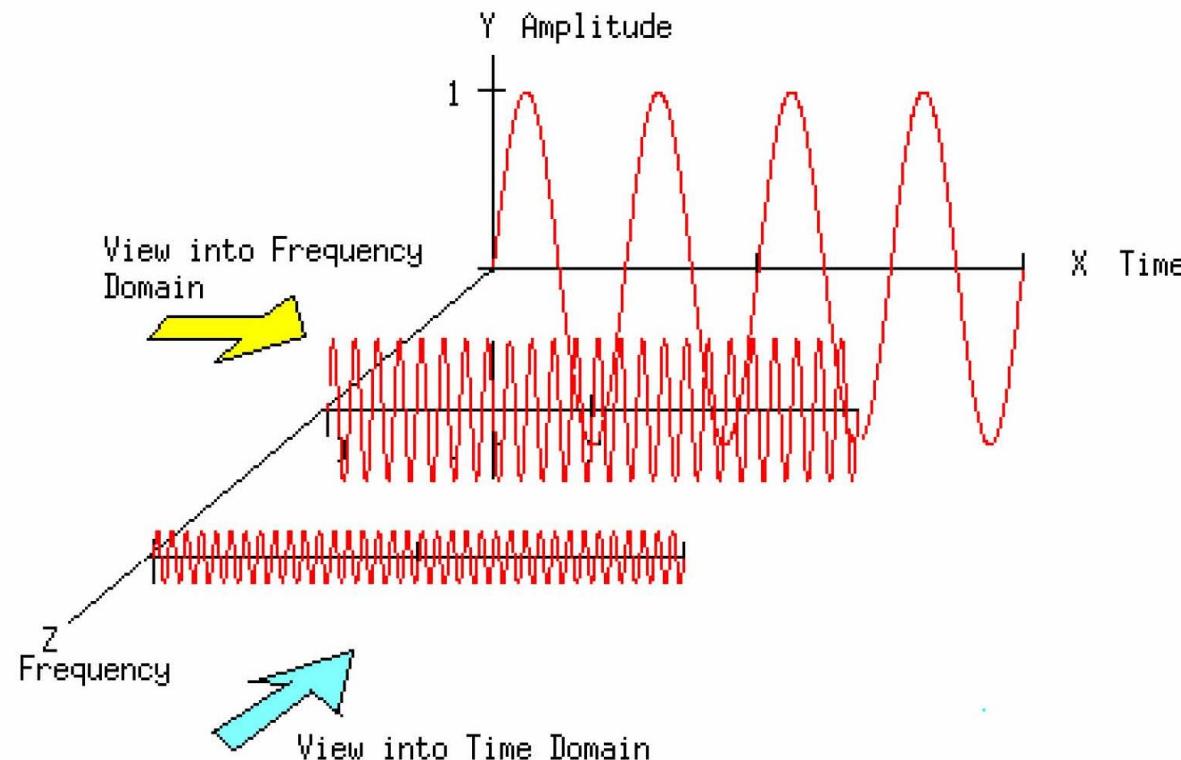


Multi-Path Effect



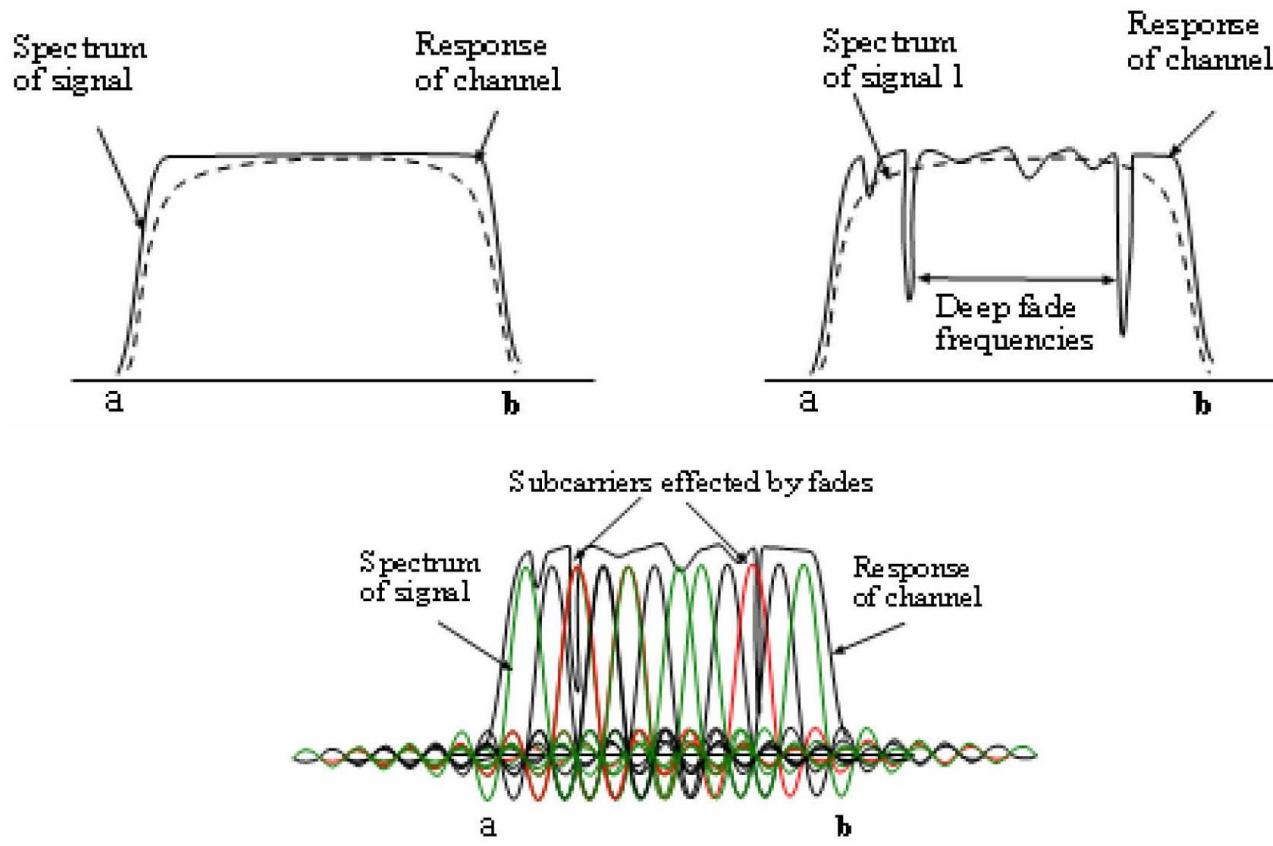
OFDM Example

- * Current symbol + delayed-version symbol
- Signals are deconstructive in only certain frequencies



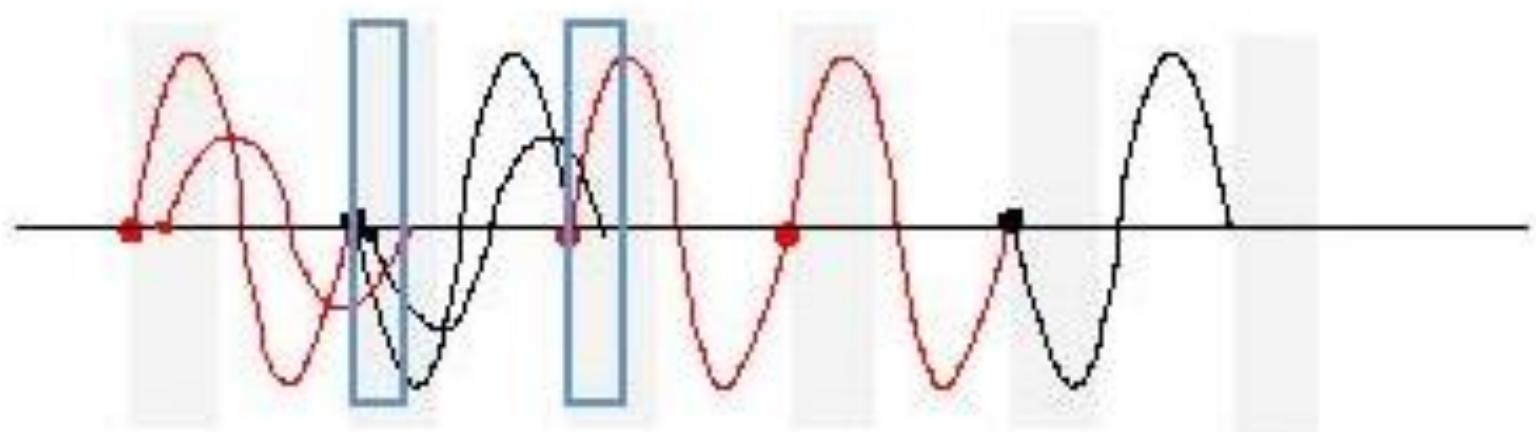
Frequency Selective Fading

- * Frequency selective fading: Only some sub-carriers get affected



Inter-symbol Interference

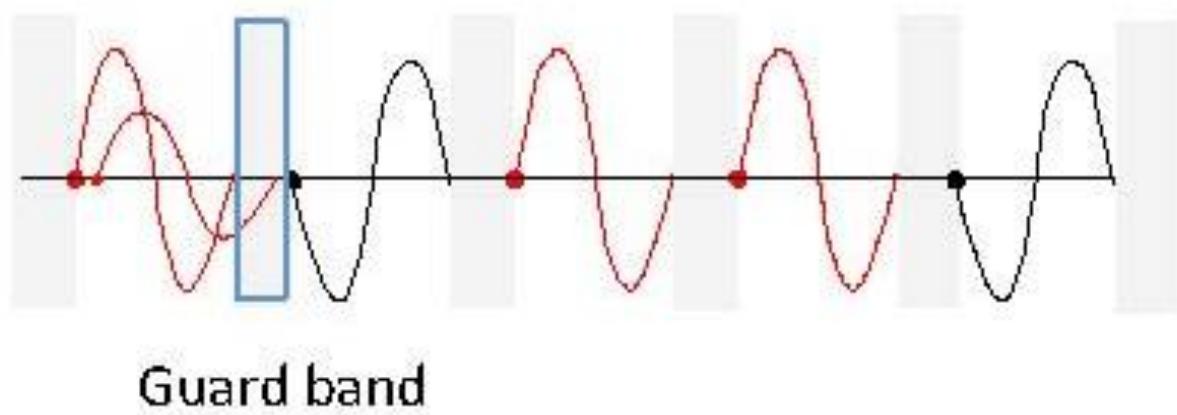
- * The delayed version of a symbol overlaps with the adjacent symbol
 - ◆ 傳送訊號在通過具有多重路徑干擾的通道後，會造成前一個符元的後端部份干擾到下一個符元的前端，此稱之為「符元間的干擾（ISI）」



Inter-symbol Interference

- * One simple solution to avoid this is to introduce a guard-band (guard-interval)

為了克服ISI的問題，在OFDM symbol前端加入一保護區間（Guard Interval）。為了對抗訊號因通道延遲的影響



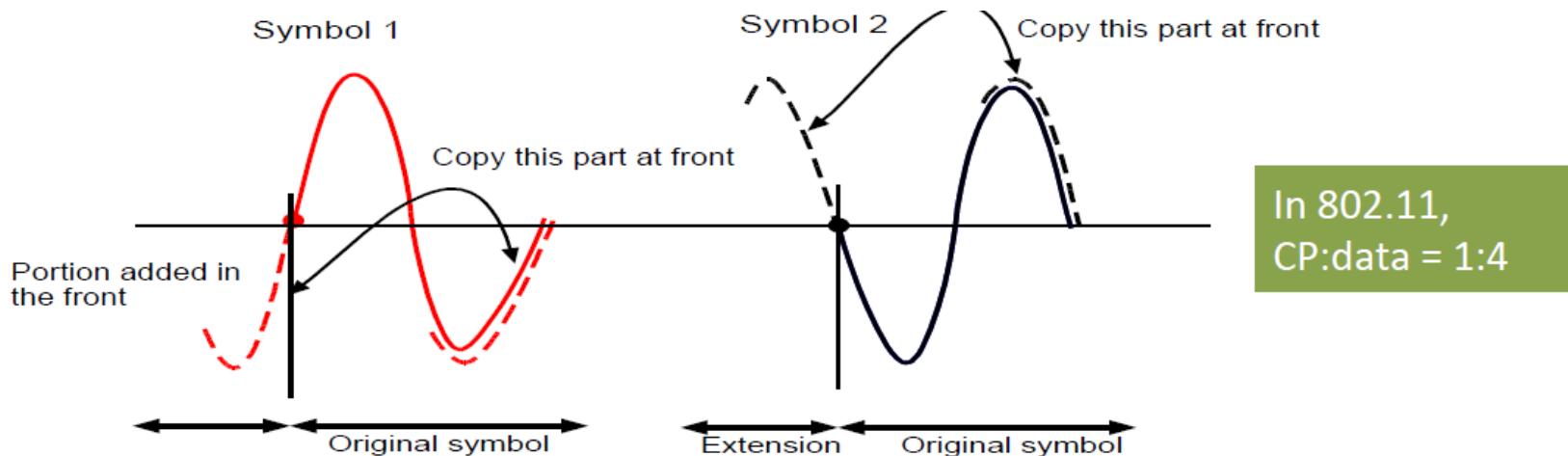
- * However, we don't know the delay spread exactly
 - ◆ The hardware doesn't allow blank space because it needs to send out signals continuously

Solution: Cyclic Prefix

- * Solution: Cyclic Prefix

- ◆ Make the symbol period longer by copying the tail and glue it in the front

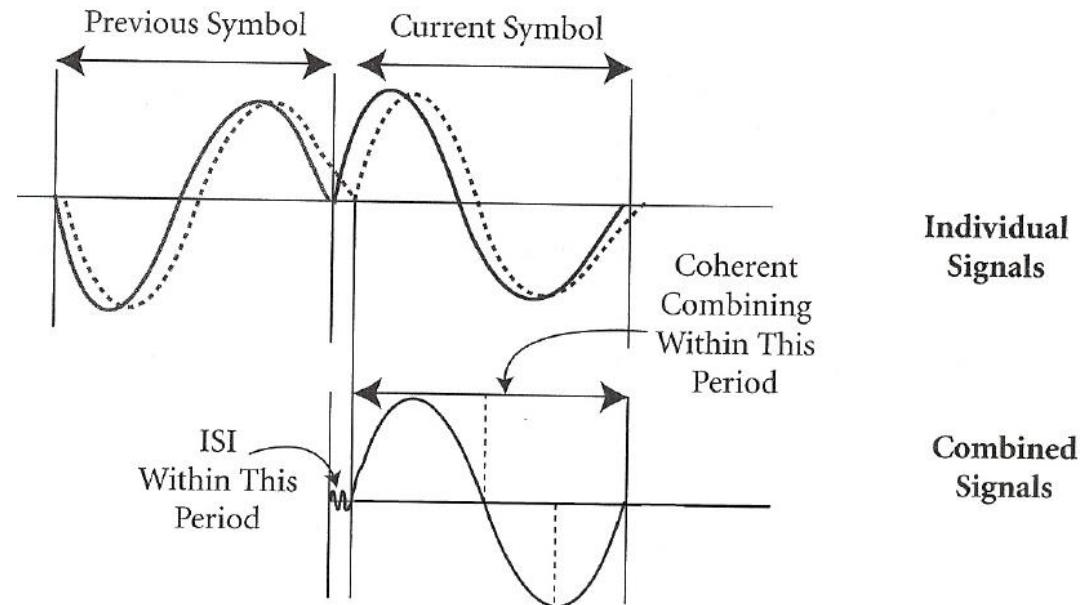
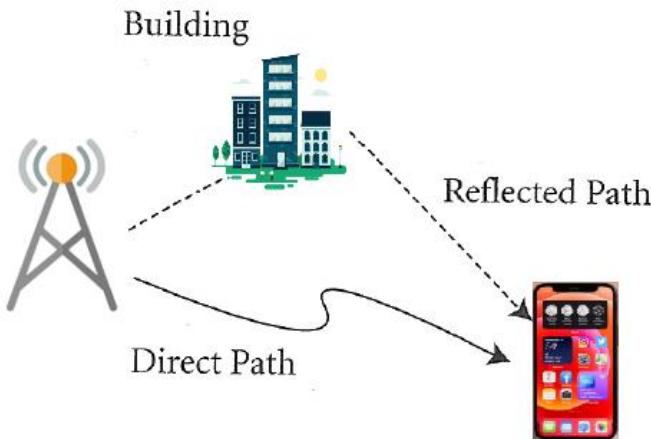
複製OFDM symbol後半段訊號並擺放於保護區間內，稱之為循環字首（cyclic prefix）；循環字首會造成頻寬效益下降，故必須小於OFDM symbol長度的1/4。如：一個OFDM symbol共有256個子載波，則其循環字長度為64個位元。



Solution: Cyclic Prefix

- * Solution: Cyclic Prefix

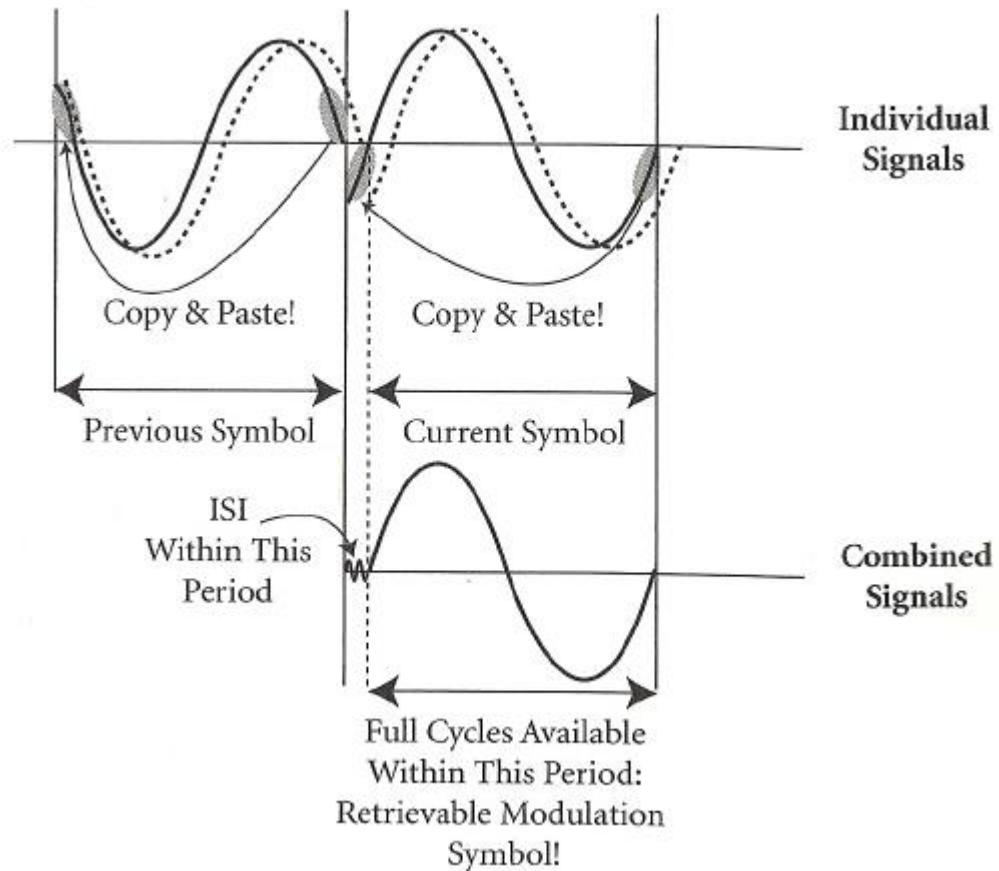
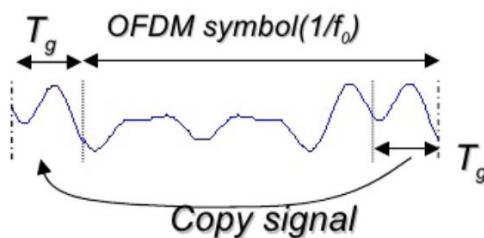
- The figure shows two copies of the signal (direct path and reflected path with time difference).



Solution: Cyclic Prefix

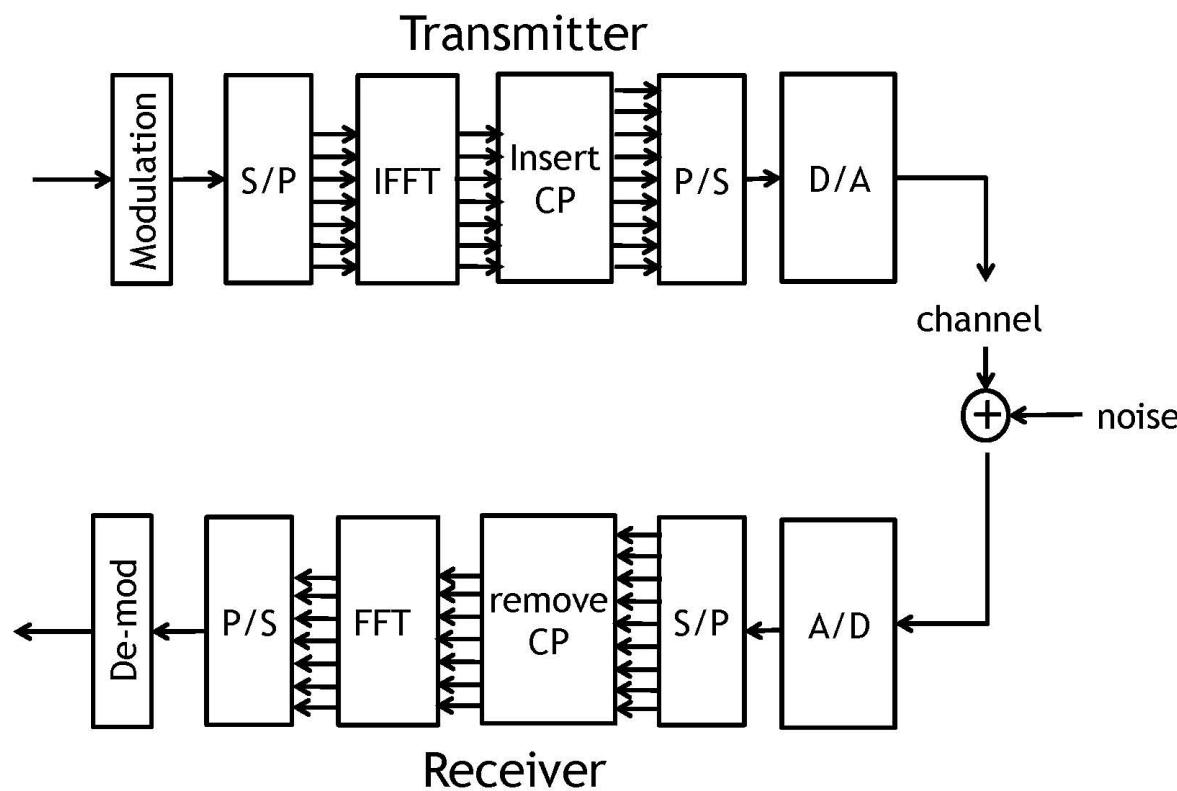
* Solution: Cyclic Prefix

- For the purpose to eliminate the effect of ISI, the guard interval could consist of no signals at all
- Adding the Guard Interval Period, ISI can be avoided.



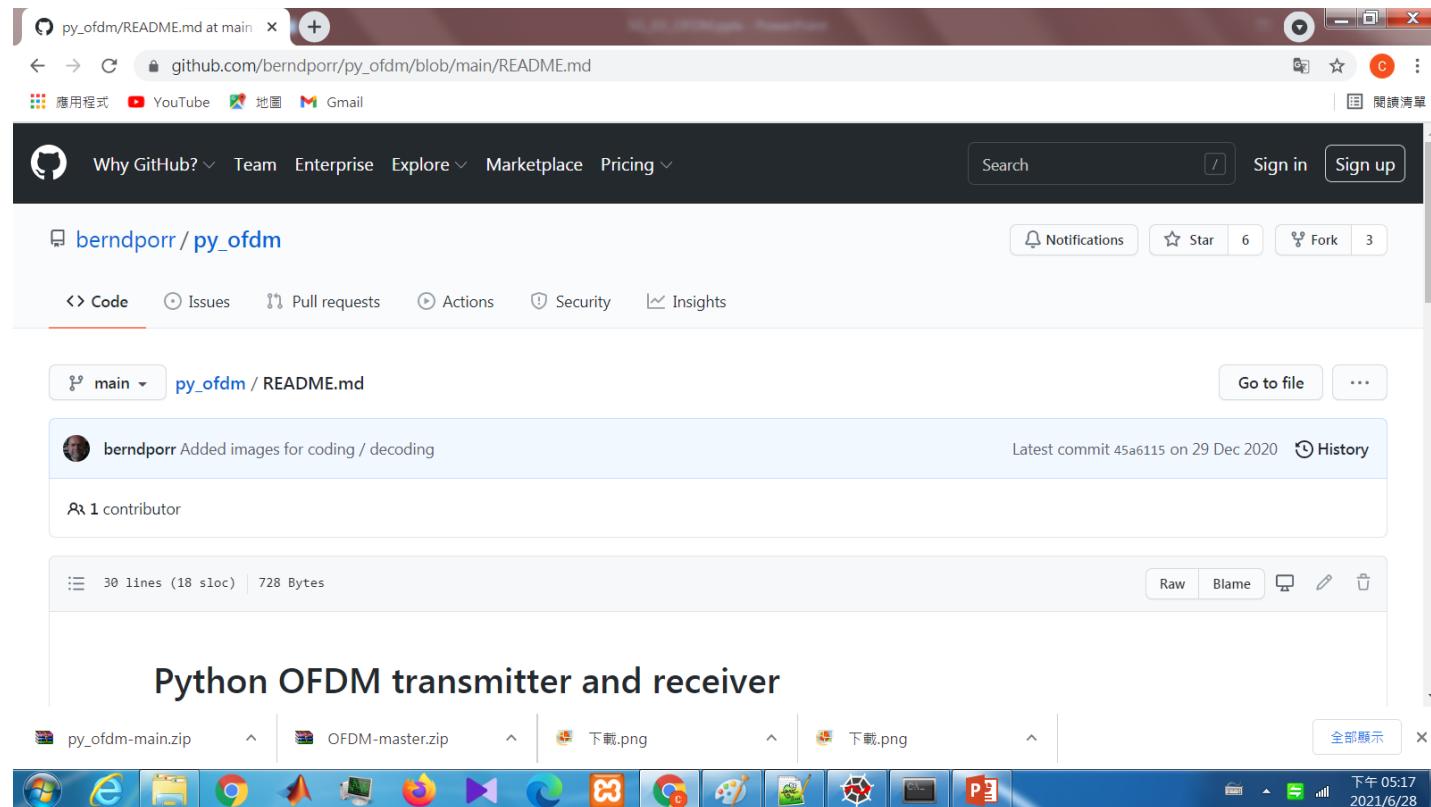
OFDM Diagram

* OFDM with **cyclic prefix**



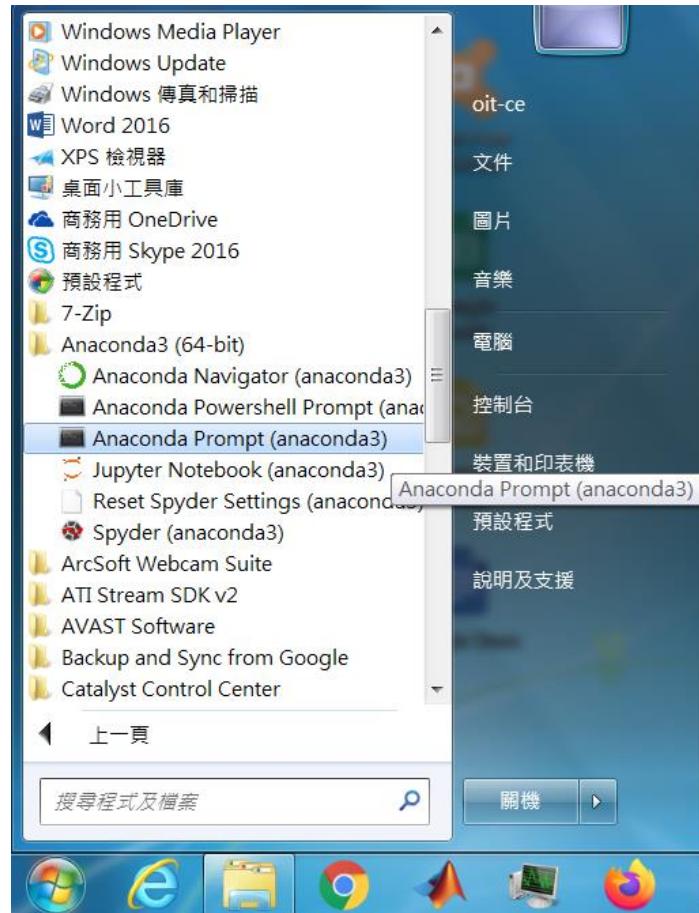
Sample Code for OFDM

* https://github.com/berndporr/py_ofdm



Sample Code for OFDM

* 進入 Anaconda Prompt(anaconda3)



Sample Code for OFDM

- * conda update anaconda-navigator
- * conda update navigator-updater

```
conda update anaconda-navigator
(base) C:\Users\oit-ce>conda update anaconda-navigator
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\oit-ce\anaconda3

added / updated specs:
- anaconda-navigator

The following packages will be downloaded:

  package          | build
  -----|-----
  anaconda-navigator-2.0.3 | py38_0      5.2 MB
  backports.functools_lru_cache-1.6.4 | pyhd3eb1b0_0   9 KB
  backports.tempfile-1.0 | pyhd3eb1b0_1   11 KB
  conda-4.10.1 | py38haa95532_1   2.9 MB
  conda-content-trust-0.1.1 | pyhd3eb1b0_0   56 KB
  conda-package-handling-1.7.3 | py38h8cc25b3_1  721 KB
  conda-repo-cl i-1.0.4 | pyhd3eb1b0_0   47 KB
  conda-token-0.3.0 | pyhd3eb1b0_0   10 KB
```

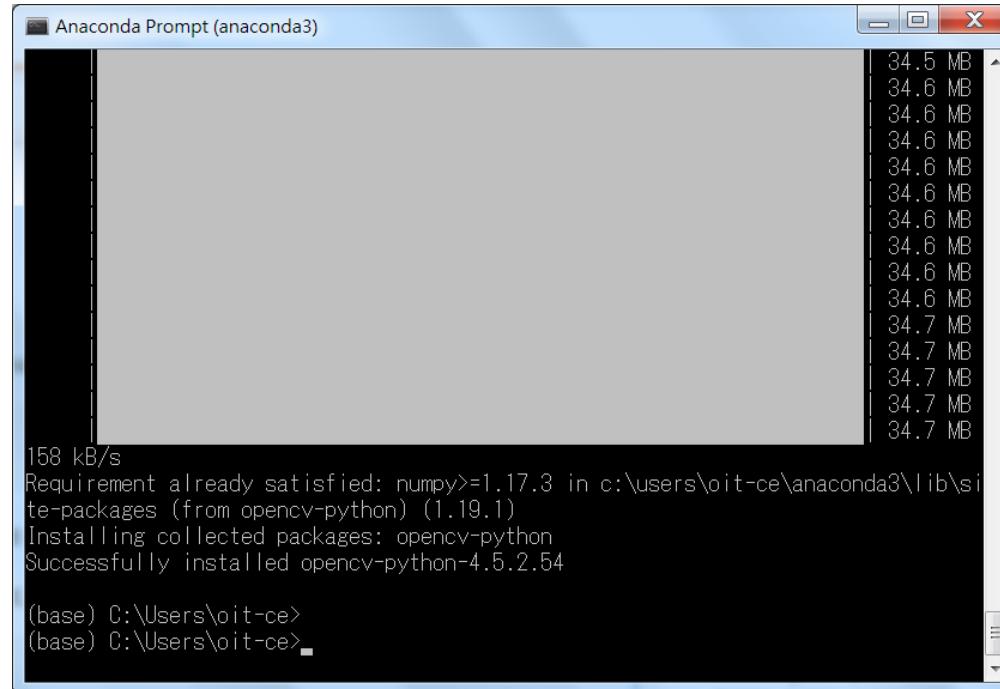
```
Anaconda Prompt (anaconda3)
(base) C:\Users\oit-ce>conda update navigator-updater
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) C:\Users\oit-ce>
```

Sample Code for OFDM

- * pip install opencv-python

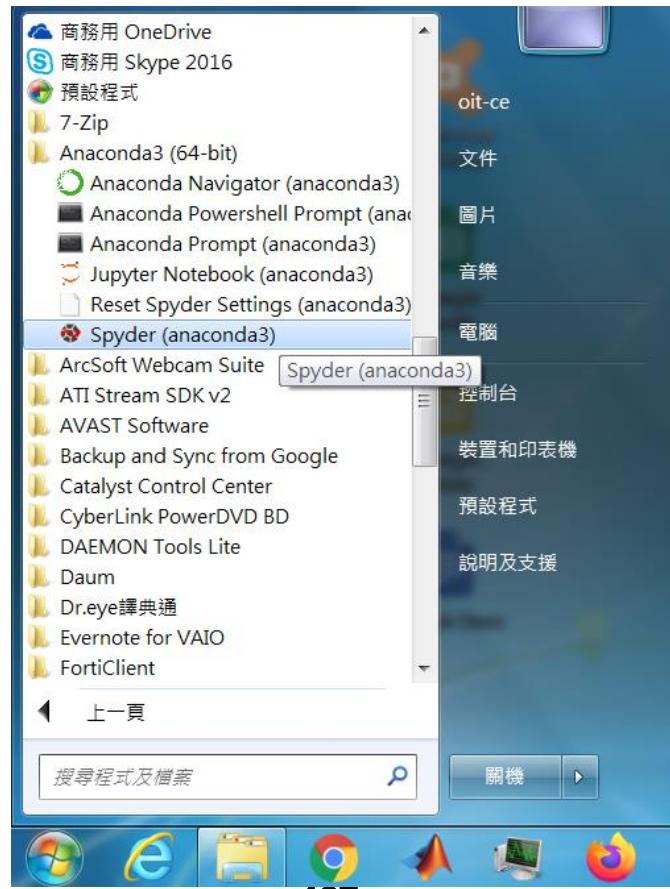


```
Anaconda Prompt (anaconda3)
158 kB/s
Requirement already satisfied: numpy>=1.17.3 in c:\users\oit-ce\anaconda3\lib\site-packages (from opencv-python) (1.19.1)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.2.54

(base) C:\Users\oit-ce>
(base) C:\Users\oit-ce>
```

Sample Code for OFDM

- * 進入Anaconda 的 Spyder 整合環境



Sample Code for OFDM



- * 開啟 C:\py_ofdm-main\ofdm_tx.py

The screenshot shows the Spyder Python 3.8 IDE interface. The main window displays the code for 'ofdm_tx.py'. The code includes a detailed copyright notice and imports from cv2, numpy, scipy.io.wavfile, matplotlib.pyplot, and ofdm_codec. It then loads a grayscale image named 'greystee.png' and prints its dimensions. A 'Usage' help dialog is open, providing instructions on how to get help for objects. The bottom right corner of the screen shows a taskbar with various application icons.

```
# Copyright (C) 2020 Bernd Porr <mail@berndporr.me.uk>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU Lesser General Public License as published
# by the Free Software Foundation; either version 2 of the license, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
# Transmitter and test receiver on the same data
# Transmits a test grey scale image. Every horizontal line turns into a syn
#
import cv2
import numpy as np
import scipy.io.wavfile as wavfile
import matplotlib.pyplot as plt
import ofdm_codec
#
# load the cheesy image
a = cv2.imread('greystee.png',cv2.IMREAD_GRAYSCALE)
#
# the number of lines in the image
ymax,xmax= np.shape(a)
print(xmax,ymax)
```

Sample Code for OFDM

* 執行



Spyder (Python 3.8)

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\oit-ce\Downloads\py_ofdm-main\py_ofdm-main\ofdm_tx.py
polarcodes.py x simple_simulation.py x untitled0.py x ofdm_tx.py
47     # save it as a wav file to listen to
48     wavfile.write('ofdm8000.wav',8000,signal)
49
50     plt.subplot(131)
51     plt.title("OFDM complex spectrum")
52     plt.plot(np.abs(ofdm.spectrum))
53
54     plt.subplot(132)
55     plt.title("TX OFDM spectrum")
56     plt.xlabel("Normalised frequencies")
57     plt.ylabel("Freq-amplitudes")
58     plt.plot(np.linspace(0,1,len(signal)),np.abs(np.fft.fft(signal))/len(signal))
59
60 ##### reception
61
62 # our image
63 rx_image = np.empty((ymax,xmax))
64
65 # Let's do a sanity check. We cheat here as we know the index of the symbol
66 ofdm.initDecode(signal,offset)
67
68 # loop for the y coordinate
69 for y in range(ymax):
70     row,1 = ofdm.decode()
71     rx_image[y,:] = row
72
73
74 plt.subplot(133)
75 plt.imshow(rx_image, cmap='gray')
76 plt.show()
```

OFDM complex spectrum TX OFDM spectrum

Figure showing the OFDM complex spectrum and TX OFDM spectrum. The TX spectrum shows a flat-top windowed sinusoidal signal. The RX spectrum shows the received signal with noise. An inset image shows a person's face.

Console 1/A

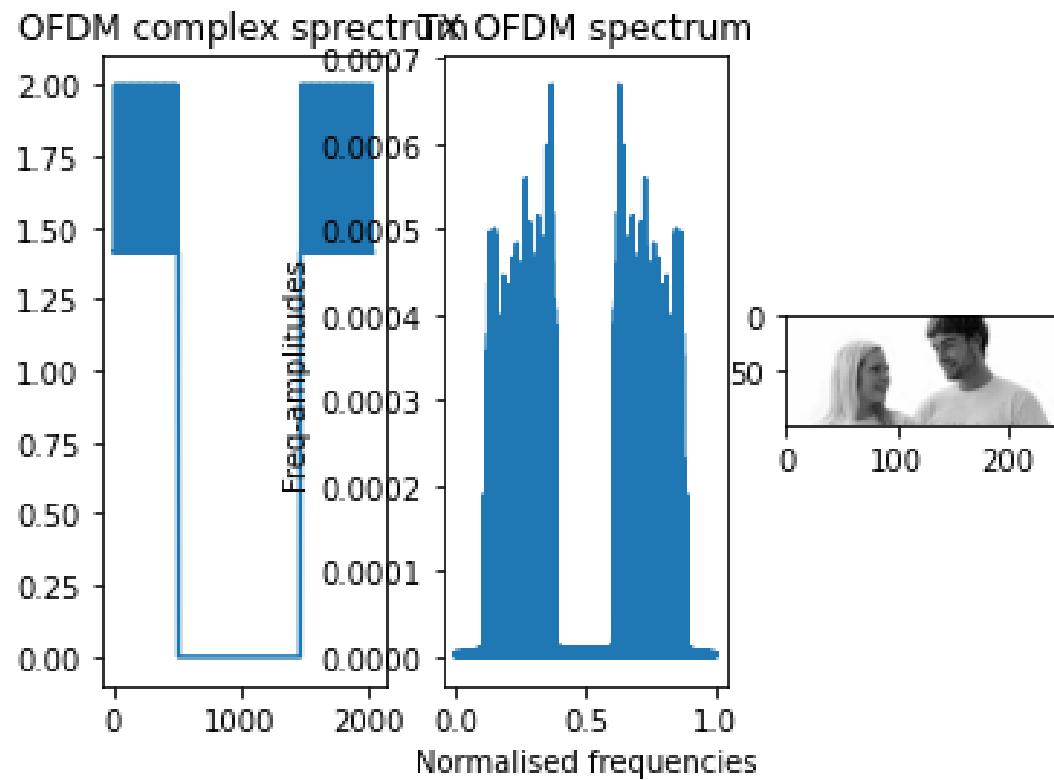
```
ModuleNotFoundError: No module named 'cv2'
In [2]: runfile('C:/Users/oit-ce/Downloads/py_ofdm-main/py_ofdm-main/ofdm_tx.py', wdir='C:/Users/oit-ce/Downloads/py_ofdm-main/py_ofdm-main')
256 100
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

LSP Python: ready conda: base (Python 3.8.5) Line 1, Col 1 ASCII LF RW Mem 76%

Sample Code for OFDM

* 執行



Sample Code for OFDM



- * 開啟 C:\py_ofdm-main\ofdm_rx.py

The screenshot shows the Spyder Python 3.8 IDE interface. The left pane displays the source code for `ofdm_rx.py`, which includes a copyright notice and imports for cv2, numpy, wavfile, plt, mcolors, and ofdm_codec. The right pane shows the IPython console output:

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]:
```

Sample Code for OFDM

* 執行



Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\oit-ce\Downloads\py_ofdm-main\py_ofdm-main\ofdm_rx.py

ofdm_tx.py x ofdm_rx.py x

```
1 # Copyright (C) 2020 Bernd Porr <mail@berndporr.me.uk>
2
3 # This program is free software; you can redistribute it and/or modify
4 # it under the terms of the GNU Lesser General Public License as published
5 # by the Free Software Foundation; either version 2 of the License, or
6 # (at your option) any later version.
7
8 # This program is distributed in the hope that it will be useful,
9 # but WITHOUT ANY WARRANTY; without even the implied warranty of
10 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
11 # GNU General Public License for more details.
12
13 # You should have received a copy of the GNU Lesser General Public License
14 # along with this program; if not, write to the Free Software
15 # Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
16
17 # Transmitter and test receiver on the same data
18 # Transmits a test grey scale image. Every horizontal line turns into a sym
19
20 import cv2
21 import numpy as np
22 import scipy.io.wavfile
23 import matplotlib.pyplot as plt
24 import matplotlib.colors as mcolors
25 import ofdm_codec
26
27 # geometry of the expected image
28 ymax = 100
29 xmax= 256
30
31 # let's instantiate the OFDM codec
32 ofdm = ofdm_codec.OFDM()
```

Cross correlation auto-correlation between the imaginary parts of the pilots

Decoded image

Sample index Relative sample index

Variable explorer Help Plots Files

In [1]: runfile('C:/Users/oit-ce/Downloads/py_ofdm-main/py_ofdm-main/ofdm_rx.py', wdir='C:/Users/oit-ce/Downloads/py_ofdm-main/py_ofdm-main')
fs = 8000
Symbol start sample index = 6144

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

LSP Python: ready conda: base (Python 3.8.5) Line 1, Col 1 ASCII LF RW Mem 77%
下午 05:23
2021/6/28

Sample Code for OFDM

* 執行

Cross correlation of the real and imaginary parts of the pilots

