



Digital Transmission

Paramate Horkaew

School of Computer Engineering
Institute of Engineering, Suranaree University of Technology



Lecture Outline

- Line Coding
- Block Coding
- Sampling
 - Pulse Amplitude Modulation
 - Pulse Code Modulation
 - Nyquist Theorem
- Transmission Mode
 - Parallel Transmission
 - Serial Transmission

Class 1

Class 2



The Overview

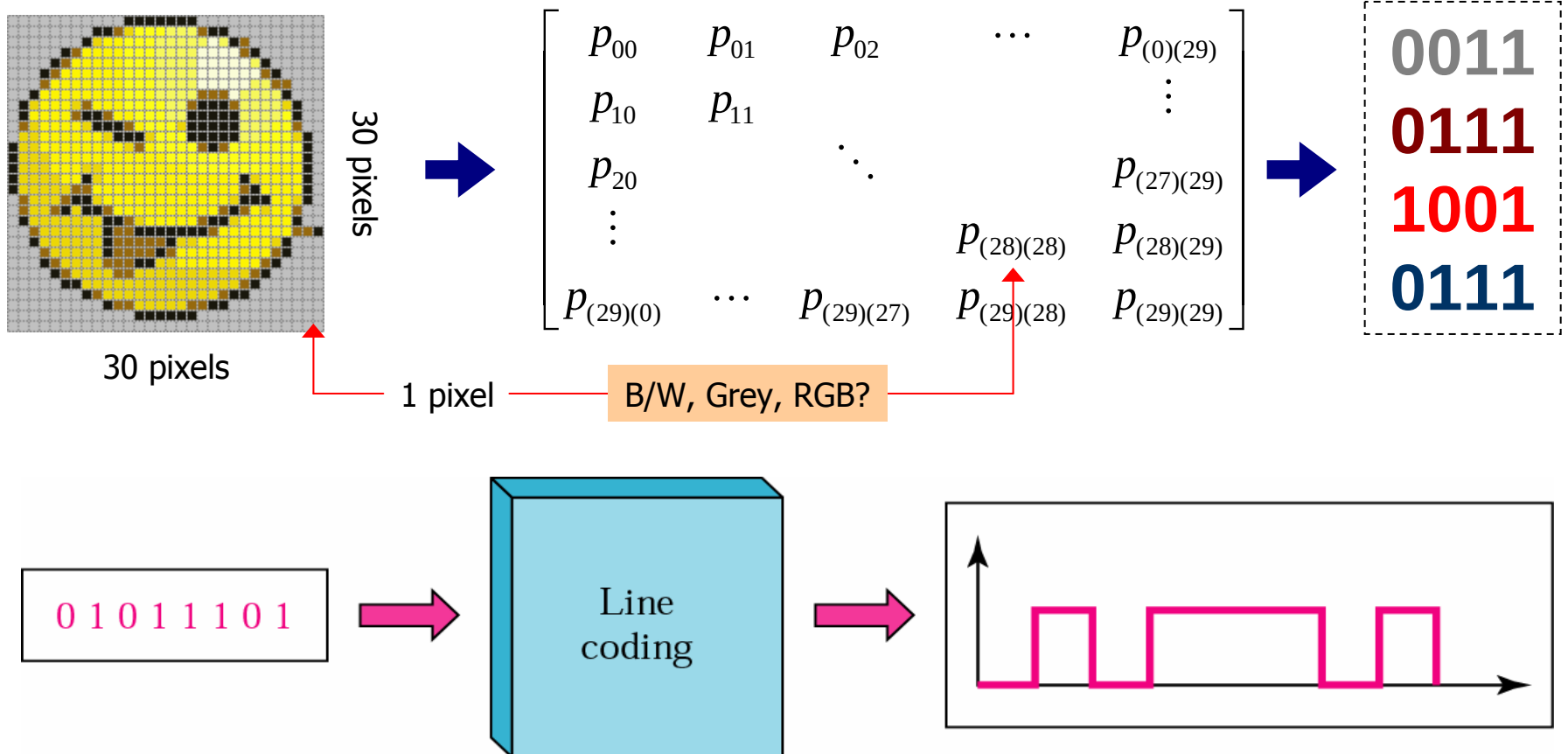
การส่งผ่านข้อมูล (Data Transmission)

การส่งผ่านข้อมูล-ข่าวสาร ระหว่างอุปกรณ์สองชุด ภายใน Network สามารถทำได้ 2 วิธี

- ส่งข่าวสารผ่านตัวกลางไปในรูปของ **สัญญาณ Digital**
เหมาะสำหรับการส่งผ่านตัวกลางแบบ Low-Pass หรือตัวกลางที่มีการแบ่งปันช่องสัญญาณแบบ Temporal (Time)
- แปลงข่าวสารเป็น **สัญญาณ Analog** ก่อนส่งผ่านตัวกลาง
เหมาะสำหรับการส่งผ่านตัวกลางแบบ Band-Pass หรือตัวกลางที่มีการแบ่งปันช่องสัญญาณแบบ Frequency

Line Coding (I)

“Line Coding คือการแปลงข้อมูล Binary เป็นสัญญาณ Digital”





Line Coding (II)

Line Coding Characteristics

- Signal Level *versus* Data Level
- Pulse Rate *versus* Bit Rate
- DC Components
- Self Synchronization

Line Coding Schemes

- Unipolar, Polar
- Non-return to Zero, Return to Zero,
- Manchester, Differential Manchester, Bipolar, 2B1Q, MLT3



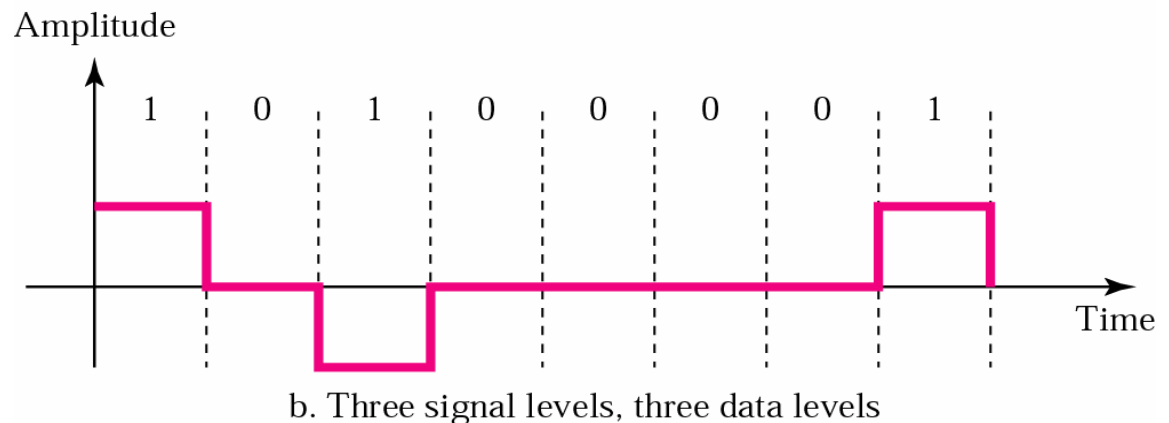
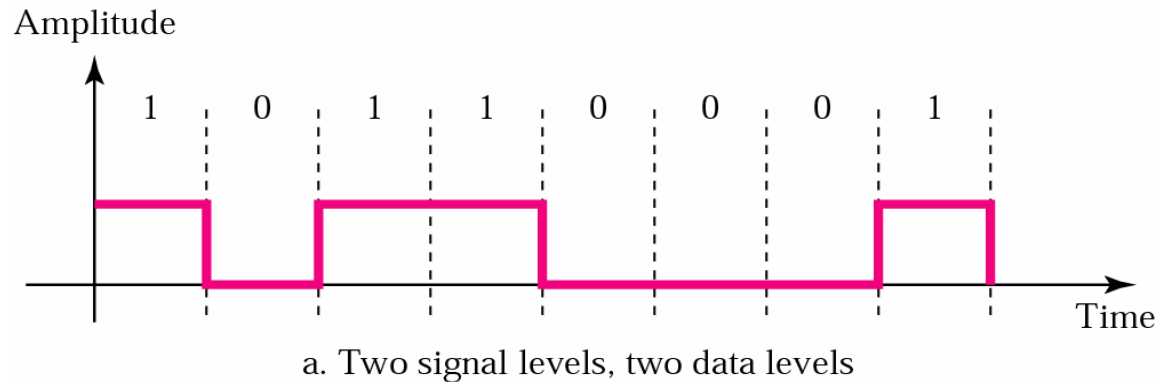
Signal vs. Data Levels

Digital Signal สัญญาณที่มี **จำนวนของค่า** (Levels) ที่เป็นไปได้จำกัด โดยที่บางค่าอาจใช้สำหรับจุดประสงค์อื่นด้วย (ไม่จำกัดเฉพาะนำเสนอข้อมูลเท่านั้น)

- เรียกจำนวนของค่าที่เป็นไปได้ทั้งหมดว่า **Signal Levels**
- เรียกจำนวนของค่าที่ใช้เฉพาะนำเสนอข้อมูลว่า **Data Levels**

Examples

- 2 Signal, 2 Data Lev.
- 3 Signal, 2 Data Lev.





Pulse vs. Bit Rate

Pulse คือ ช่วงหนึ่งของสัญญาณที่ใช้สำหรับส่งข้อมูล 1 สัญลักษณ์

Pulse Rate คือ จำนวนของ Pulse ที่สามารถส่งได้ใน 1 วินาที

Bit คือ หน่วยเล็กที่สุดของข้อมูล ย่อมาจาก Binary Digit

Bit Rate คือ จำนวนของ Bit ที่สามารถส่งได้ใน 1 วินาที

ถ้า 1 Pulse บรรจุข้อมูล N Bits จะได้ว่า $\text{Bit Rate} = \text{Pulse Rate} \times N$

เนื่องจาก N Bits สามารถใช้แทนระดับสัญญาณได้ $L = 2^N$ ระดับ

ดังนั้น Take $\log_2$ ทั้งสองข้าง จะได้ว่า $\log_2 L = N$

ซึ่งจะได้ความสัมพันธ์ $\text{Bit Rate} = \text{Pulse Rate} \times \log_2 L$



Examples

Example 1

สัญญาณหนึ่งประกอบด้วย Data Levels 2 ระดับ และมีช่วงเวลา 1 Pulse เท่ากับ 1 ms. จงคำนวณหา Pulse Rate และ Bit Rate ของสัญญาณนี้

$$\text{Pulse Rate} = 1 / 10^{-3} = 1000 \text{ pulses/s}$$

$$\text{Bit Rate} = \text{Pulse Rate} \times \log_2 L = 1000 \times \log_2 2 = 1000 \text{ bps}$$

Example 2 สัญญาณในข้อ 1 ประกอบด้วย Data Levels 2 ระดับ

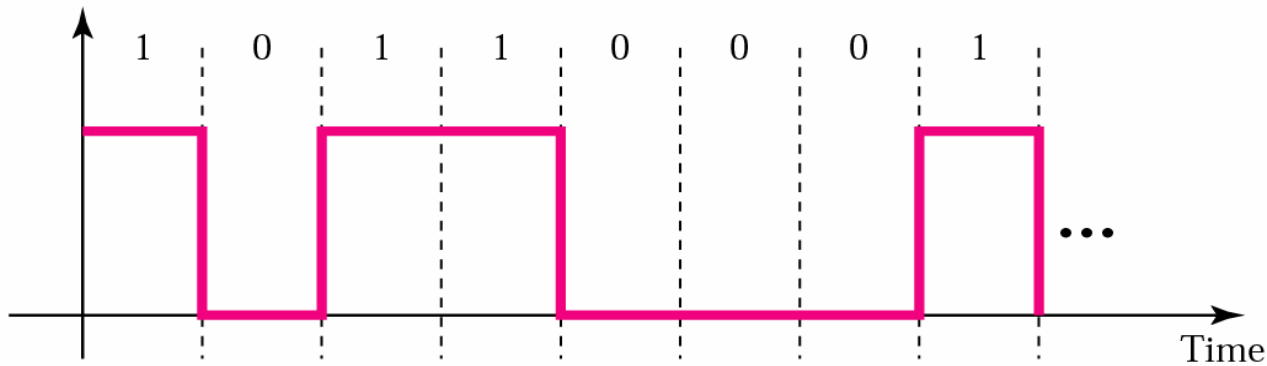
$$\text{Pulse Rate} = 1 / 10^{-3} = 1000 \text{ pulses/s}$$

$$\text{Bit Rate} = \text{Pulse Rate} \times \log_2 L = 1000 \times \log_2 4 = 2000 \text{ bps}$$



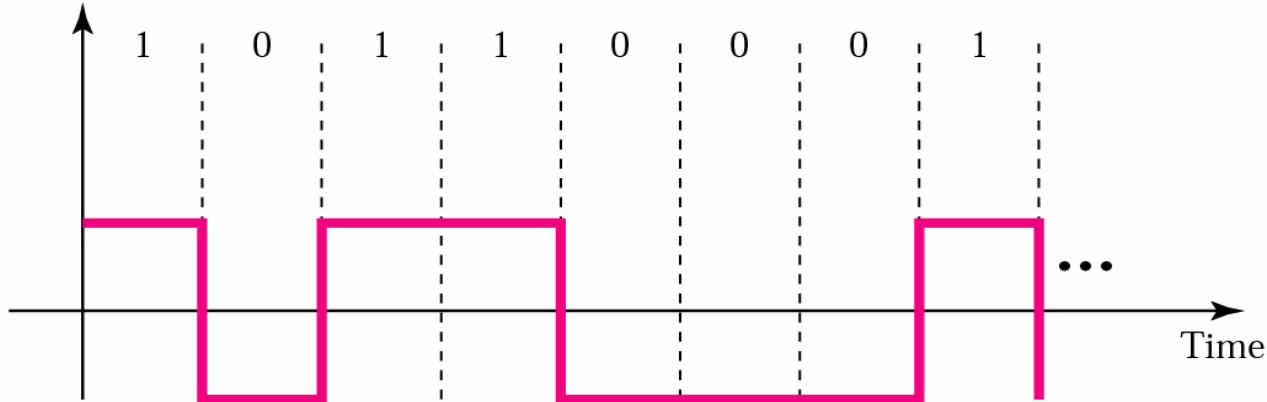
DC Components

Amplitude



a. A signal with dc component

Amplitude



b. A signal without dc component

คือ องค์ประกอบ
ของสัญญาณที่มี
ความถี่เป็น 0 (a.)

ข้อดี

- ทำให้ ออกแบบ
เครื่องรับ/ส่ง ง่าย

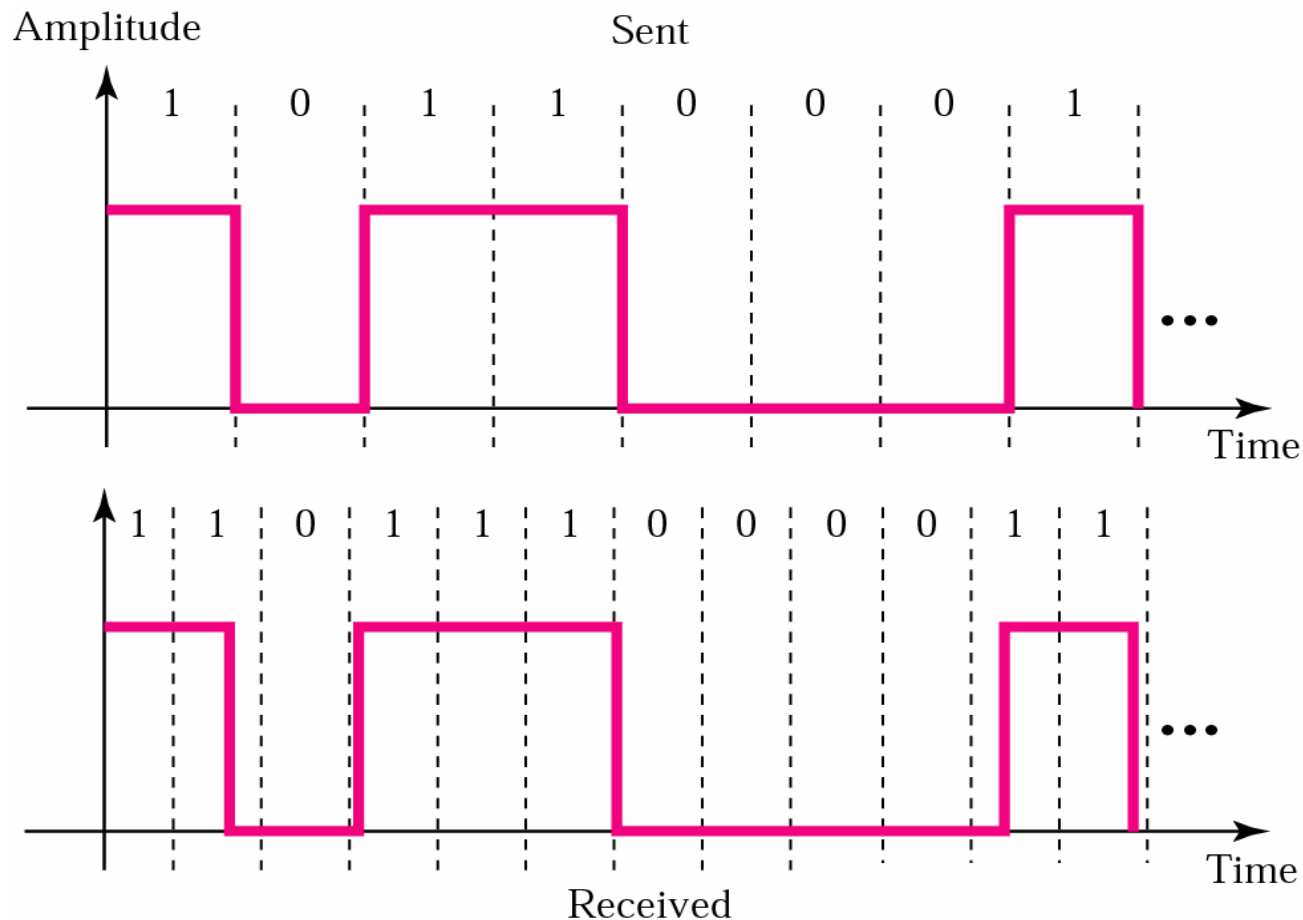
ข้อเสีย

- ส่งสัญญาณผ่าน
หม้อแปลงไม่ได้
- เกิดพลังงานที่
สูญเสียในสายส่ง



Self Synchronization

ถ้าความถี่ของสัญญาณนาฬิกา (CLK) ที่ด้านรับ/ส่งไม่เท่ากันจะเกิด Error ดังรูป



Self Synchronize

คือ การปรับสัญญาณนาฬิกาแบบอัตโนมัติในกรณีที่เกิด Error โดยอาศัยเทคนิคเพิ่มตัวบ่งชี้ในส่วนของ Pulse ที่ส่งออกไป

เมื่อ CLK เบี่ยงเบนด้านรับจะทำการปรับให้ตรงโดยอาศัยตัวบ่งชี้



Self Synchronization

Example 3

ในการส่งผ่านสัญญาณดิจิทัล ถ้าเครื่องรับมีสัญญาณนาฬิกาที่เร็วกว่าด้านส่ง 0.1% จงหาว่าด้านรับจะได้รับข้อมูลเกินมากี่บิตต่อวินาที ถ้าหากว่า ข้อมูลส่งมาในอัตราเร็ว 1Kbps และ 1Mbps ตามลำดับ

ที่ 1 Kbps:

1000 bits sent → 1001 bits received → 1 extra bps

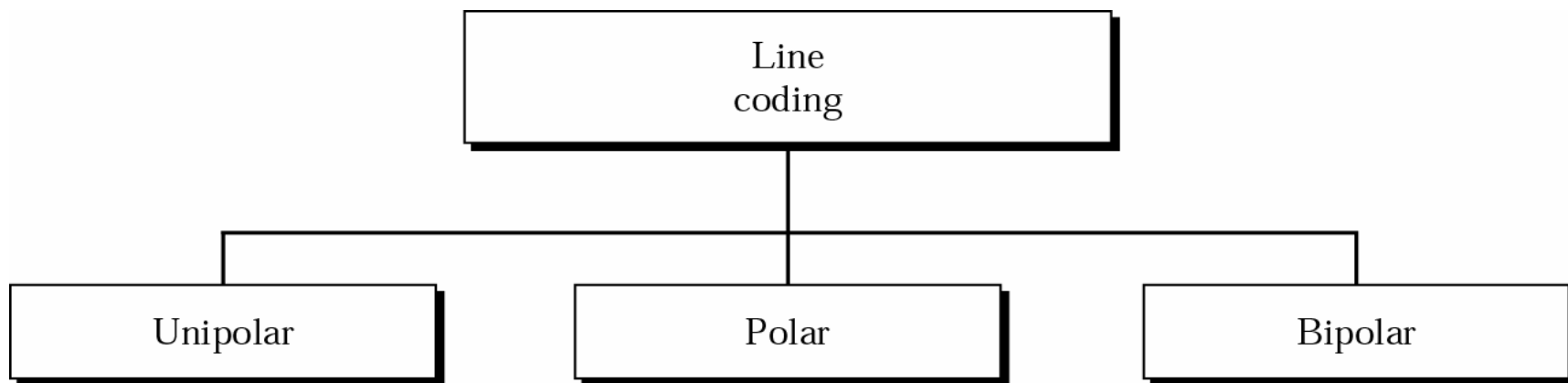
ที่ 1 Mbps:

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

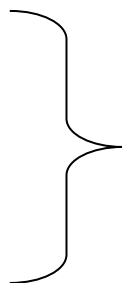
Bonus ถ้าเครื่องรับสุมเก็บข้อมูล ณ จุดกึ่งกลางของ Pulse จงหาว่า จะเกิด Error เมื่อใด ถ้าข้อมูลส่งมาในอัตราเร็ว 1 Kbps?

Line Coding Schemes

Line Coding แบ่งออกได้เป็น 3 วิธีตามลักษณะของระดับสัญญาณ ดังต่อไปนี้



- Unipolar
- Polar
- Bipoloar



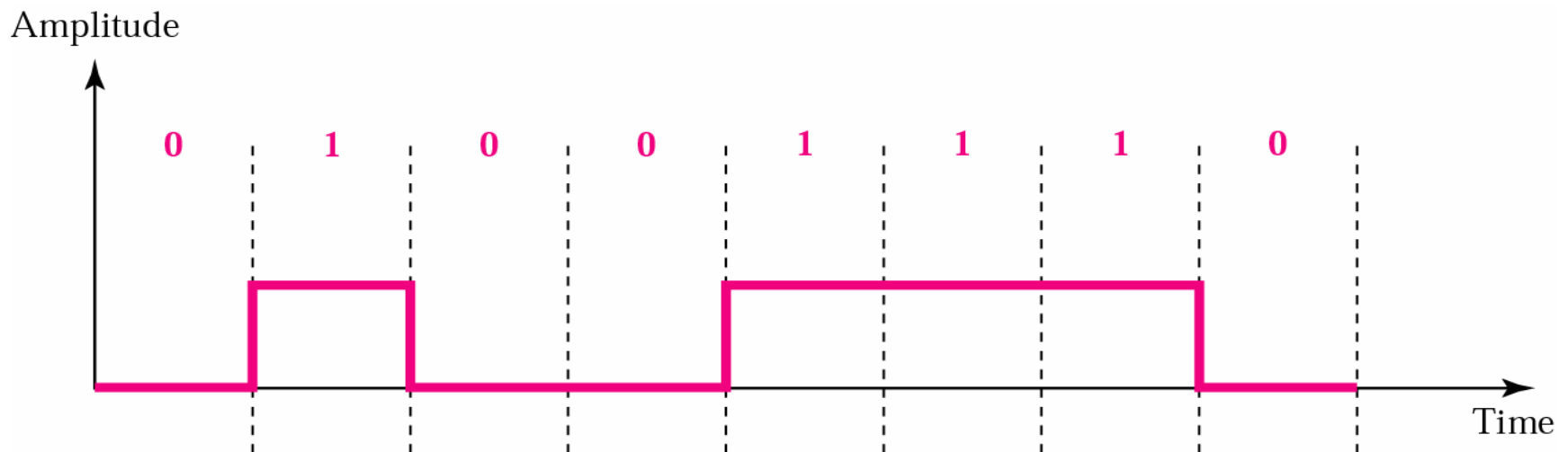
ข้อ ในที่นี้ยึดมาจากการเรียกตำแหน่งของจุดจ่ายพลังงานไฟฟ้า ซึ่งมีระดับที่แตกต่างกัน เมื่อเทียบกับจุดอ้างอิง



Unipolar

เป็นการส่งสัญญาณดิจิทัล **ที่ง่ายที่สุด** (ไม่พบการใช้ในการสื่อสารในปัจจุบัน
นอกจาก ระบบควบคุมอิเล็กทรอนิกส์อัตโนมัติ)

Polar หมายถึงความเป็น (+/−) ของระดับสัญญาณ **Unipolar** ใช้ระดับสัญญาณ
เพียงชั่วเดียว (+ หรือ −) โดย ระดับหนึ่งแทนเลข 1 และอีกระดับหนึ่งแทนเลข 0



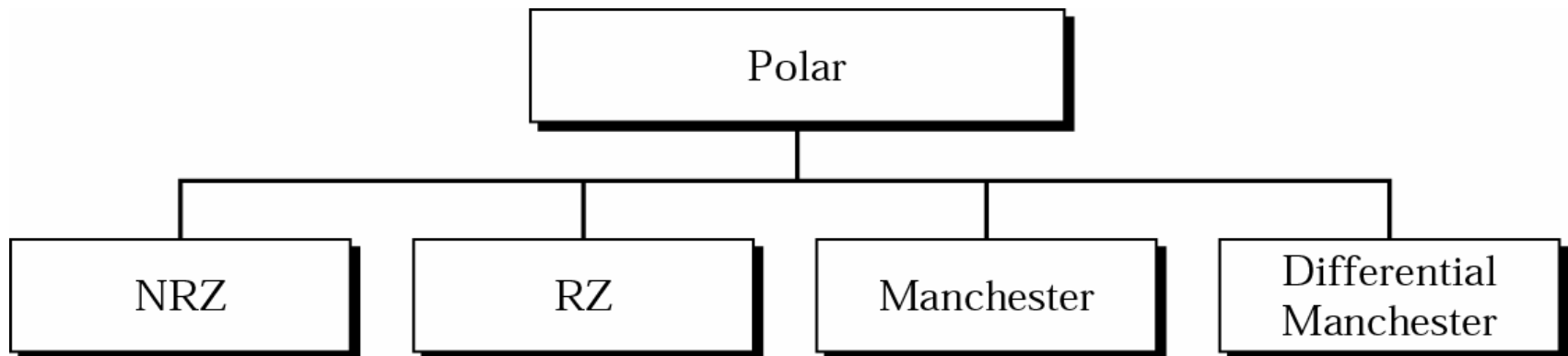
ข้อเสีย ไม่มี Self Synchronization และมี DC Component (ถ้าไม่ใช้ *Fourier*??)

Polar

Polar ใช้ระดับสัญญาณสองขั้ว (+ และ -) โดยขั้วหนึ่งแทนเลข 1 และอีกขั้วหนึ่งแทนเลข 0

ข้อมูลโดยปกติจะมีจำนวน 0 กับ 1 เท่าๆ กัน ดังนั้นระดับแรงดันเฉลี่ยของสัญญาณจึงมีค่าเข้าใกล้ 0 → มี DC Components น้อย (หรือไม่มีเลย)

การส่งสัญญาณแบบ Polar ที่นิยมใช้กันมี 4 วิธี ดังต่อไปนี้





Non-Return to Zero (NRZ)

การส่งสัญญาณที่ระดับสัญญาณเป็นไปได้อีก 2 กรณีเท่านั้นคือมีขั้วเป็น + หรือเป็น - แบ่งออกได้เป็น 2 รูปแบบ ได้แก่

- **NRZ-L (Level)**

สัญญาณขึ้นอยู่กับ **สถานะ** ของบิตเท่านั้น กล่าวคือ บิต = 1 จะมีระดับเป็น - และ บิต = 0 จะมีระดับเป็น +

ข้อเสีย ถ้าเกิดบิตมีค่าเดียวกันติดกัน จะเสีย Synchronization

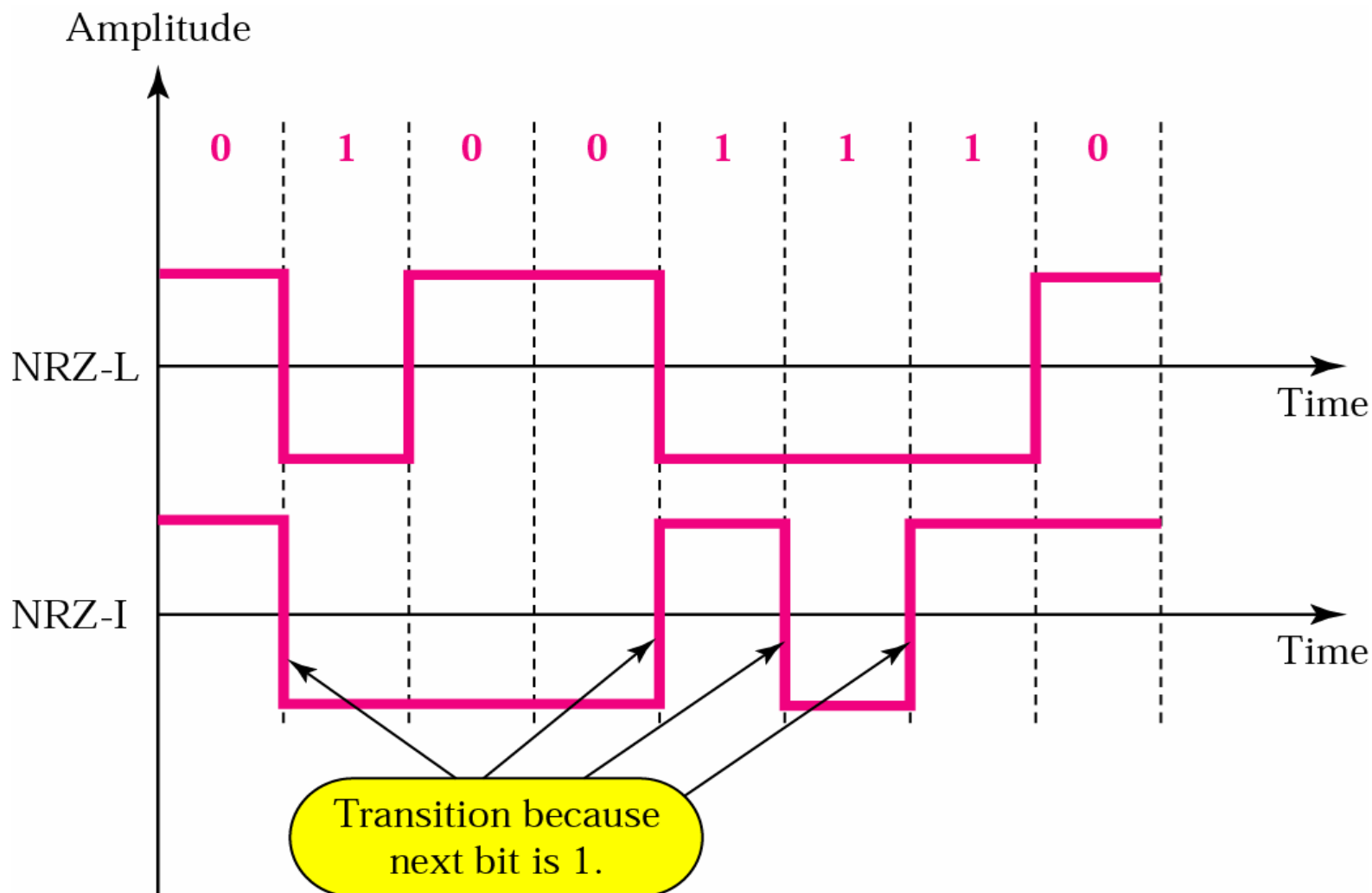
- **NRZ-I (Invert)**

สัญญาณจะเปลี่ยนขั้วก็ต่อเมื่อพบ บิต = 1 ในข้อมูล

หมายเหตุ ถ้าเกิดบิต = 0 ในตัวกลาง ติดกัน จะไม่เกิดการเปลี่ยนแปลง ซึ่งอาจจะเสีย Synchronization ได้ แต่มีโอกาสน้อย



NRZ-I and L Examples

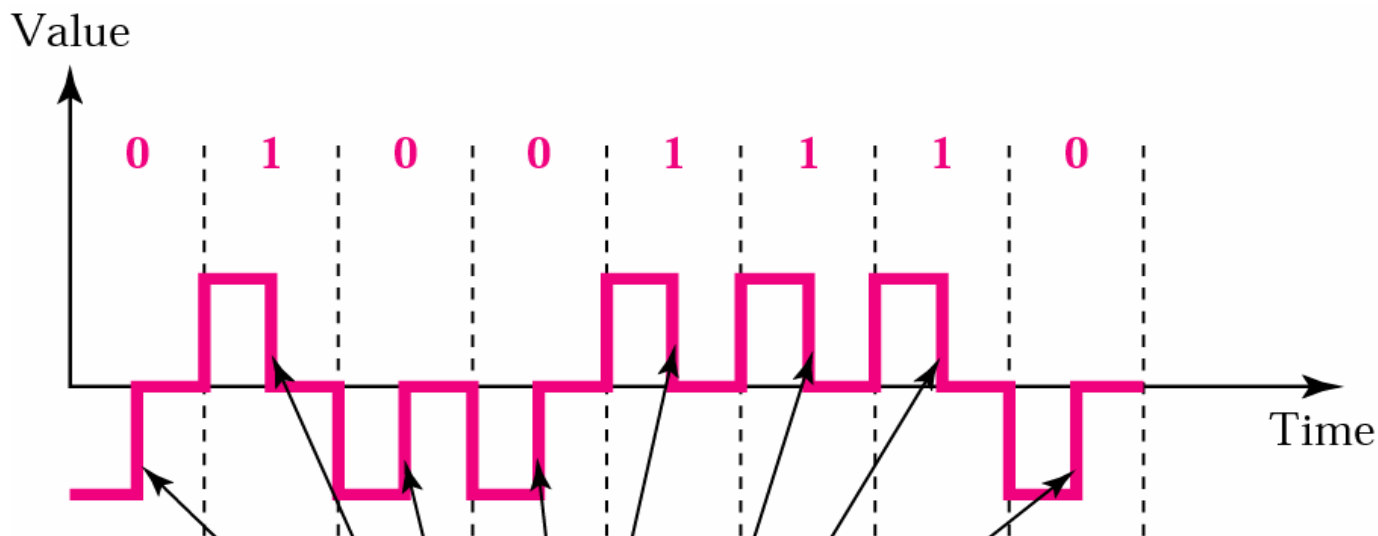




Return to Zero (RZ)

การส่งสัญญาณที่มีการเพิ่ม **ตัวบ่งชี้** ภายใน Pulse เพื่อสร้าง **Self-Sync.** ป้องกัน
ในกรณีที่อาจเกิด 0 หรือ 1 ติดกัน

ด้วยการวิธีคล้ายกับ NRZ-L (+/-) ทุกครั้งสัญญาณจะกลับไประดับ 0 ที่ กลางบิต



These transitions can be used
for synchronization.

ข้อเสีย

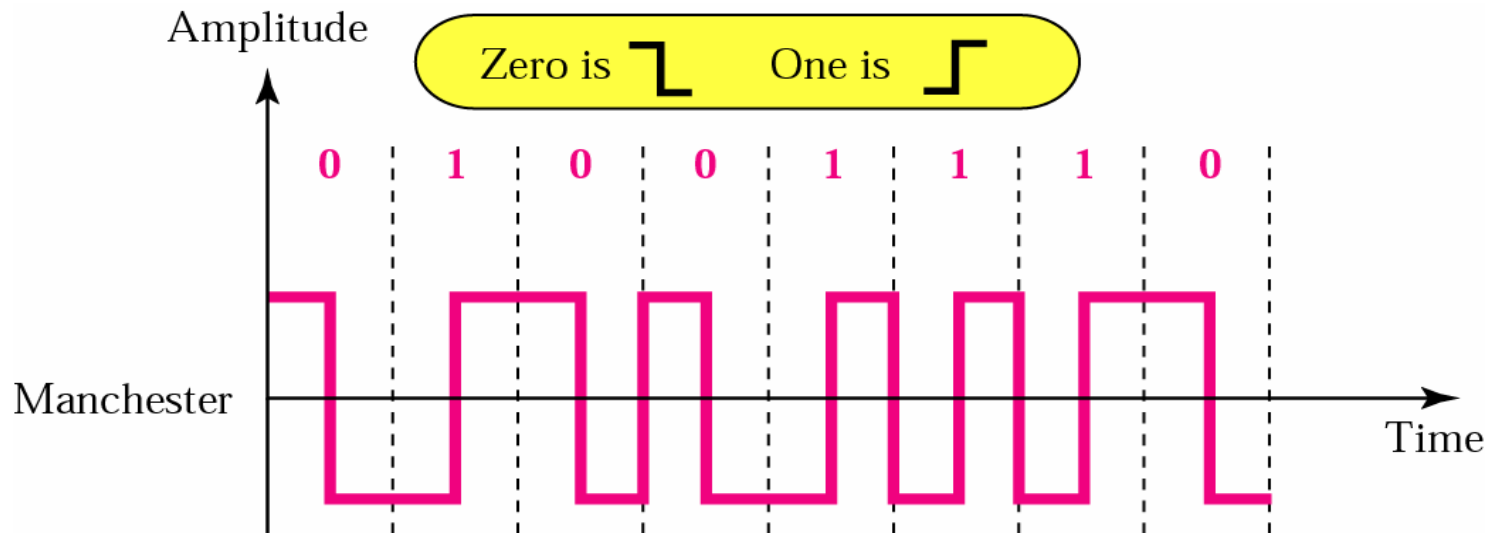
มีการเปลี่ยน
ระดับสัญญาณ
2 ครั้ง ต่อการ
แสดงข้อมูล 1
บิต

Pulse Rate > Bit Rate

Manchester

การเข้ารหัสแบบนี้ ใช้การเปลี่ยนแปลง (Transition) กึ่งกลางบิต เพื่อจุดประสงค์

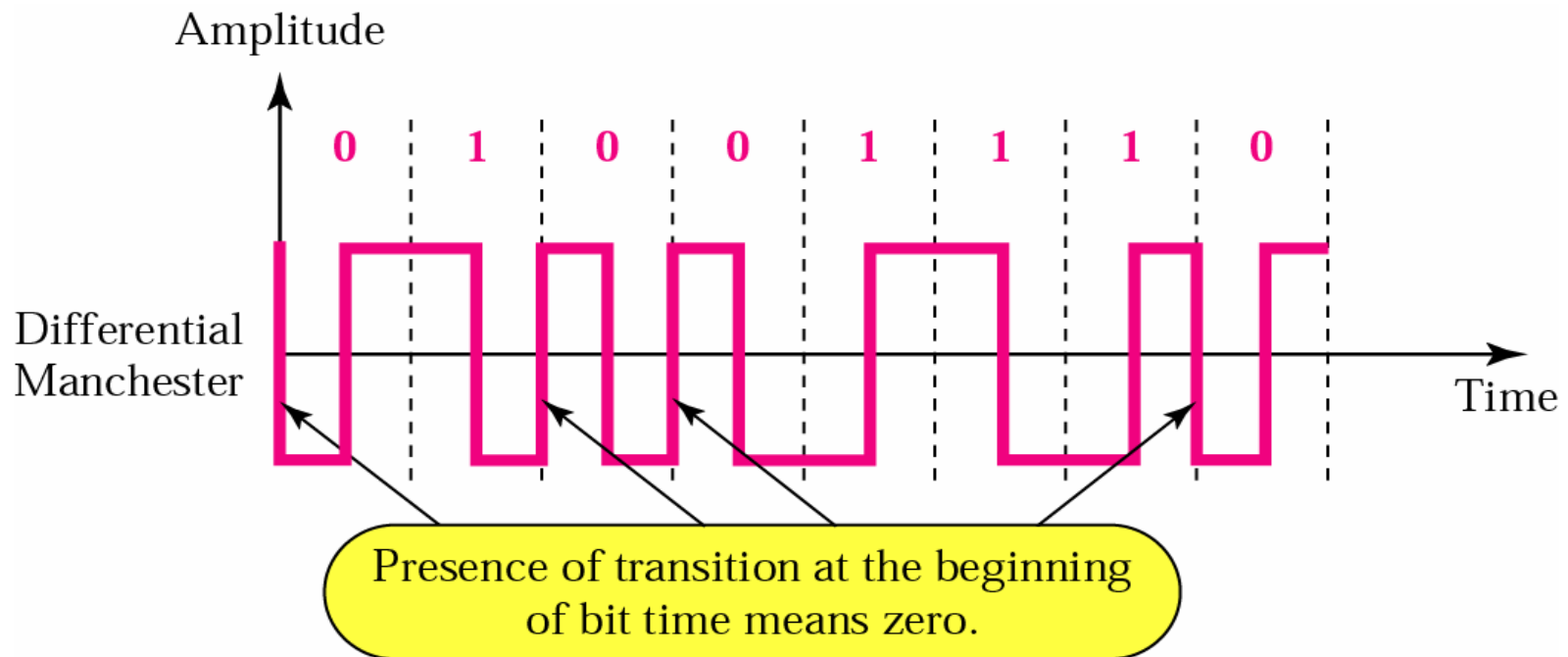
- 1) นำเสนอข้อมูล 0 และ 1 (รูปแบบ Transition ต่างกัน) และ
- 2) ใช้เป็น Self Synchronization



ข้อดี แม้ว่า มีการเปลี่ยนระดับสัญญาณ 2 ครั้ง ต่อการแสดงข้อมูล 1 บิต แต่ใช้ระดับสัญญาณเพียง 2 ค่า ซึ่งทำให้ Bandwidth เฉลี่ย ต่ำกว่าแบบ RZ

Differential Manchester

การเข้ารหัสแบบนี้ ใช้การเปลี่ยนแปลง (Transition) ที่กลางบิตเพื่อใช้เป็น Self Synchronization เท่านั้น ส่วนค่าของบิต จะพิจารณาจาก Transition ที่จุดเริ่มต้นของบิต ถ้ามี = 0 มิฉะนั้น = 1



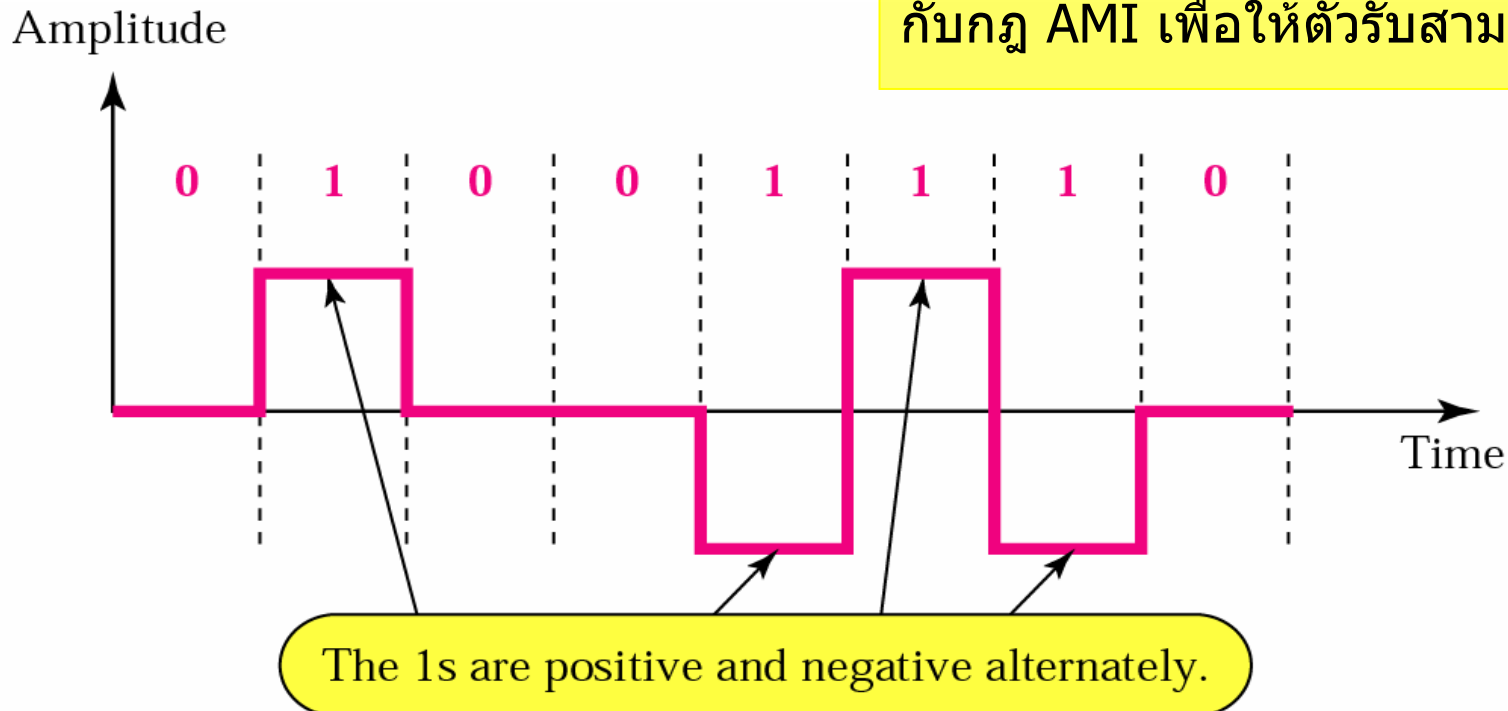
ข้อดี ถ้ามีเลข 1 ติดกันจำนวนมาก ค่า Bandwidth โดยเฉลี่ยจะลดลง

Bipolar

Bipolar คล้ายกับ RZ โดยใช้ระดับสัญญาณ 3 ระดับ (+ – และ 0) โดยแทนเลข 0 ด้วยระดับ 0 และเลข 1 ด้วย +/– สลับกัน

Alternate Mark Inversion (AMI)

ถ้ามีเลข 0 ที่ติดกันยาว **BnZS** จะแทนเลข 0 ตัวที่ n ด้วย + หรือ – ซึ่งขัดแย้งกับกฎ AMI เพื่อให้ตัวรับสามารถแยกได้

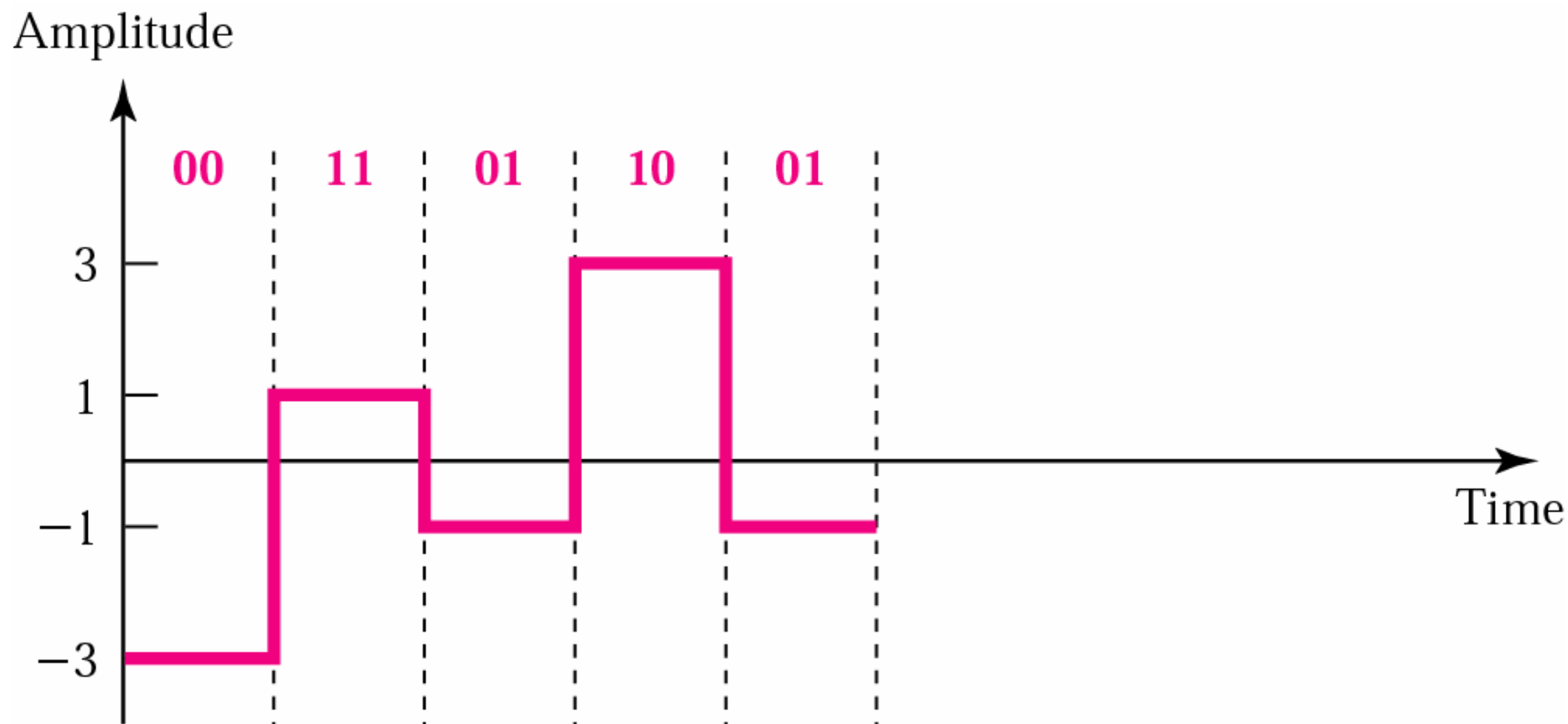




2-Binary 1-Quaternary

2B1Q เพิ่มประสิทธิภาพของการส่งสัญญาณ โดยให้ 1 Pulse แทนตัวเลข 2 บิต

$$\text{Bit Rate} = \text{Pulse Rate} \times \log_2 4$$





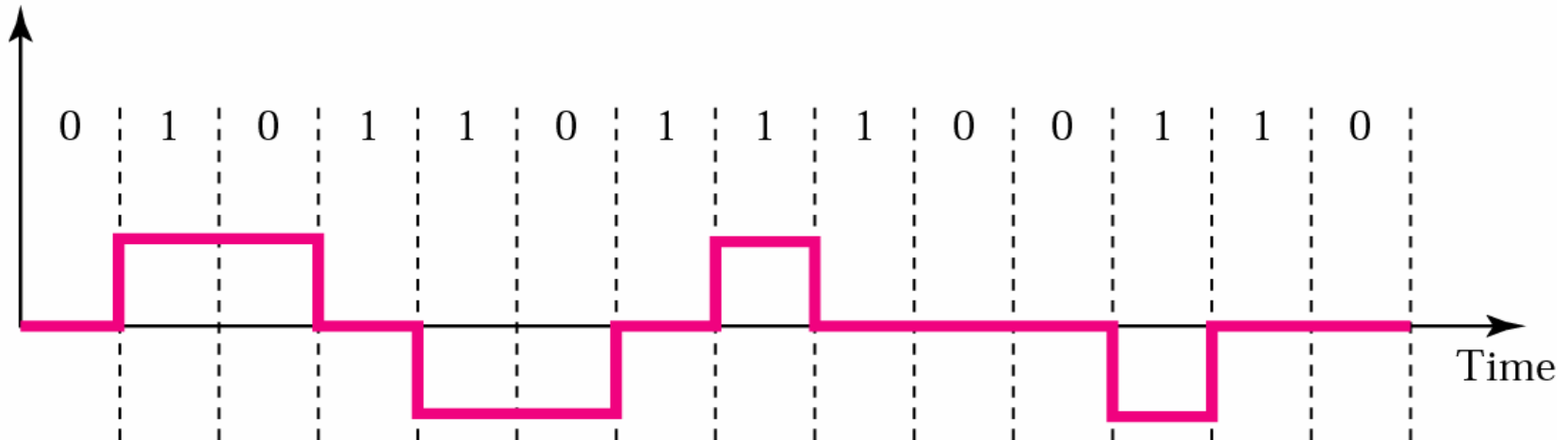
Multiline Transmission 3 Levels

MLT-3 คล้ายกับวิธีการเข้ารหัสแบบ NRZ-I แต่ใช้ 3 ระดับ (+, - และ 0)

ถ้าข้อมูลเป็น 0 จะไม่มี Transition

ถ้าข้อมูลเป็น 1 จะมี Transition ดังนี้ $+$ $\rightarrow$ $0 \rightarrow -$ และย้อนกลับ ตามลำดับ

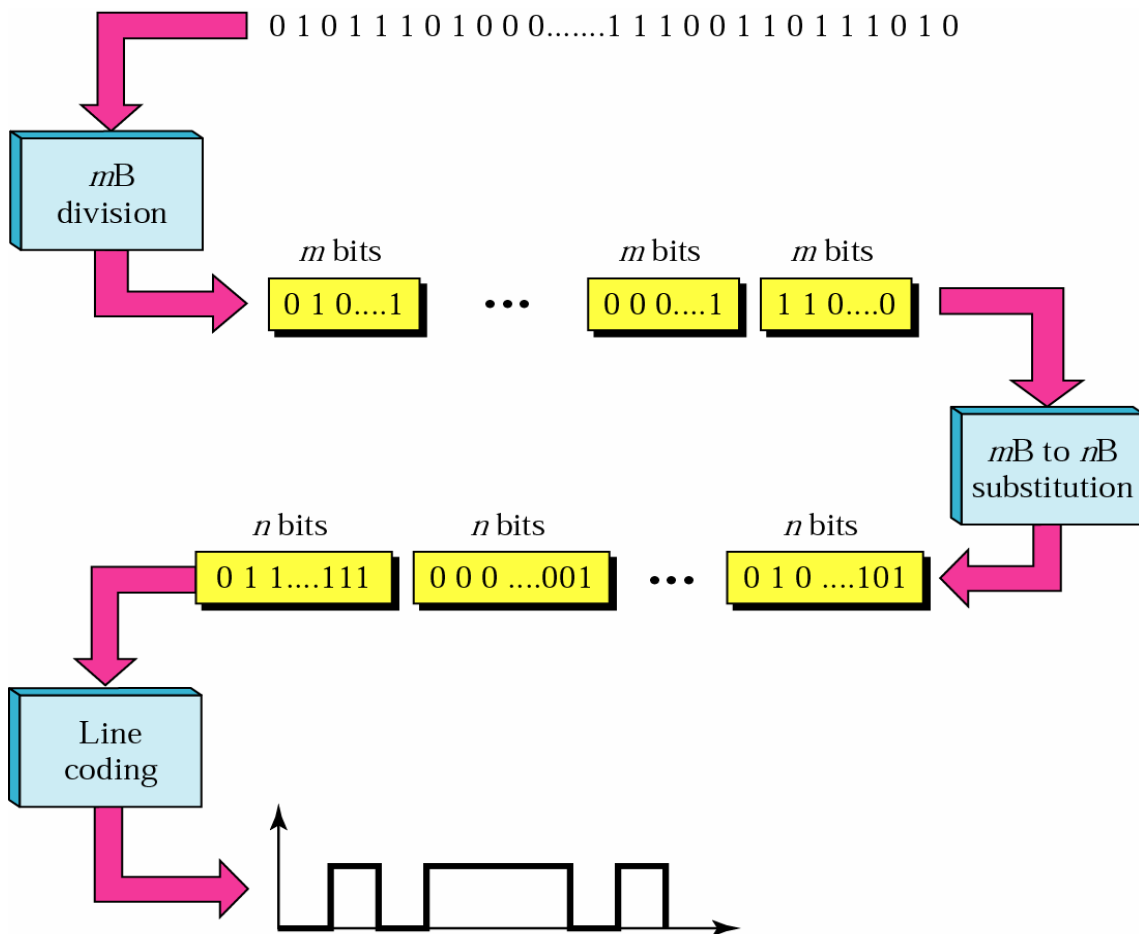
Amplitude



หมายเหตุ การเปลี่ยนแปลงจะเกิดที่ต้นของบิต

Block Coding

คือการเข้ารหัสเป็นกลุ่มเพื่อ Self-Synchronization และ ตรวจจับ Error



Steps

1. Division
2. Substitution
3. Line Coding

Common Schemes

1. 4B/5B
2. 8B/10B
3. 8B/6T



Block Coding Steps

- **Division**

คือการแบ่งอนุกรมของข้อมูลออกเป็นกลุ่มๆ เช่น 4B/5B แบ่งข้อมูลออกเป็นกลุ่มละ 4 บิต

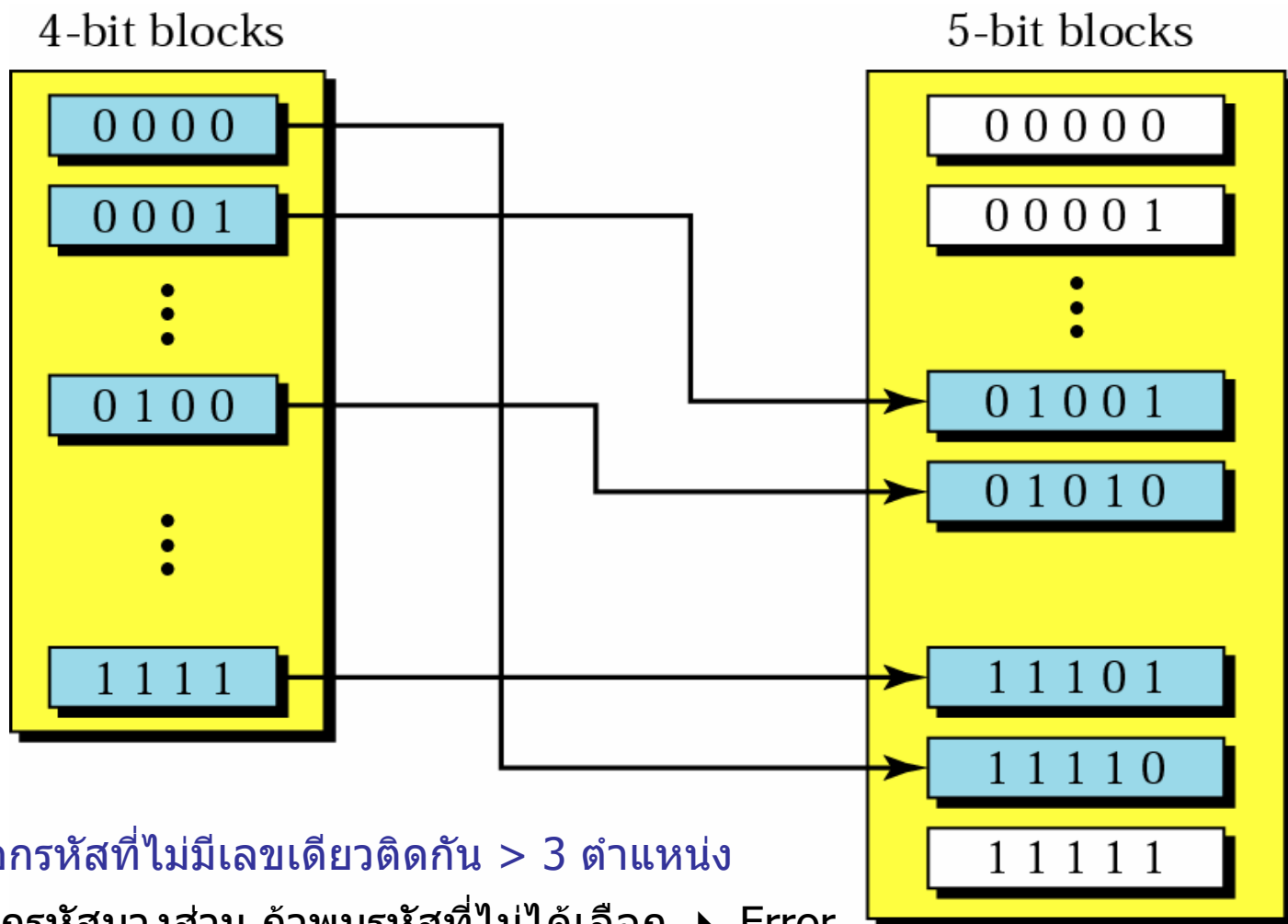
- **Substitution**

คือการแทนรหัส m บิตด้วยกลุ่มที่มีขนาด n บิต ($n > m$) เช่น 4B/5B แทนรหัส 4 บิต (มีกลุ่มที่แตกต่างกัน 16 กลุ่ม) ด้วยกลุ่มที่มีขนาด 5 บิต (มีกลุ่มที่แตกต่างกัน 32 กลุ่ม) โดยเลือกมาเพียงบางส่วน

- **Line Coding**

คือการแปลงข้อมูลเป็นสัญญาณ มักใช้วิธีที่ง่ายที่สุด

Substitution





4B/5B

Data	Code	Data	Code
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

ไม่มี 0 นำหน้าเกิน 1 ตัว ไม่มี 0 ต่อท้ายเกิน 2 ตัว (= ไม่มี 0 ติดกันเกิน 3 ตัว)



4B/5B Control

Data	Code
Q (Quiet)	00000
I (Idle)	11111
H (Halt)	00100
J (Start Delimiter)	11000
K (Start Delimiter)	10001
T (End Delimiter)	01101
S (Set)	11001
R (Reset)	00111

รหัสควบคุมจะไม่ใช่ไป
ตามกฎของรหัสข้อมูล

รหัส 8B/10B คล้ายกับ
4B/5B

แต่เพิ่มความสามารถใน
การตรวจจับ Error ดีกว่า

การบ้าน แปลงรหัส นศ.
โดยใช้ 4B/5B

(Bonus +2) สำหรับการ
เขียนโปรแกรม C/C++



8B/6T

ออกแบบมาเพื่อประหยัด Bandwidth โดยแทนกลุ่มเลข 8 บิต ด้วยรหัส 6 บิต

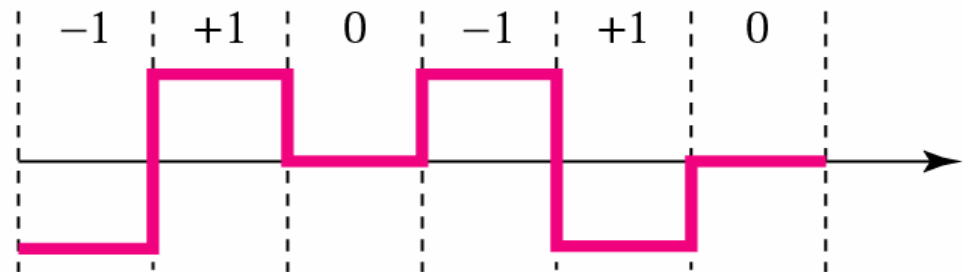
Why is it Possible?

รหัส 8 บิตแทนตัวเลขได้ $2^8 = 256$ ตัวเลข

สัญญาณ 6 บิต มี 3 ระดับ (+, -, 0) แทนตัวเลขได้ $3^6 = 729$ ตัวเลข

หมายเหตุ Line Coding ถูกรวมไว้ในขั้นตอน Substitution

00011111



Appendix D ในหนังสือแสดงตารางรหัสดังกล่าว



Conclusions

- Line Coding
- Block Coding
- Sampling
 - Pulse Amplitude Modulation
 - Pulse Code Modulation
 - Nyquist Theorem
- Transmission Mode
 - Parallel Transmission
 - Serial Transmission