



Error Detection and Correction

Paramate Horkaew

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



Lecture Outline

- **Type of Errors**
 - Single Bit Error
 - Burst Error
- **Detection**
 - Redundancy
 - Parity Check
 - Cyclic Redundancy Check (CRC)
 - Checksum
- **Error Correction**
 - Retransmission
 - Forward Error Correction
 - Burst Error Correction



Error Correction

เมื่อตรวจพบ Error (ด้วยวิธี Parity Check, CRC หรือ Checksum ฯลฯ) แล้ว กระบวนการถัดไป คือการแก้ไข Error นั้น (**Error Correction**) ซึ่งมี 3 วิธี ดังนี้

- **Retransmission**

วิธีแก้ไข Error ที่ง่ายที่สุด คือเมื่อตรวจพบ อุปกรณ์ด้านส่งจะแจ้งไปยัง อุปกรณ์ด้านส่ง เพื่อร้องขอ (REQ) ให้มีการส่งข้อมูลมาใหม่ทั้งหมด

- **Forward Error Correction (FEC)**

คือวิธีที่แก้ไข Error อัตโนมัติที่ด้านรับ ทั้งนี้โดยอาศัยกลไกของการเข้ารหัส แบบพิเศษ เรียกว่า Error Correcting Code ซึ่งมีความซับซ้อนมากกว่าการ ตรวจจับ Error ทั่วไป และต้องการ Redundant Bits มากกว่า

- **Burst Error Correction**

คือวิธีที่แก้ไข Burst Error โดยอาศัยการจัดเรียงข้อมูลในรูปแบบใหม่



Forward Error Correction

ในทางทฤษฎี Error Correcting Code สามารถแก้ไข Error ได้ทุกชนิด ทุกรูปแบบ เรียกการแก้ไข Error แบบนี้ว่า **FEC**

ตัวอย่าง Single Bit Error

การตรวจจับและแก้ไข Error ชนิดนี้ ต้องสามารถระบุ "ข่าวสาร" 2 ประการ

- มี Error เกิดขึ้นหรือไม่ (Detection)
- Error ที่เกิดขึ้นนั้น อยู่ที่ตำแหน่งใด (Identification)

เพียงตรวจจับ Error ต้องการ Redundant Bit อีกเพียง 1 บิตเท่านั้นเพื่อระบุว่าชุดข้อมูลนั้นมี Error เกิดขึ้น (1) หรือไม่ (0)

แต่การแก้ไข Error นั้นต้องสามารถ "ระบุตำแหน่งที่เกิด Error" (Identification of Invalid Bit) ได้ ซึ่งอุปกรณ์ด้านรับ เพียงกลับค่าบิต ของข้อมูล ณ ตำแหน่งนั้น



Identifying Invalid Bit(s)

สำหรับข้อมูลชนิด ASCII การแก้ไข Single Bit Error ต้องสามารถระบุ **สถานะ** ของข้อมูล ถึง 8 สถานะ ได้แก่

000 ไม่มี Error	001 Error ณ บิต 1	010 Error ณ บิต 2	011 Error ณ บิต 3
100 Error ณ บิต 4	101 Error ณ บิต 5	110 Error ณ บิต 6	111 Error ณ บิต 7

จากตารางดูเหมือนว่า Redundancy ต้องการเพียง 3 Bit ในการระบุตำแหน่งที่เกิด Error ได้ 7 ตำแหน่ง ($2^3 - 1$) แต่เงื่อนไขนี้ ไม่สามารถระบุตำแหน่งที่เกิด Error ใน Redundant Bit เองได้

ดังนั้นถ้าต้องการ แก้ไข Error ได้ทุกกรณี ในที่นี้ หมายถึง Error ที่เกิดขึ้นใน Data Bit (**7 สถานะ**) ใน Redundant Bit (**3 สถานะ**) และไม่เกิด Error (**1 สถานะ**) ระบบจึงต้องการบิตเพิ่มเติม



Counting Redundant Bits

ถ้าต้องการส่ง Data จำนวน m bit และ Redundancy จำนวน r bit แล้ว Redundant Bit จะต้องสามารถระบุตำแหน่ง Error ได้ถึง $m + r + 1$ สถานะ (สถานะสุดท้ายหมายถึง ไม่เกิด Error) ดังนั้น ความสัมพันธ์ระหว่างความยาว m และ r จึงเขียนได้เป็น

$$2^r \geq m + r + 1$$

ตัวอย่าง

ถ้าต้องการส่งข้อมูลในรหัส ASCII ความยาว 7 บิต จงหาว่าต้องใช้ Redundant Bit อย่างน้อยจำนวนเท่าใด

กำหนดให้ $m = 7$

ดังนั้น $2^r \geq 7 + r + 1$

ย้ายข้างสมการ $2^r - 1 \geq 8$

จะได้ว่า $r = 4$; ต้องส่งข้อมูลทั้งสิ้น $m + r + 1 = 7 + 4 + 1 = 11$ บิต



Hamming Code

Hamming Code เป็นเทคนิคหนึ่งของ FEC ซึ่งสามารถตรวจจับและแก้ไขข้อผิดพลาดของข้อมูลซึ่งมีความยาวใดๆ ได้

โดยที่ **Redundant Bit** จะปรากฏในตำแหน่งที่เป็นกำลังของ 2 ดังรูป

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

โดยที่ R Bit แต่ละตำแหน่งเป็น Parity ของกลุ่ม Data Bit ที่กำหนด (ลวงหน้า)

- r_1 : Bits 1, 3, 5, 7, 9, 11
- r_2 : Bits 2, 3, 6, 7, 10, 11
- r_3 : Bits 4, 5, 6, 7
- r_4 : Bits 8, 9, 10, 11

สังเกตว่า

- Data Bit มีการคำนวณอย่างน้อย 2 ครั้ง
- R Bit มีการคำนวณ 1 ครั้ง



Redundancy Bits Calculation

r_1 will take care of these bits.

11		9		7		5		3		1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_2 will take care of these bits.

11	10			7	6			3	2	
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_4 will take care of these bits.

				7	6	5	4			
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

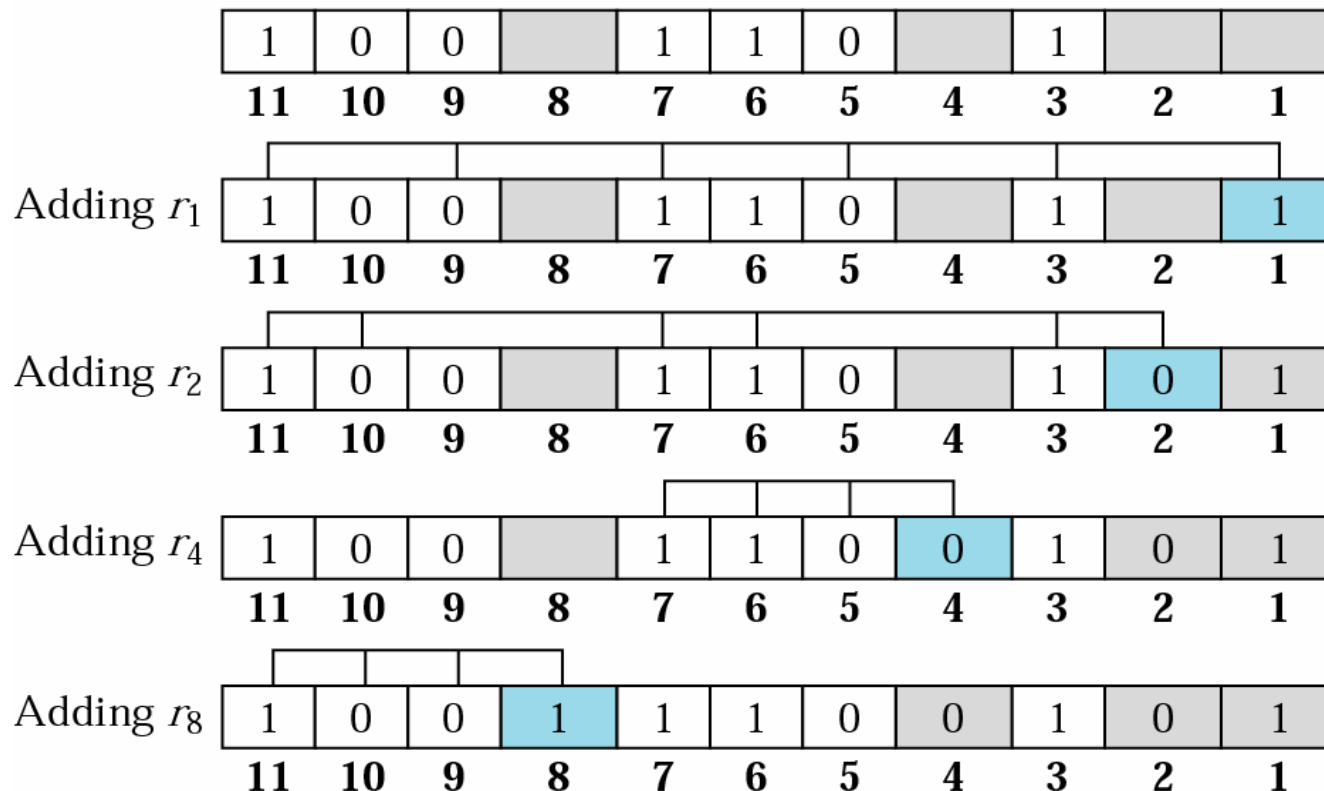
r_8 will take care of these bits.

11	10	9	8							
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

ค่า r_i คำนวณได้จากค่า **Even Parity** ของ Data Bit ในแต่ละกลุ่ม

Example

Even Parity เพิ่มบิตท้ายหน่วยข้อมูล เพื่อให้จำนวน 1 ทั้งหมดรวม Parity เป็นคู่
ตัวอย่างต่อไปนี้แสดงการคำนวณ r bit จาก data $m = 100\ 1101$

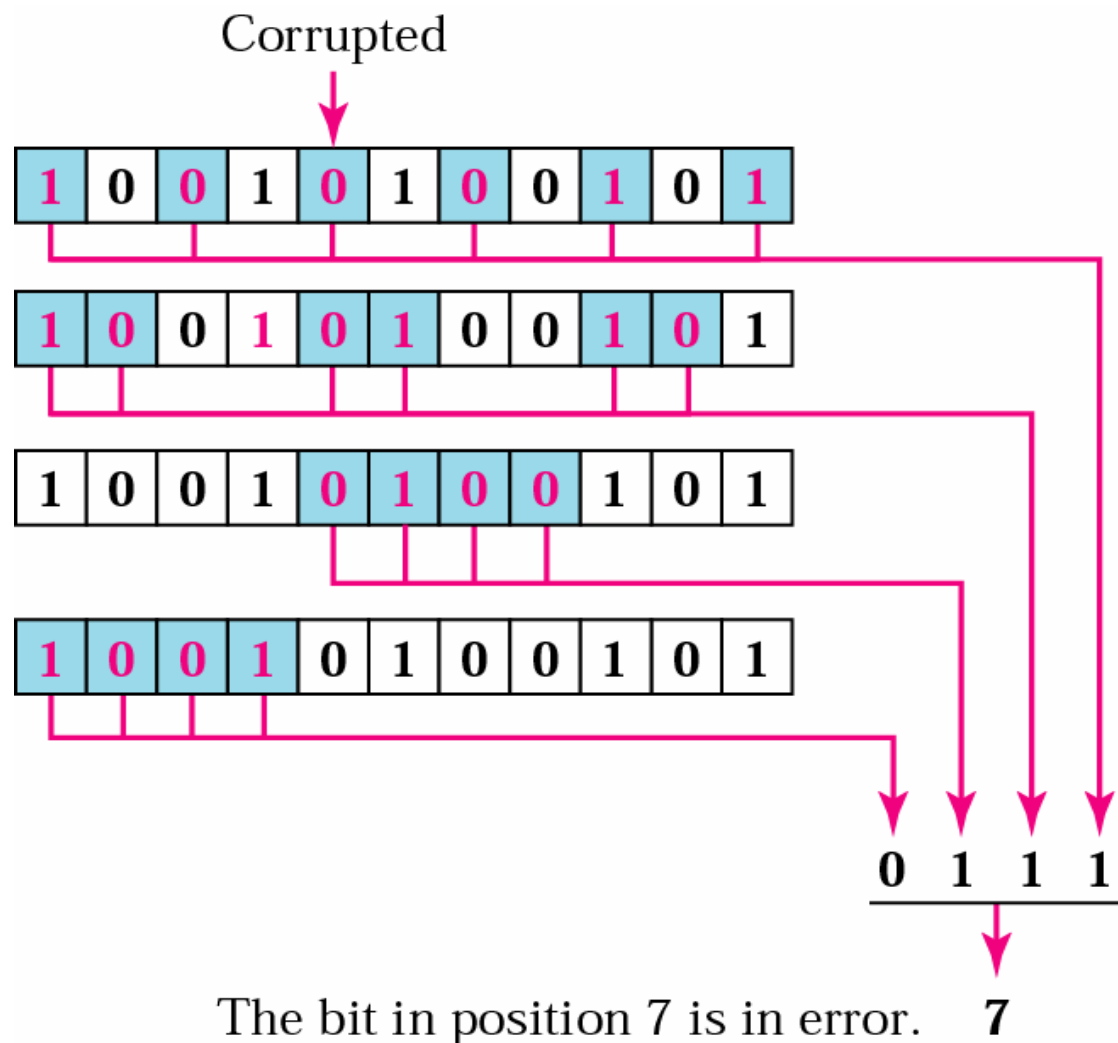


Data:
1 0 0 1 1 0 1

Code:
1 0 0 1 1 1 0 0 1 0 1



Error Detection and Correction



ในที่นี้สมมติให้ Bit ข้อมูลในตำแหน่งที่ 7 เกิด Error

ทางด้านรับคำนวณ Parity Check สำหรับบิตข้อมูลแต่ละกลุ่ม เหมือนทางด้านส่ง

โดยเขียนเรียงตัวเลขตามตำแหน่ง (r_8, r_4, r_2, r_1) ดังรูป

ซึ่งจากตัวอย่างเห็นว่า Parity Check ที่ได้เป็นเลขฐาน 2 ของ 7 ซึ่งแสดงถึงตำแหน่งที่เกิด Error



Burst Error Correction

วิธีนี้ไม่สามารถแก้ไข Burst Error ได้โดยตรง แต่ถ้าจัดข้อมูลเป็นกลุ่มกลุ่มละ N ชุด (แถวละ 1 ชุด) แล้วส่งข้อมูลนั้นเรียงไปตาม Column ถ้าเกิด Burst Error จำนวน M บิต ถ้า ($M < N$) แล้วจะเกิดข้อผิดพลาด ต่อข้อมูลชุดละ 1 บิตเท่านั้น

Error → 1111?000011

Error → 1010?011111

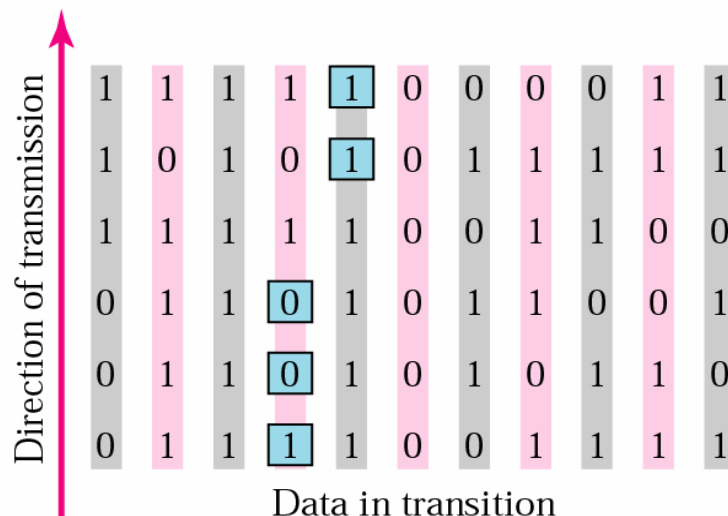
11111001100

Error → 011?1011001

Error → 011?1010110

Error → 011?1001111

Received data



11111000011

10101011111

11111001100

01101011001

01101010110

01111001111

Data before being sent



Lecture Outline

- **Type of Errors**
 - Single Bit Error
 - Burst Error
- **Detection**
 - Redundancy
 - Parity Check
 - Cyclic Redundancy Check (CRC)
 - Checksum
- **Error Correction**
 - Retransmission
 - Forward Error Correction
 - Burst Error Correction