데이터 통신

제 9강 오류검출과 정정

- 소　　속 : 한국기술교육대 컴퓨터공학부
- 담당교수 : 김 원 태 교수
- 이 메 일 : wtkim@koreatech.ac.kr

9.1 오류의 종류

9.2 오류의 검출

9.3 오류의 정정

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 단일-비트 오류(Single-Bit Error)

- 데이터 부분의 한 비트만 변경

  (예 : ASCII STX - ASCII LF)

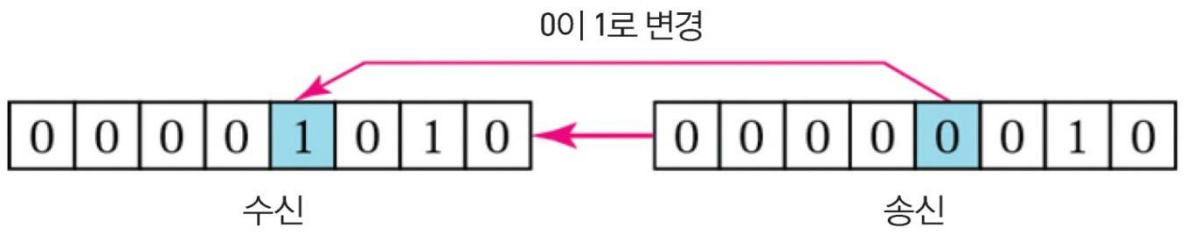

그림 9.2 단일 비트 오류

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 다중-비트 오류(Multiple-Bit Error)

- 데이터 부분의 2개 또는 그 이상의 비연속적인 비트가 변경(예 : ASCII B - ASCII LF)

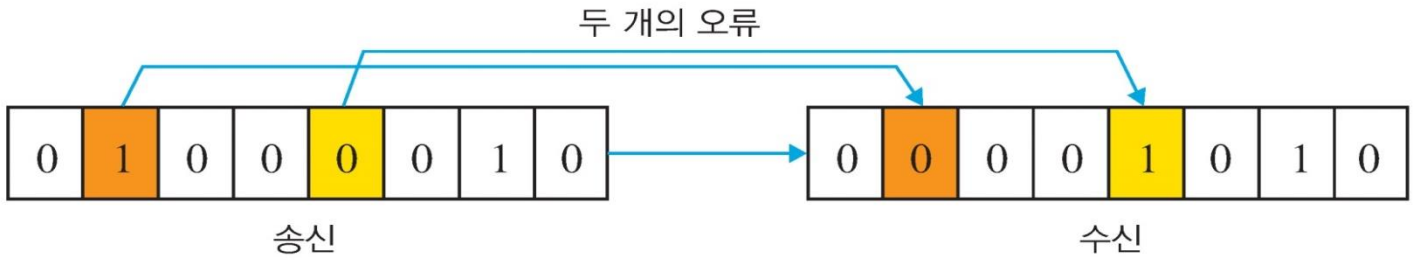

그림 9.3 다중 비트 오류

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 집단 오류(Burst Error)

- 데이터 부분의 2개 또는 그 이상의 연속적인 비트가 변경

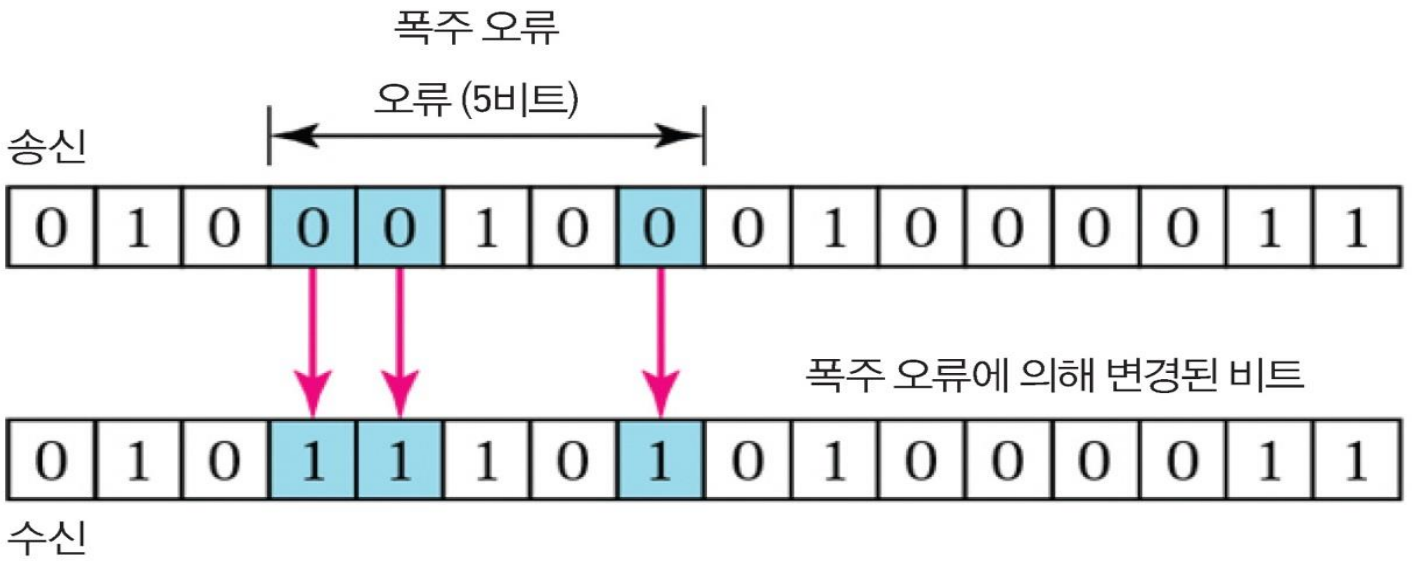

그림 9.4 폭주 오류

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 오류 검출은 목적지에서 오류를 검출하기 위해서 여분의 비트를 추가하는 중복(redundancy, 잉여) 개념을 이용

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 중복(redundancy)

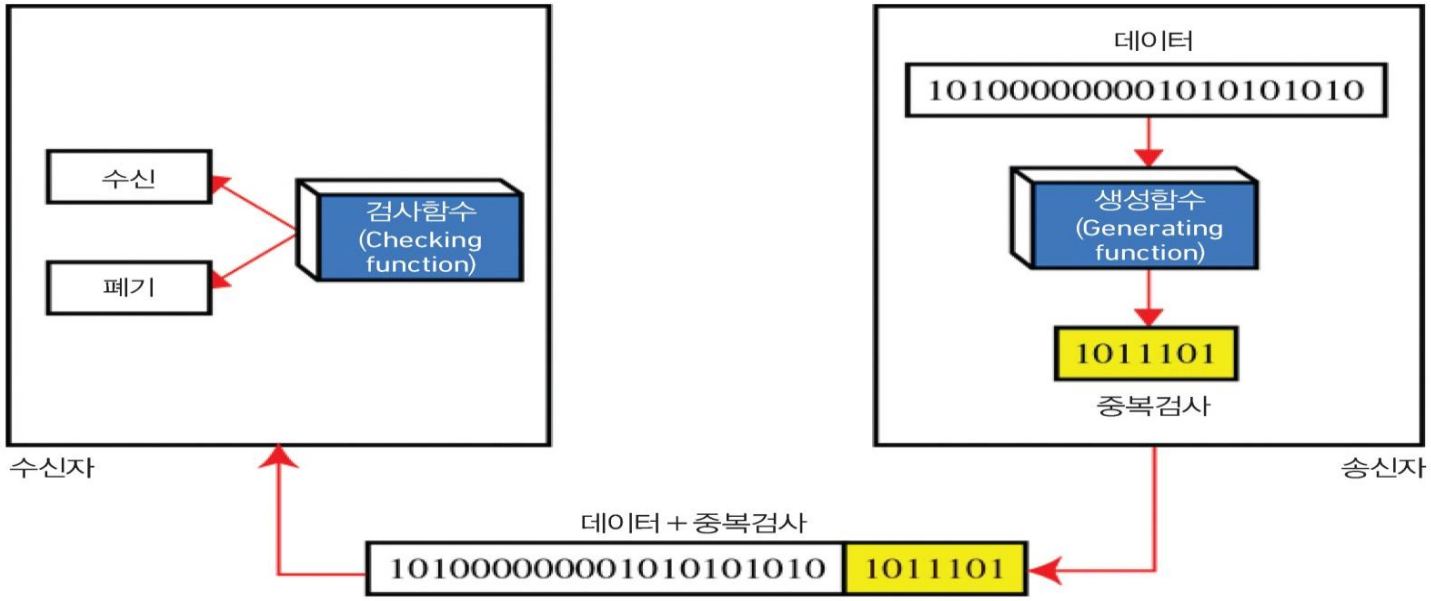

그림 9.5 중복

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 오류 검출 방법

- VRC(Vertical Redundancy Check)
- LRC(Longitudinal Redundancy)
- CRC(Cyclical redundancy Check)
- Checksum

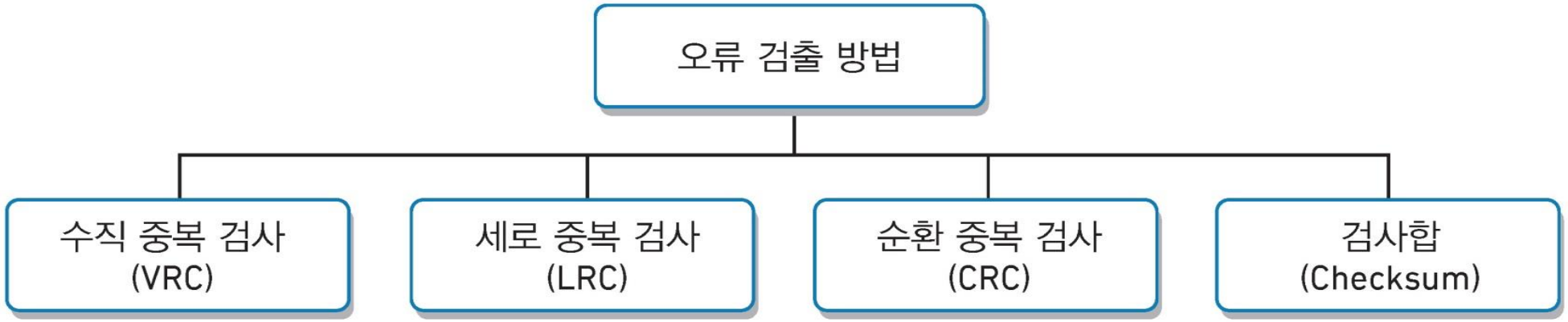

그림 9.6 오류 검출 방법

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ VRC(Vertical Redundancy Check)

- 각 데이터 단위에 패리티 비트가 추가되는데, 이 패리티 비트는 전체 데이터 단위에서 1의 개수가 홀수 또는 짝수가 되게 한다

❖ 짝수 패리티 VRC(Vertical Redundancy Check)

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

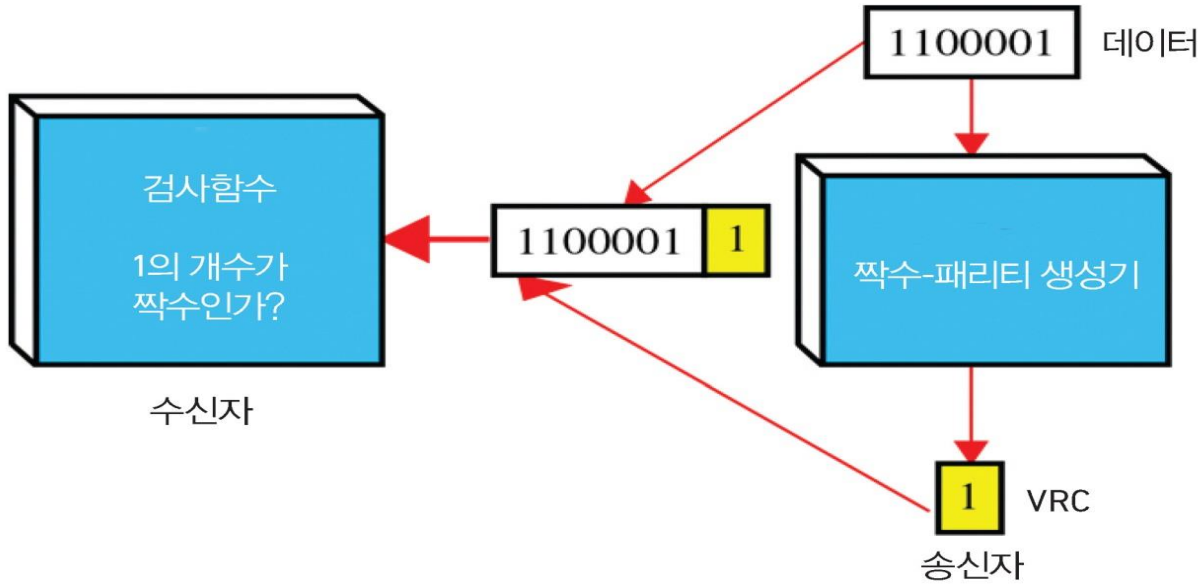

그림 9.7 짝수 패리티 VRC 개념

❖ 세로중복검사(LRC; Longitudinal Redundancy)

- 모든 바이트의 짝수 패리티를 모아서 데이터 단위로 만들어서 데이터 블럭의 맨 뒤에 추가

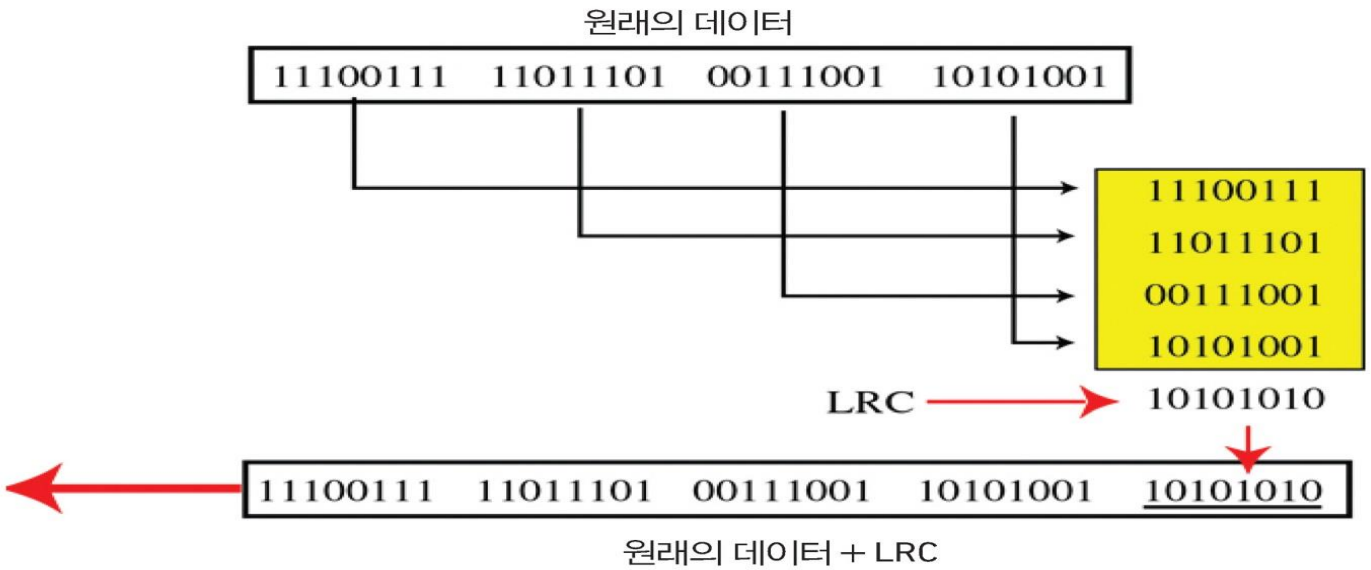

그림 9.8 세로 중복 검사

❖ 수직 중복검사와 세로 중복검사

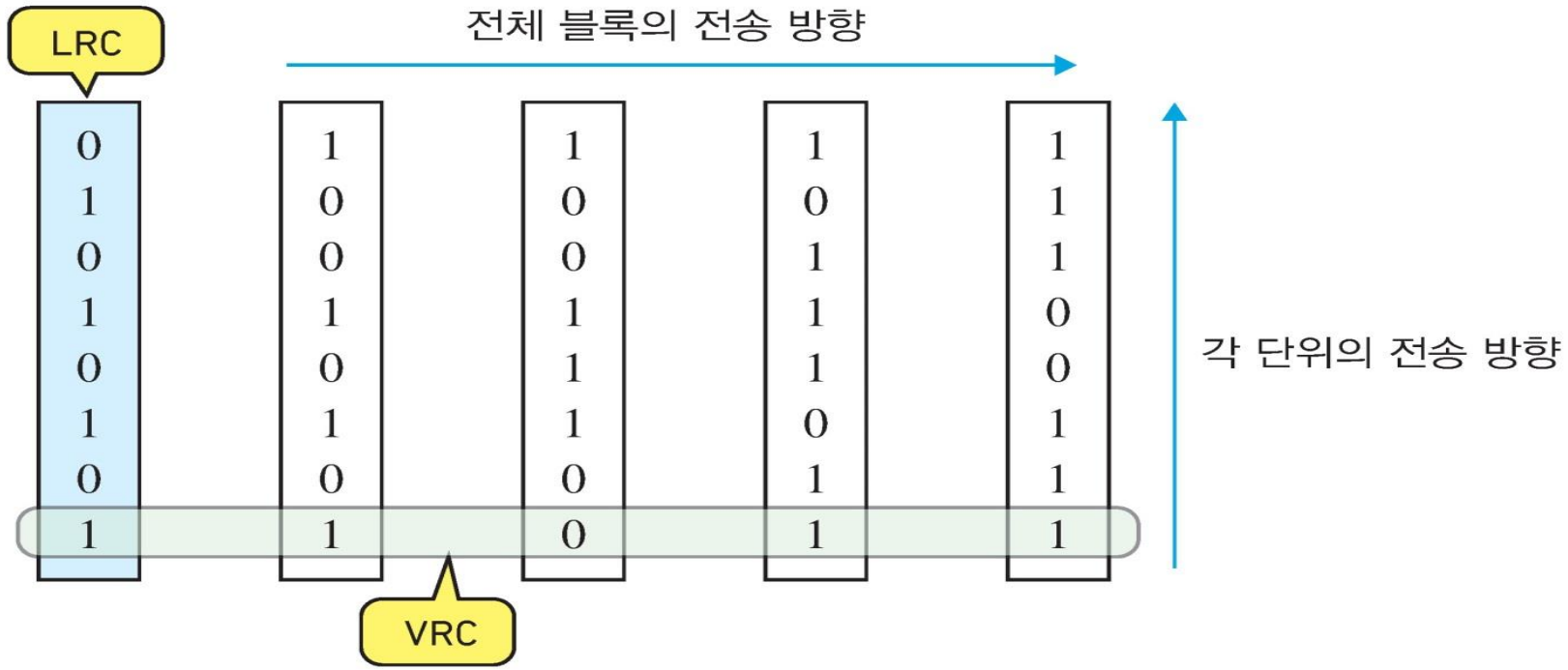

그림 9.9 수직 중복 검사와 세로 중복 검사

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 순환 중복 검사(CRC: Cyclic Redundancy Check)

- 2진 나눗셈을 이용

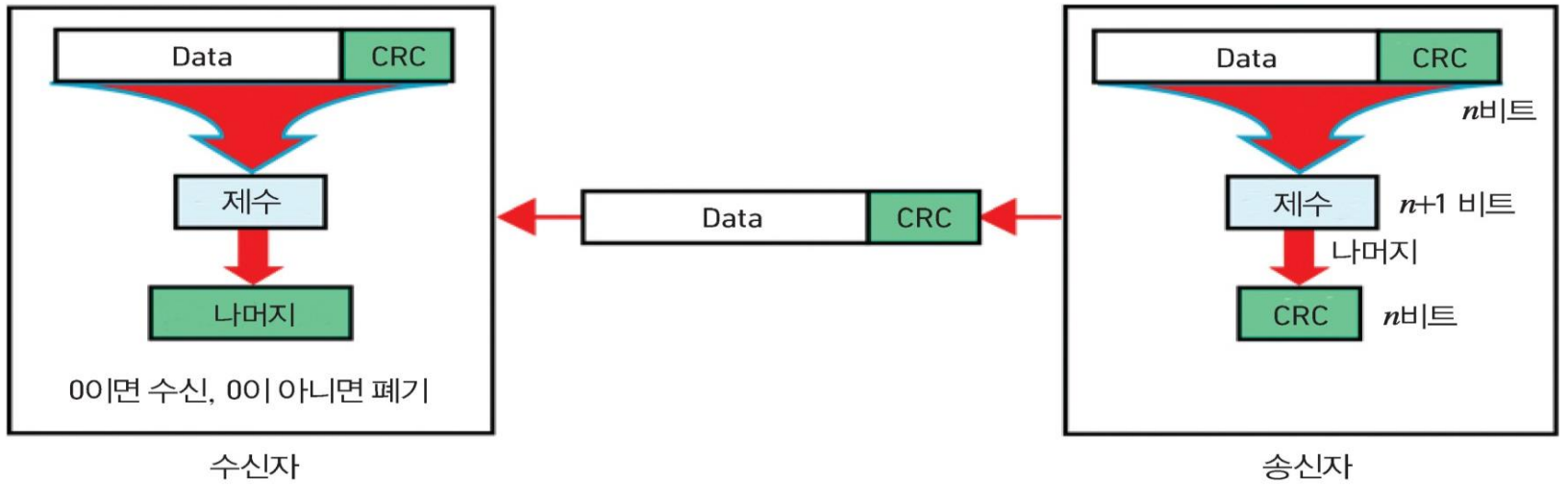

그림 9.10 CRC 발생기와 검사기

❖ CRC 발생기

- 모듈러-2 나눗셈을 이용

❖ 2진 나눗셈

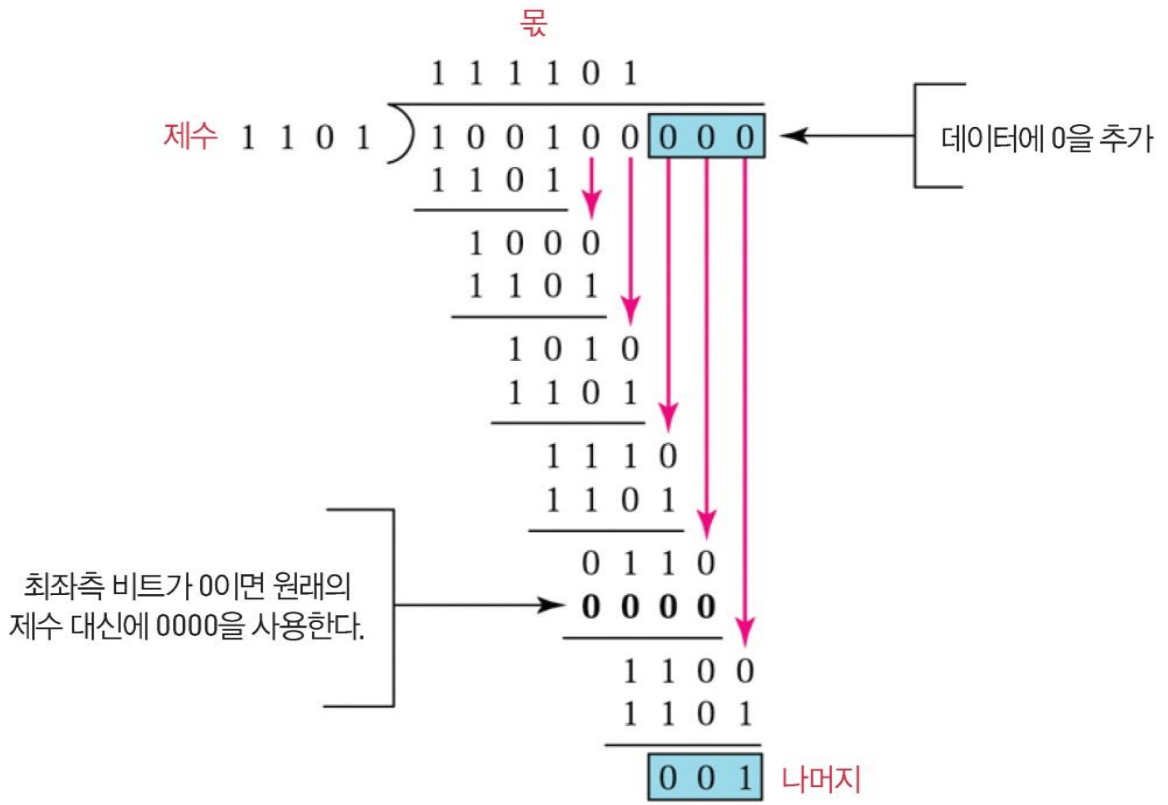

그림 9.11 2진 나눗셈

❖ CRC 검사기

- 모듈러-2 나눗

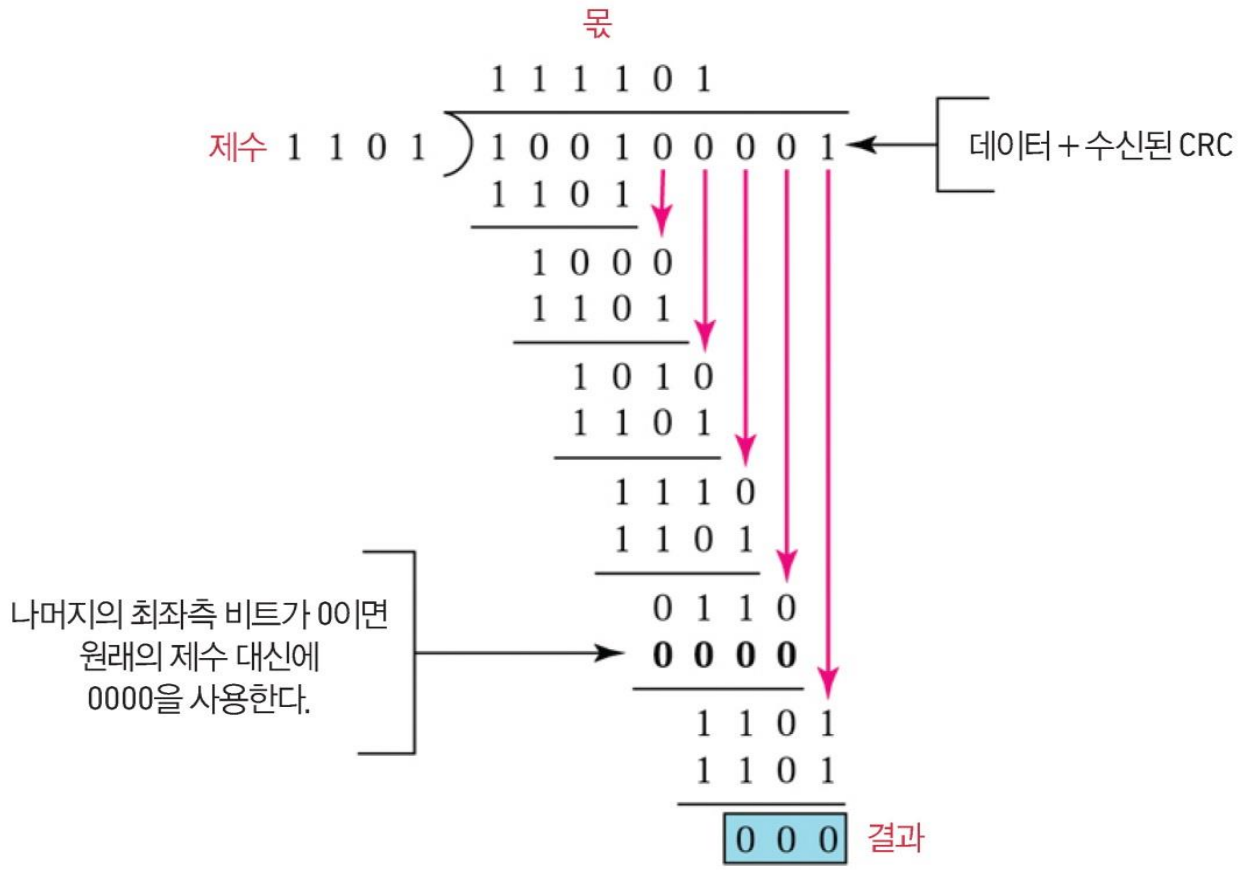

그림 9.12 CRC 검사기

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 다항식

- CRC 발생기는 1과 0의 스트링 보다는 대수식으로 표현

$$x^7 + x^5 + x^2 + x + 1$$

그림 9.13 다항식

❖ 하나의 다항식은 하나의 제수를 표현

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

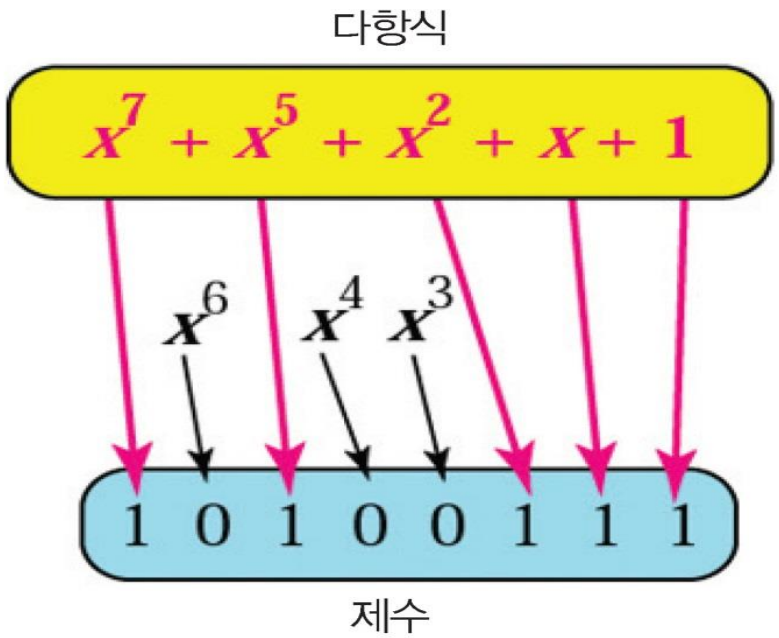

그림 9.14 제수를 표현하는 다항식

❖ 표준 다항식

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

명칭	다항식	응용
CRC-8	$x^8 + x^2 + x + 1$	ATM 헤더
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
ITU-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
ITU-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LAN

그림 9.15 표준 다항식

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 검사합(Checksum)

- 상위 계층 프로토콜에서 사용
- 중복(VRC, LRC, CRC ….) 개념을 기반으로 한다

■ 검사합을 생성하기 위해 송신자는 다음을 수행한다

➲ 단위를 길이가 n 비트인 K 섹션으로 나눈다

➲ 섹션 1과 2를 1의 보수를 이용하여 더한다

➲ 앞의 결과를 섹션 3과 더한다

➲ 앞의 결과를 섹션 4와 더한다

➲ 이 과정을 섹션 K까지 반복한다

➲ 최종 결과는 검사합을 만들기 위해 보수를 취한다

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 검사합(Checksum) 발생기

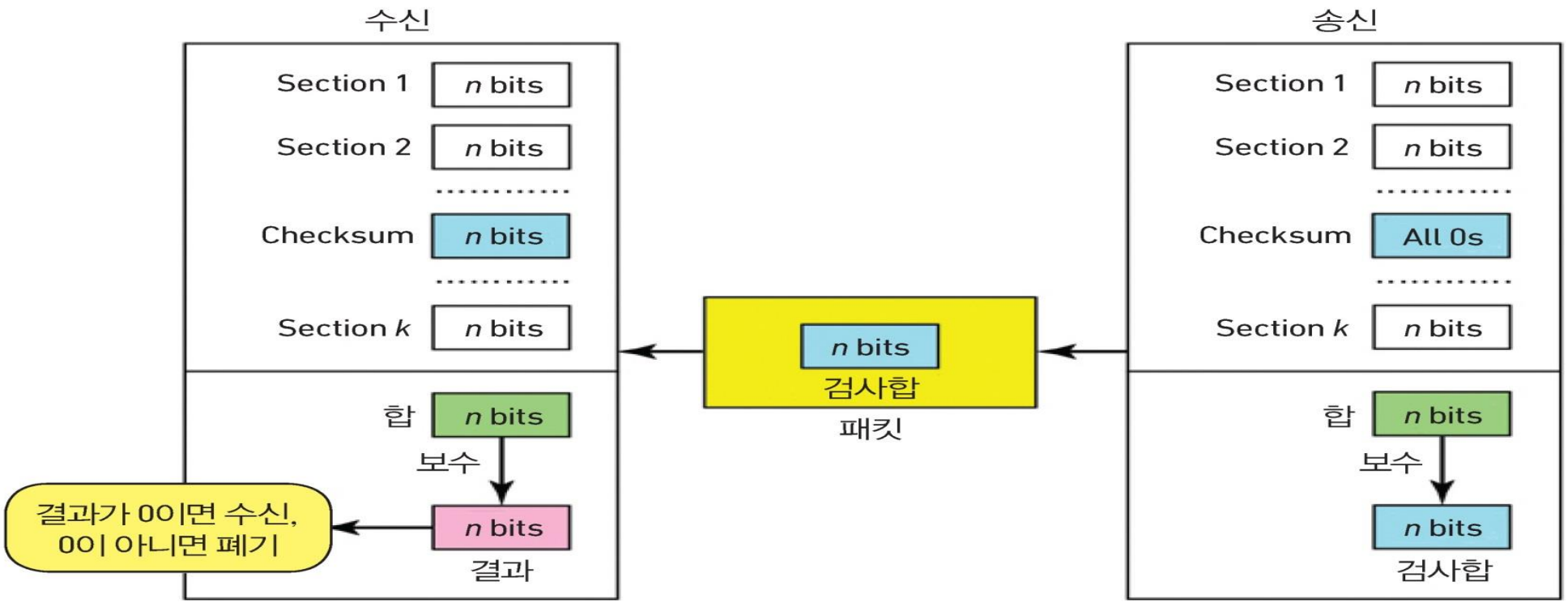

그림 9.16 검사합

❖ 데이터 단위와 검사합

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

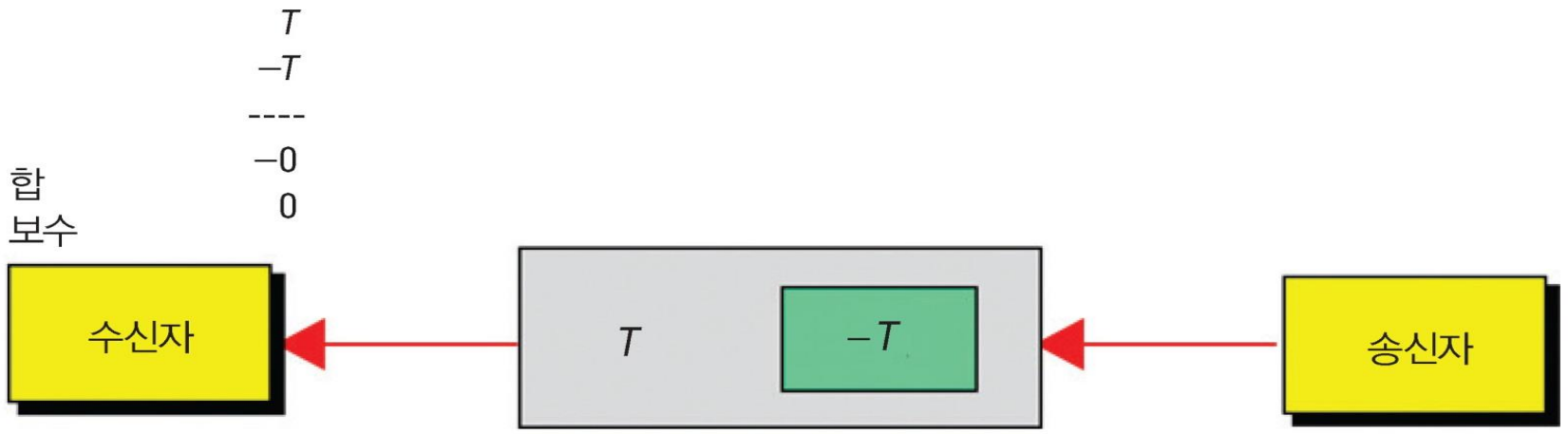

그림 9.17 데이터 단위와 검사합

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 두 가지 방법으로 처리한다

- 수신자가 송신자에게 전체 데이터 재전송 요구
- 수신자가 오류 교정 코드를 이용하여 자동으로 수행

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 단일 비트 오류 정정

- 패리티 비트
- 오류 정정의 비밀은 잘못된 비트의 위치를 알아내는 것
- ASCII 코드는 3-비트 잉여코드가 필요하다(000-111)

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 중복 비트

- 주어진 데이터 비트의 수(m)를 정정하기 위해 요구되는 중복비트 수(r)을 계산하기 위해 m과 r의 관계를 알아야 한다

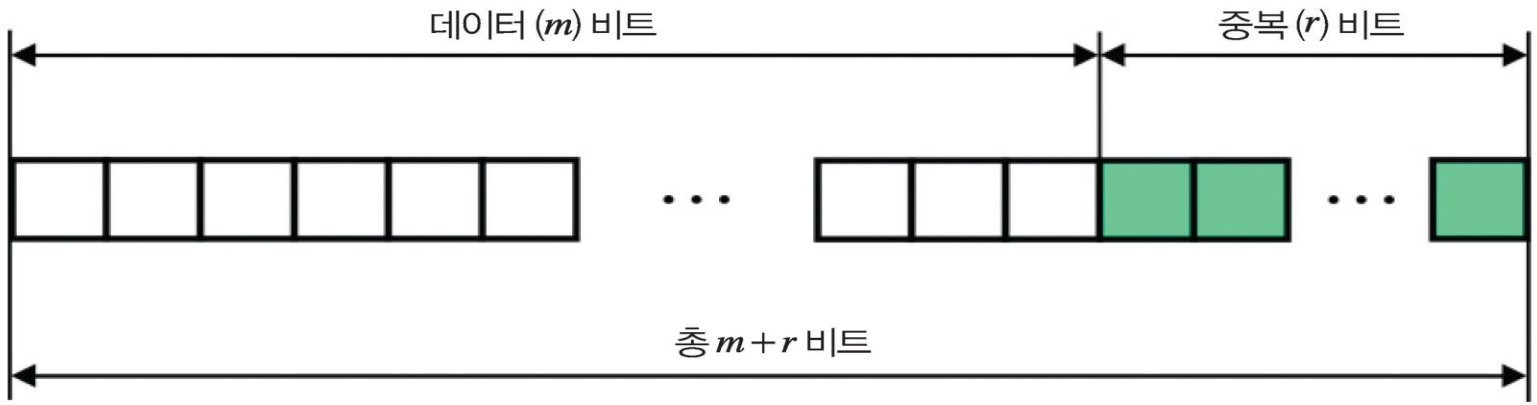

그림 9.18 데이터와 중복 비트

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 전송할 수 있는 비트의 전체 수가 m + r이면 r은 적어도 다음 조건을 만족해야 한다

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

예) 7 비트(ASCII) m에 대해 가장 적은 r의 값은 4이다$2^4 \geq 7 + 4 + 1$

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 데이터와 중복 비트간의 관계

표 9.1 데이터 비트와 중복 비트간의 관계

데이터 비트의 수(m)	중복 비트의 수(r)	총 비트의 수($m + r$)
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 해밍 코드

- R.W. Hamming에 의해 개발
- Hamming 코드에서 중복 비트의 위치

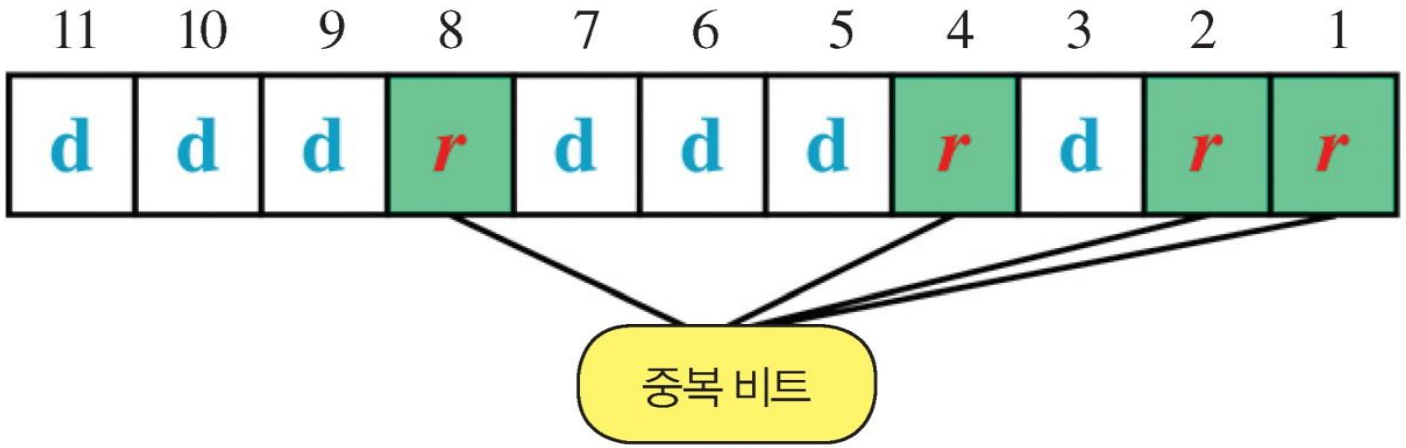

그림 9.19 해밍 코드에서 중복 비트의 위치

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 각 비트는 데이터 비트의 조합을 위한 VRC 비트이다

r_1 = bits 1, 3, 5, 7, 9, 11

r_2 = bits 2, 3, 6, 7, 10, 11

r_4 = bits 4, 5, 6, 7

r_8 = bits 8, 9, 10, 11

❖ 중복 비트 계산

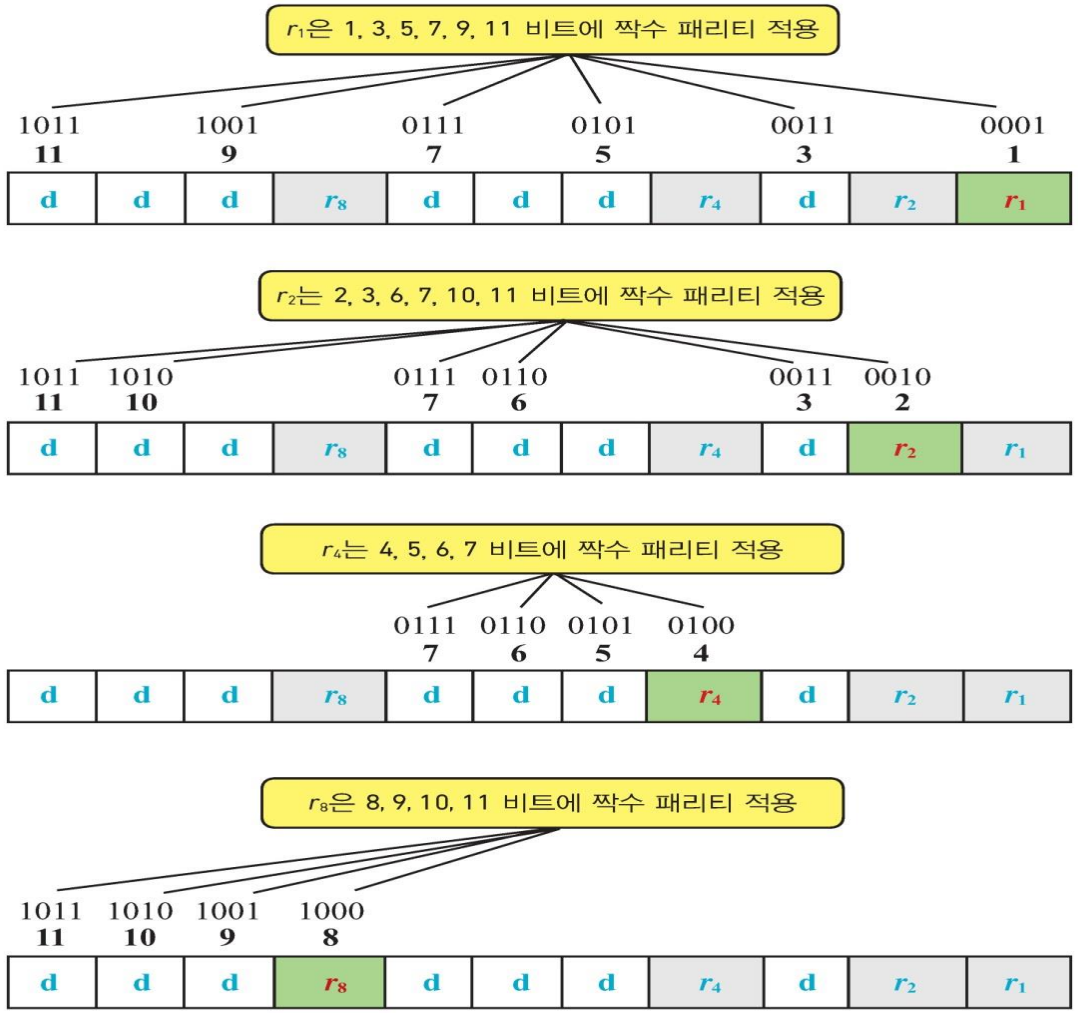

그림 9.20 중복 비트 계산

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 값 계산

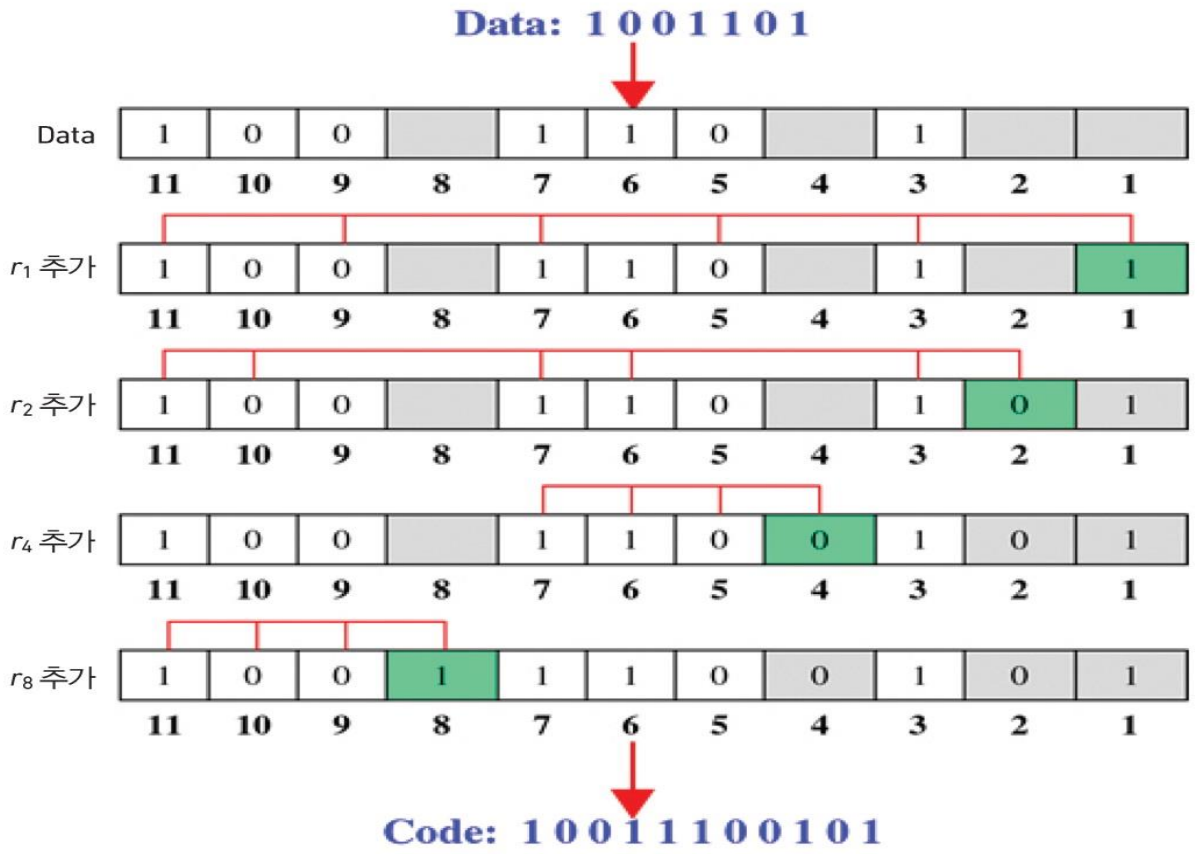

그림 9.21 중복 비트 계산의 예제

❖ 오류 발견과 정정

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

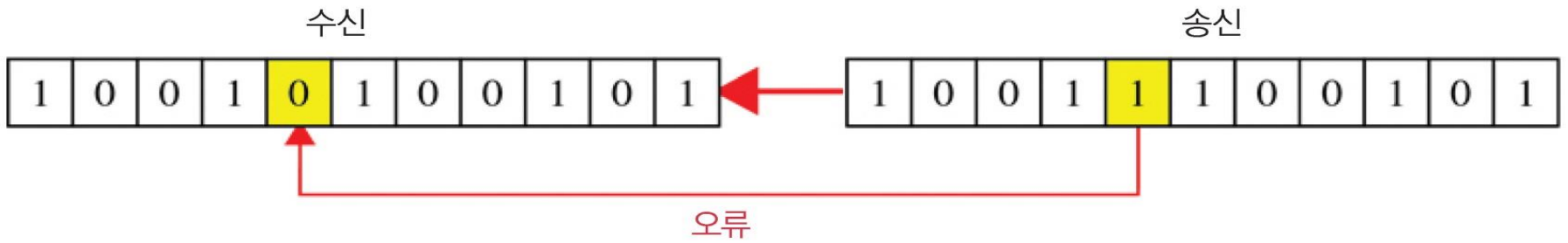

그림 9.22 단일비트 오류

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ Hamming 코드를 이용한 오류발견

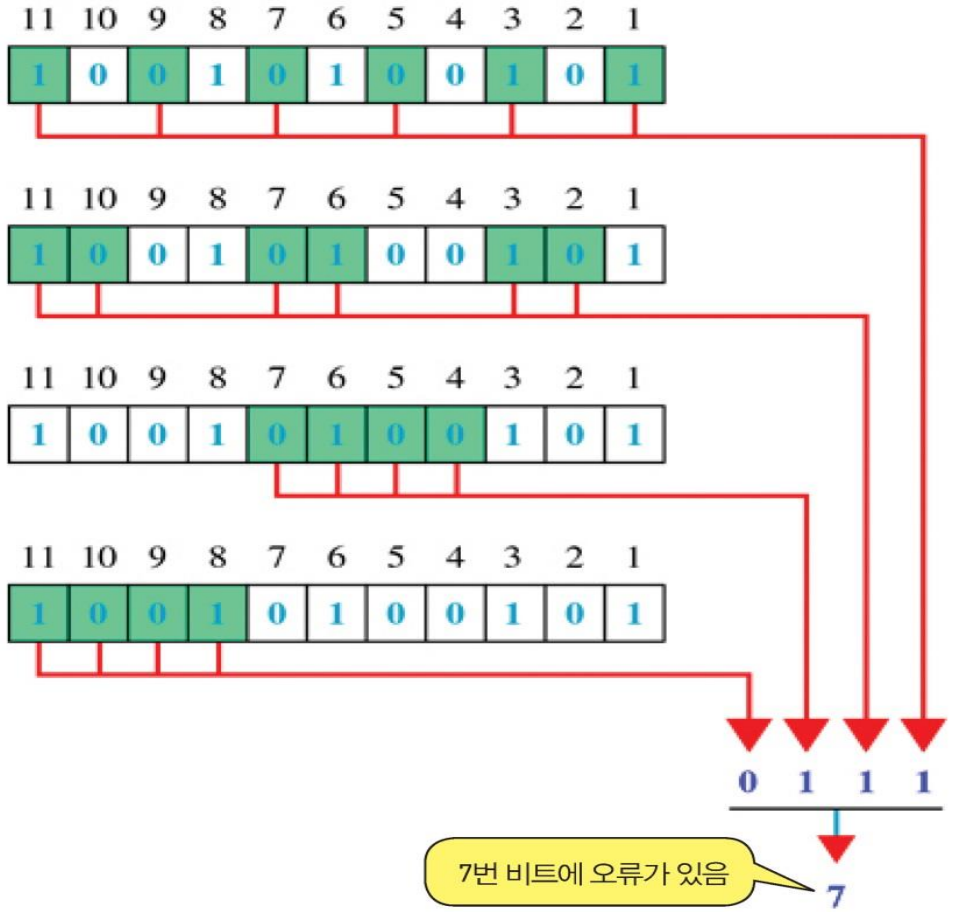

그림 9.23 해밍 코드를 이용한 오류 검출

❖ 오류 종류
❖ 오류 검출
❖ 오류 정정

❖ 다중-비트 오류 정정

- 데이터 비트의 집합을 중복하여 계산되는 중복 비트는 다중 비트 오류를 정정하는 데로 사용할 수 있다

KOREATECH
한국기술교육대학교

THANK YOU