Communication

endian: byte ordering within a word

- Internet
- big endian
- IBM mainframe
- little endian
- Intel

m | i | k | e ≠ e | k | i | 'm'

"mike"

32 bits 00...0 ≡ 32 bit 00...0

507

Error Detection

EVEN

1 bit error

data

Parity bit

Sender

1011 1 - - - - -> 1001 1

receiver

1100 0 → 1101 1

2 bit error

Simple parity scheme detects one bit errors only.

Error Correcting Coding Scheme

Hamming Code

Single bit error correction

? no. of parity bits needed.

no. of data bits : m (4)

no. of parity bits : r (?)

Total bits transmitted : $(m+r)$ bits

Sender $\boxed{m+r}$ bits

- 1 { no error (all $m+r$ bits are correct)
- m { 1st data bit flipped; 2nd " "; 3rd ... "; m^{th} data bit "
- r { 1st parity bit "; r^{th} parity bit flipped

We need $(m+r+1) \Rightarrow$ Possibilities

Receiver

receives $(m+r)$ bits

binary combination of values (absolute no. of possible combinations)

$= 2^{(m+r)}$ possible values.

Receiver has to infer the location of error bit only from parity bit values

The no. of possible combination of values for r parity bits $= 2^r$

To locate the single bit error position,

$$\boxed{(m+r+1) \le 2^r}$$

no. of possible combination with single bit error or no error

no. of possible combination that can be used by the receiver

Parity mode: EVEN

$m = 4$

$r = ?$ (smallest value)

$$(m + r + 1) \le 2^r$$

$r = 2$

$$4 + 2 + 1 \le 2^2$$

$$7 \le 4 \quad \times$$

$r = 3$

$$4 + 3 + 1 \le 2^3$$

$$8 \le 8 \quad \checkmark$$

We need 3 parity bits.

Sender: Data: 1011 (a b c d)

EVEN

$m = 4$

$r = 3$

$m + r = 7$

1	2	3	4	5	6	7
0	1	1	0	0	1	1
P_0	P_1	a	P_2	b	c	d

Parity bits: P_0, P_1, P_2

Positions of parity bits: $2^0\ 2^1\ 2^2$ = 1, 2, 4

$P_0 = \{0, 1, 0, 1\}$ = 1, 3, 5, 7

$P_1 = \{1, 1, 1, 1\}$ = 2, 3, 6, 7

$P_2 = \{0, 0, 1, 1\}$ = 4, 5, 6, 7

Position	P_2 (2^2)	P_1 (2^1)	P_0 (2^0)
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Receiver: $m = 4$, EVEN, $r = ?$ 3 P_0 P_1 P_2 (2^0 2^1 2^2 = 1 2 4)

1	2	3	4	5	6	7
0	1	1	0	1	1	1

Parity test

$\times\ P_0 = \{0, 1, 1, 1\}$ = 1, 3, 5, 7

$\checkmark\ P_1 = \{1, 1, 1, 1\}$ = 2, 3, 6, 7

$\times\ P_2 = \{0, 1, 1, 1\}$ = 4, 5, 6, 7

Error bit position = Sum of the positions of the parity bits that failed the parity test.

= Position(P_0) + Position(P_2) = 1 + 4 = 5

Sender: Data: 1011 (a b c d)

EVEN

$m = 4$

$r = 3$

$m + r = 7$

SEC + DED

Single Error Correction

Double Error Detection

DED Parity

0	1	1	0	0	1	1	
1	2	3	4	5	6	7	8
P_0	P_1	a	P_2	b	c	d	

Parity bits: P_0, P_1, P_2

Positions of Parity bits: $2^0, 2^1, 2^2$

1, 2, 4

$P_0 = \{1, 3, 5, 7\}$ (0 1 0 1)

$P_1 = \{2, 3, 6, 7\}$ (1 1 1 1)

$P_2 = \{4, 5, 6, 7\}$ (0 0 1 1)

	4	2	1
	2^2	2^1	2^0
Position	P_2	P_1	P_0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

DED Parity $= \{1, 2, 3, 4, 5, 6, 7, 8\}$ = 0 1 1 0 0 1 1 ? ← 0

Transmitted bits: 0 1 1 0 0 1 1 0 (1 2 3 4 5 6 7 8)

Receiver: $m = 4$, EVEN, $r = ?$ 3

0	1	0	0	1	1	1	0
1	2	3	4	5	6	7	8

P_0 P_1 P_2

2^0 2^1 2^2

1 2 4

Parity test for SEC (on the left m+r bits)

✓ $P_0 = \{1, 3, 5, 7\}$ (0 0 1 1)

✗ $P_1 = \{2, 3, 6, 7\}$ (1 0 1 1)

✗ $P_2 = \{4, 5, 6, 7\}$ (0 1 1 1)

Error bit position = Sum of the positions of the parity bits that failed the parity test

$= \text{Position}(P_1) + \text{Position}(P_2) = 2 + 4 = 6$

+DED

Double Error Detection

	4	2	1
	2^2	2^1	2^0
Position	P_2	P_1	P_0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

DED Parity $= \{1, 2, 3, 4, 5, 6, 7, 8\}$ = 0 1 1 0 0 1 1 ? ← 0

Transmitted bits: 0 1 1 0 0 1 1 0 (1 2 3 4 5 6 7 8)

Receiver

After performing SEC process

Value: 0 1 0 0 1 0 1 0

1 2 3 4 5 6 7 8

Perform simple parity test on the revised (m+r+1) bits

Parity test fails (no. of ones is three)

Hence there is a double bit error.

Receiver (SEC+DED)

Perform SEC on the left most (m+r) bits.

If the error position > (m+r)

there is a double bit error

else

- correct the error bit
- test the revised (m+r+1) bits for a simple parity test

if the parity test fails

There is a double bit error.

else

No double bit error

remove all parity bits and extract the original data bits