

Fehlererkennung und -korrektur auf Bitebene

Zum Speichern und Übertragen von Daten, werden diese binär codiert. Bei der Verarbeitung und Übertragung von Daten kann es zu technischen Fehlern kommen, die unabhängig von der Korrektheit der Algorithmen auftreten und dazu führen, dass sich einzelne Bits umkehren. Das heißt, aus einer 0 wird eine 1 oder andersherum.

Aufgabe 1: Das Wort *Maus* wird mit dem ASCII-Code binär codiert und übertragen:

0100 1101 0110 0001 0111 0101 0111 0011

Das achte Bit wird fehlerhaft übertragen, so dass der Empfänger eine andere Bitfolge erhält:

0100 1101 0110 0001 0111 0101 0111 0011

Diese wird decodiert zu dem Wort *Laus*.

Fehler lassen sich bei der Datenübertragung nicht vermeiden. Sammeln Sie daher Ideen, wie man die Codierung der Daten verändern bzw. erweitern könnte, so dass der Empfänger den Fehler erkennen oder vielleicht sogar korrigieren kann.

Das Paritätsbit

Um Fehler bei der Datenspeicherung oder Übertragung zu erkennen, können ein oder mehrere Prüfbits ergänzt werden. Eine einfache Form eines Prüfbits, ist ein Paritätsbit. Ein Paritätsbit wird für jeden Datenblock einer festen Länge von z. B. 8 Bit erstellt. Beim Erstellen des Prüfbits unterscheidet man zwischen gerader und ungerader Parität. Bei **gerader Parität** wird das Paritätsbit so gewählt, dass die Anzahl der 1en inklusive Paritätsbit gerade ist. Bei **ungerader Parität** wird das Paritätsbit so gewählt, dass die Anzahl der 1en inklusive Paritätsbit ungerade ist.

Aufgabe 2: Ergänzen Sie in der Tabelle in Abbildung 1 jeweils das Paritätsbit für gerade bzw. für ungerade Parität.

Datenbits	Paritätsbit	
	bei gerader Parität	bei ungerader Parität
0101 0101	0	1
1111 1111		
0000 0001		
0010 1001		

Abbildung 1: Paritätsbit für einen Datenblock von 8 Bit

Aufgabe 3: Bei der Übertragung von Daten wird folgende Bitfolge empfangen:

0011 0101 0 1101 0011 1 0011 1010 1 0111 0000 1

Vor dem Versenden wurde jedem Datenblock aus 8 Bit ein Paritätsbit mit gerader Parität angehängt. Überprüfen Sie, ob bei der Übertragung der Daten ein Fehler aufgetreten ist.

Aufgabe 4: Begründen Sie, dass ...

- mithilfe eines Paritätsbits ein fehlerhaftes Bit zwar erkannt, aber nicht korrigiert werden kann.
- mithilfe eines Paritätsbits nicht erkannt werden kann, dass zwei Bits in einem Datenblock falsch übertragen wurden.
- mithilfe eines Paritätsbits nur eine ungerade Anzahl an Fehlern in einem Datenblock erkannt werden kann.

Aufgabe 5: Stellen Sie eine Vermutung auf, weshalb Paritätsbits in der Regel für einzelne Datenblöcke und nicht für die gesamte Bitfolge erstellt werden.

Aufgabe 6: Untersuchen Sie das Erstellen eines Prüfbits und das Erstellen einer Prüfziffer nach dem Luhn-Algorithmus auf Gemeinsamkeiten und Unterschiede.

Aufgabe 7: In der ursprünglichen Version umfasst der ASCII-Code 128 Zeichen, die mit 7 Bit codiert werden. Das achte Bit eines Byte wurde später zur Codierung von Sonderzeichen wie z. B. Ä, Ö usw. in der deutschen Sprache verwendet. Alternativ kann das achte Bit als Paritätsbit verwendet werden. Erläutern Sie an diesem Beispiel, dass ein Code, der eine Fehlererkennung ermöglicht, nicht optimal hinsichtlich der zur Codierung benötigten Datenmenge sein kann.

Fehlerkorrektur

Aufgabe 8: In Abbildung 2 wurden 8 Byte zeilenweise in eine Matrix eingetragen (blau hinterlegt). Anschließend wurde für jede Zeile und für jede Spalte ein Paritätsbit mit gerader Parität erstellt (grün hinterlegt). Bei der Übertragung der Matrix wurde ein Bit falsch übertragen.

a) Analysieren Sie, ob Sie feststellen können, welches das fehlerhafte Bit ist.

1	1	0	1	1	0	1	1	0
0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	1
1	0	1	1	0	1	1	0	1
0	1	1	1	1	0	0	1	0
0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1
0	1	0	0	1	0	0	1	1
1	0	0	1	1	0	1	0	0

Abbildung 2: Matrix mit einem fehlerhaften Bit

- b) Erläutern Sie, wie mithilfe des in Abbildung 2 dargestellten Verfahrens 1-Bit-Fehler korrigiert werden können.
- c) Zeigen Sie, dass 2-Bit-Fehler nicht zuverlässig korrigiert werden können.

Implementierung

Aufgabe 9: Erstellen Sie ein Programm, das

- a) zu einer Bitfolge, die als Zeichenkette eingegeben wird, das passende Paritätsbit erstellt.
- b) für eine Bitfolge mit Paritätsbit prüft, ob ein Fehler enthalten ist.

Hinweis: Legen Sie fest, ob Sie gerade oder ungerade Parität verwenden oder lassen Sie den Anwender entscheiden. Gestalten Sie die Ein- und Ausgabe des Programms nach Ihren Vorstellungen.

Aufgabe 10:

- a) Begründen Sie, dass der Inhalt einer Matrix wie in Abbildung 2 als Zeichenkette eingegeben werden kann.
- b) Erstellen Sie ein Programm, das den Inhalt einer Matrix wie in Abbildung 2 in Form einer Zeichenkette als Eingabe erhält, auf Fehler überprüft und diese ggf. korrigiert. Die korrigierte Zeichenkette soll anschließend ausgegeben werden. Sie können voraussetzen, dass in der Eingabe maximal 1 Bit fehlerhaft ist.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Von der Lizenz ausgenommen ist das InfSII-Logo.