Hamming-Code

Ismail Al Shuaybi

Wintersemester 2022/2023

Inhalt der Präsentation

Definition

Was ist der Hamming-Code?

1. Fehlerkorrektursystem mit zwei wesentlichen Aufgaben:
   1. Fehlererkennung
   2. Fehlerkorrektur
2. Benennung nach seinem Erfinder Richard W. Hamming
3. Anwendungsbereiche:
   1. digitale Signalverarbeitung
   2. Nachrichtentechnik

Richard Hamming

Quelle: https://en.wikipedia.org/wiki/Richard_Hamming

Grund für die Anwendung des Hamming-Codes

Mögliche Datenbeschädigung in den folgenden Fällen:

- Datenübertragung
- Datenspeicherung

Dies kann in Form von Bit-Flips erfolgen.

Wie können solche Beschädigungen vermieden werden?

Zur Vermeidung kommen Fehlerkorrekturcodes wie der Hamming-Code zum Einsatz.
Das Herausfinden von Fehlern erfolgt durch das Hinzufügen von **Paritätsbits** zu den Daten.

Eigenschaften des Hamming-Codes

Stärken des Hamming-Codes

Der Hamming-Code kann:

- Einzelbitfehler korrigieren
- Zweibitfehler erkennen

Was ist, wenn zwei Bits fehlerhaft sind?

- dann sind beide Fehler als Einzelbitfehler zu betrachten
- dann wird ein zusätzliches Gesamtparitätsbit hinzufügt
- führt zur zuverlässigen Erkennung der Fehler in zwei Bits.

Solche Fälle sind als Einzelfehlerkorrektur/Doppelfehlererkennung (SECDED) bekannt.

Funktionsweise von Paritätsbits

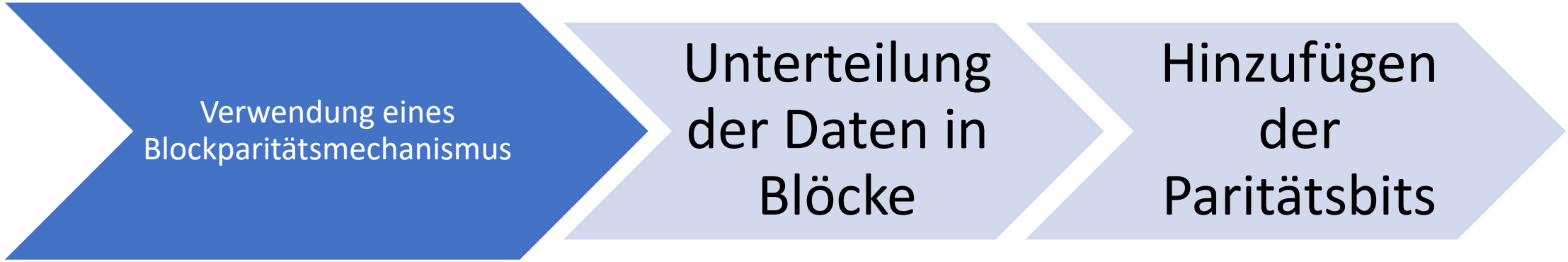
- Das Hinzufügen von Paritätsbits erfolgt durch
  - das Anhängen von Paritätsbits an eine Bitfolge
    - als LSB
    - immer an die Stellen mit der Potenz 2^x
- Davon gibt es zwei Arten:
  - even-parity
  - odd-parity

Inhalt der Präsentation

Anzahl der Paritätsbits pro Codewort bei einem Ein-Bit-Fehler

Zahl der Datenbits m	Zahl der Prüfbits k	Bits insgesamt: $m + k$
1	2	$m + 2$
2 bis 4	3	$m + 3$
5 bis 11	4	$m + 4$
12 bis 26	5	$m + 5$
27 bis 57	6	$m + 6$
...	...	...

Funktionsweise des Hamming-Codes



Schritte für die Erstellung des Hamming-Codes

Es sei u ein Nachrichtenwort mit den Bits u0 u1 u2 und u3. Um den Hamming-Code daraus zu erzeugen, sollte man wie folgt vorgehen:

1. Berechnung der benötigten Prüfbitanzahl.
2. Darstellung der Summe der Prüfbits und Nachrichtenbits durch einen Vektor V.

V0	V1	V2	V3	V4	V5	V6

3. Nummerierung der Positionen des Vektors.

V0	V1	V2	V3	V4	V5	V6
1	2	3	4	5	6	7

4. Reservierung der Positionen an den 2^n-Stellen für die Prüfbits.

V0	**V1**	V2	**V3**	V4	V5	V6
1	**2**	3	**4**	5	6	7

5. Markierung der restlichen Positionen für die Nachrichtenbits.

		u0		u1	u2	u3
V0	**V1**	V2	**V3**	V4	V5	V6
1	**2**	3	**4**	5	6	7

Schritte für die Erstellung des Hamming-Codes

6. Berechnung der Paritätsbits.

$$
\begin{aligned}
&(u0 * bin(3)) \\
&(u1 * bin(5)) \\
&(u2 * bin(6)) \\
\oplus\ &(u3 * bin(7)) \\
&\text{----------} \\
&x1\ x2\ x3
\end{aligned}
$$

Hinzufügen der Paritätsbits an den 2^n-Stellen des Vektors.

Am Ende erhält man ein Hamming-Codeword bestehend aus folgender Datensequenz:

x1 x2 u0 x3 u1 u2 u3

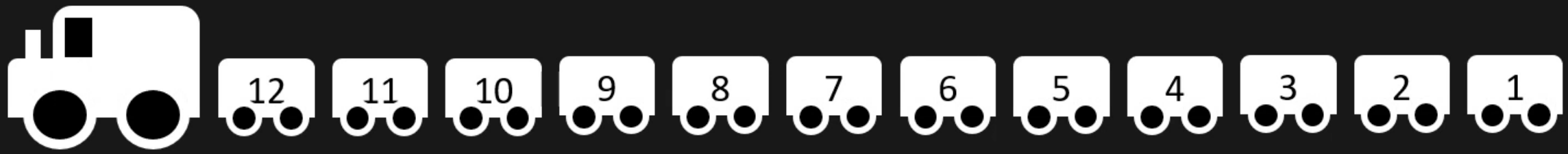
Hamming-Code berechnen

10110100

Hamming-Code berechnen

10110100

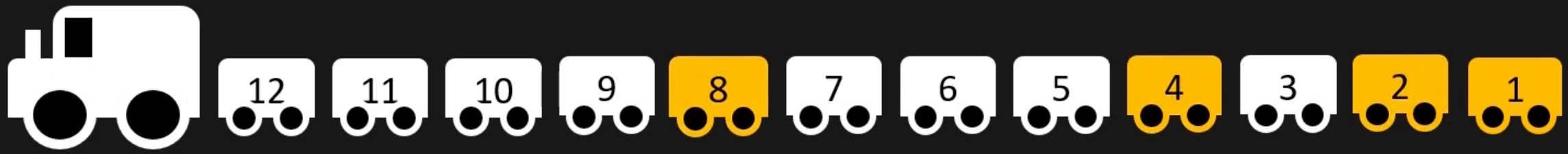
Welche Waggons sollen für die Paritätsbits reserviert werden?



$$2^n$$

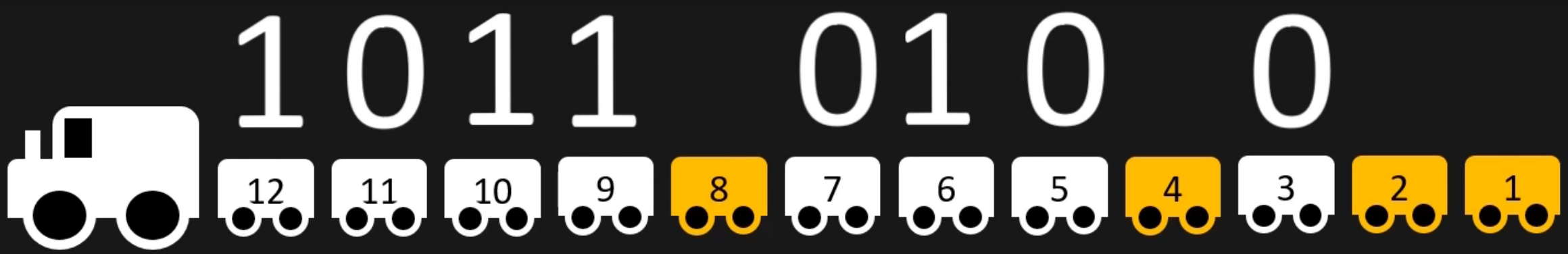
Hamming-Code berechnen

10110100

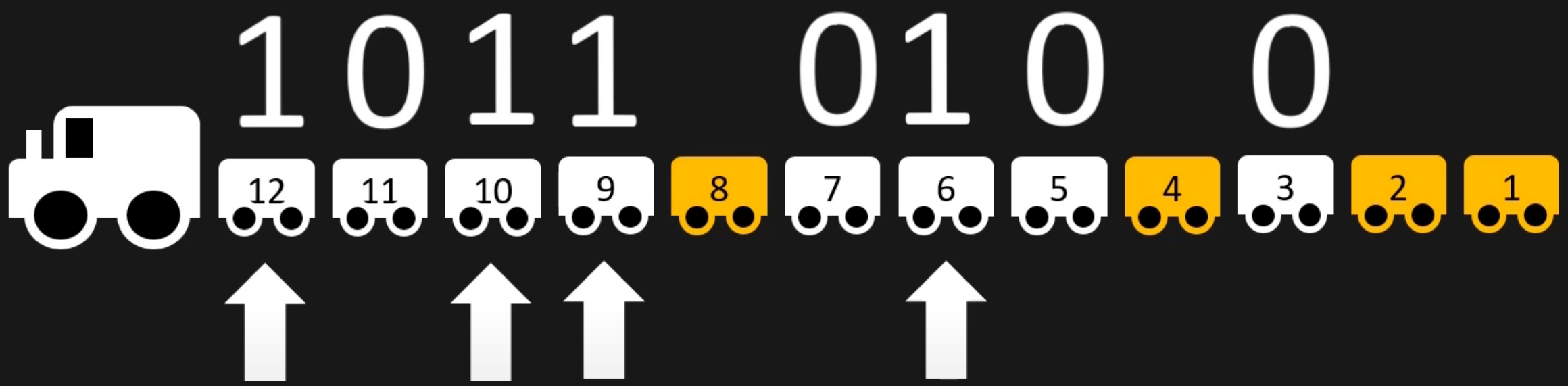


Hamming-Code berechnen

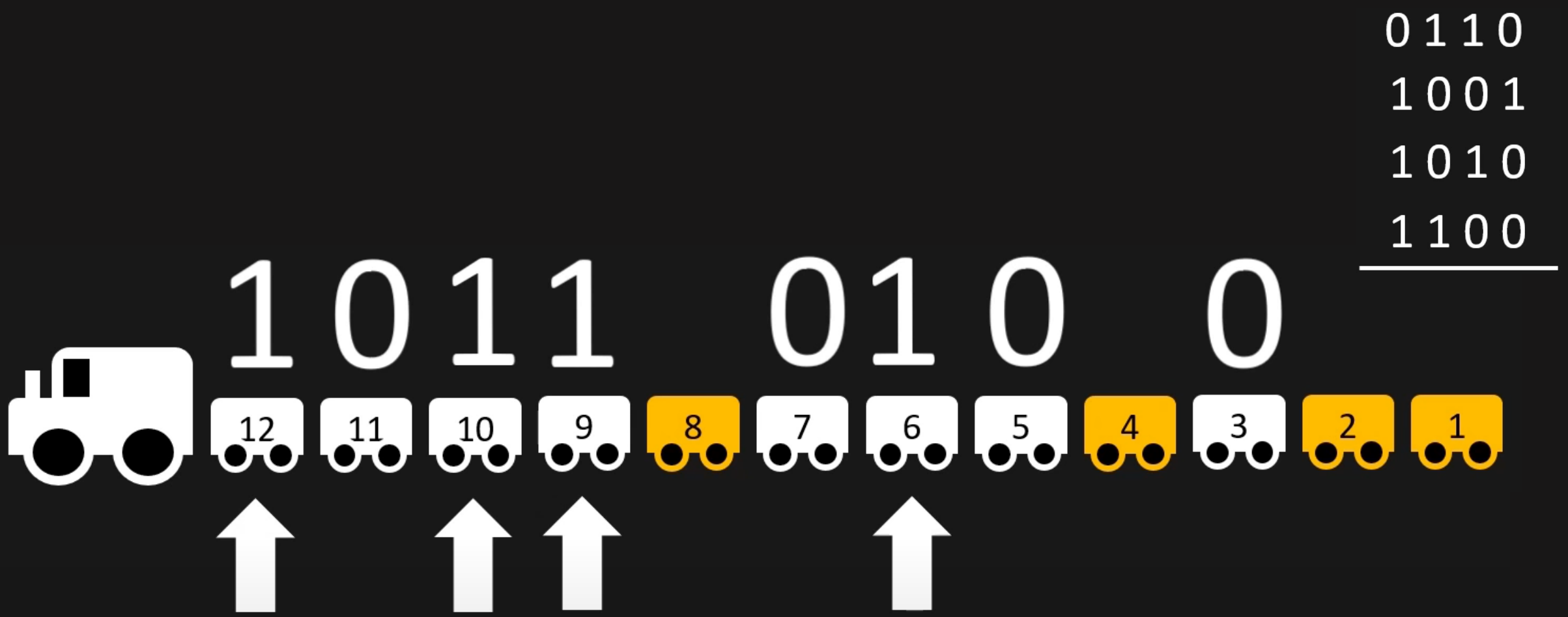
10110100



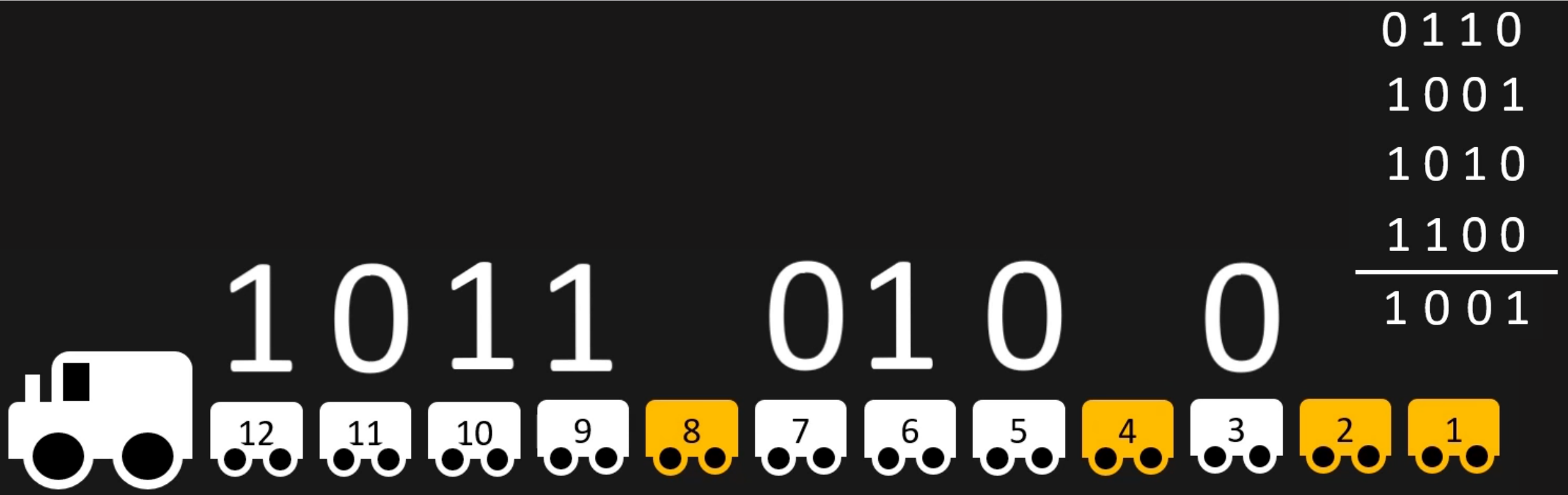
Hamming-Code berechnen



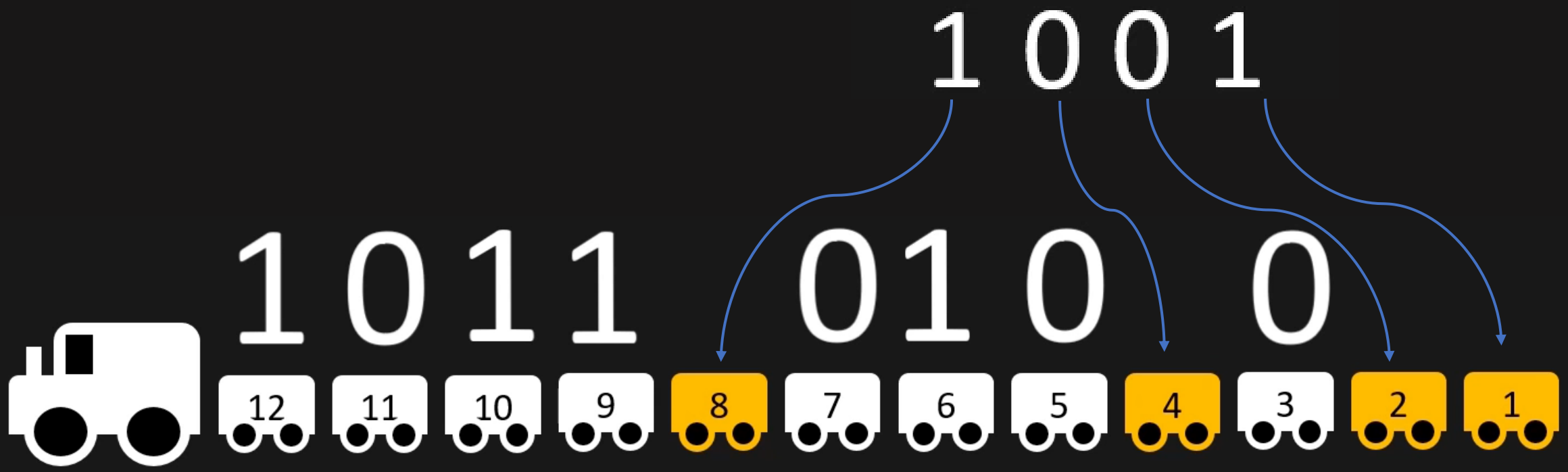
Hamming-Code berechnen



Hamming-Code berechnen



Hamming-Code berechnen



Hamming-Code berechnen

Inhalt der Präsentation

Fehlererkennung und Korrektur

Es wurde folgendes Nachrichtenwort empfangen 0011001.

Fehlererkennung:

- Summieren der binären Werte der Positionsnummern, die 1 enthalten, ohne Übertrag (XOR-Operation).

$$
\begin{array}{r}
0\ 1\ 1 \\
1\ 0\ 0 \\
\oplus\ 1\ 1\ 1 \\
\hline
0\ 0\ 0 \Rightarrow \text{fehlerlose Übertragung}
\end{array}
$$

Fehlererkennung und Korrektur

Es wurde folgendes Nachrichtenwort empfangen 0011001.

Fehlererkennung:

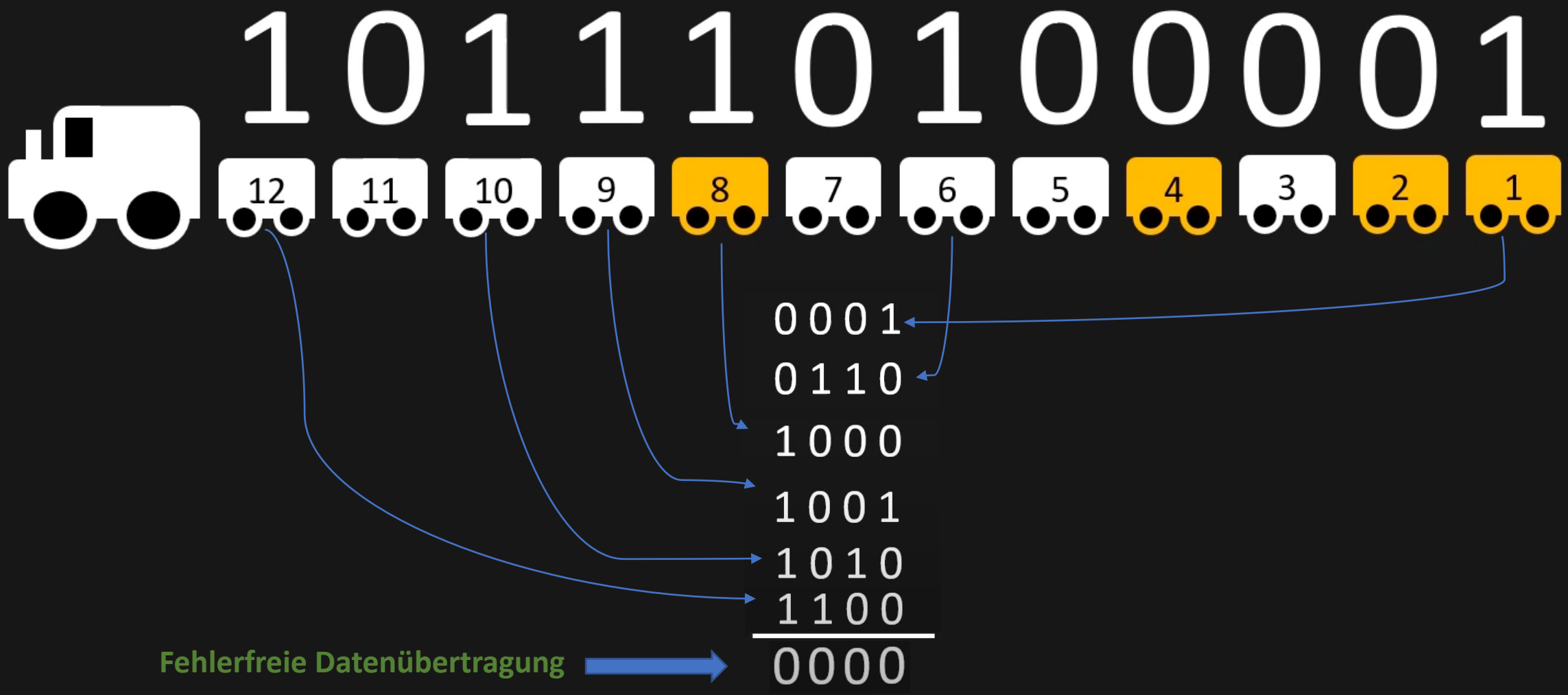
- Summieren der binären Werten der Positionsnummern, die 1 enthalten, ohne Übertrag (XOR-Operation).

$$
\begin{array}{rr}
 & 0\ 1\ 0 \\
 & 0\ 1\ 1 \\
 & 1\ 0\ 0 \\
\oplus & 1\ 1\ 1 \\
\hline
 & 0\ 1\ 0
\end{array}
$$

0 1 0 => Fehler an der zweiten Stelle

Hamming-Code berechnen

Fehlererkennung mittels XOR-Operator



Inhalt der Präsentation

Hamming-Code berechnen

Überprüfung, ob die Nachricht richtig angekommen ist.

Inhalt der Präsentation

Generatormatrix

- Erstellung eines (7,4)-Hamming-Codes für Datenbits mit der Länge 4 durch die Generatormatrix.

$$\mathbf{G}_{4\times 7} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Beispiel

- Die Erzeugung des (7,4)-Hamming-Codes des Nachrichtenwortes 1010 erfolgt durch die Formel: v = u * G

$$\begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Codierung eines Kontrollmatrix

- Ordnung aller 7 Kombinationen der Prüfbits p_i.
- Nummerierung dieser Kombinationen.

Fehlerhafte Stelle	Prübits:	p_1	p_2	p_3
1		0	0	1
2		0	1	0
3		0	1	1
4	$\boldsymbol{M} =$	1	0	0
5		1	0	1
6		1	1	0
7		1	1	1

Anwendung der Kontrollmatrix

- Überprüfung auf Fehlerlosigkeit des Nachrichtenworts durch die Multiplikation eines Nachrichtenbits mit der Kontrollmatrix.

$$\boldsymbol{s} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}$$

Literatur

- Einführung in die Information und Codierungstheorie
- Grundkurs Informatik: 7. Auflage
- Vorlesung 04: Tamim Asfour
- Berechnung des Hamming-Codes
- Definition des Hamming-Codes