Digitaltechnik

Kapitel 2, Digitale Codierung von Informationen

Prof. Dr.-Ing. M. Winzker

(Stand: 20.03.2019)

Zahlendarstellungen und Codes

- Mit Digitalschaltungen werden nicht nur 0/1-Informationen verarbeitet, sondern allgemein Daten verschiedenster Art
- Diese Daten werden durch mehrere binäre Signale („Bits“) dargestellt
- Die 0/1-Kombinationen sind **Codewörter**
- Die Zuordnung zwischen Codewort und Bedeutung ist der **Code**
  - „Code“ im Sinne der Elektronik (Nachrichtentechnik, Kommunikationstechnik) meint normalerweise keine Geheimhaltung

- Einen Code zur Darstellung und **Berechnung** von Zahlen, wird als **Zahlendarstellung** bezeichnet
  - Die Betonung der Berechnung ist nötig, da manche praktisch sinnvollen Codes für Zahlen sich nicht für Berechnungen eignen

Eigenschaften von Codes

Codes existieren mit verschiedenen Eigenschaften für verschiedene Anwendungen:

- Für die **Datenverarbeitung** sollen Codes einfach zu handhaben sein
  - Dies kann unter anderem durch eine **feste Codewortlänge** erreicht werden
  - <u>Beispiel:</u> ASCII-Zeichen haben eine feste Codewortlänge von 7 Codestellen
- Zur **Datenspeicherung** und **Datenübertragung** sollen Codes wenig Platz (wenige binäre Codestellen) einnehmen
  - Hierzu können Codewörter mit **variabler Codewortlänge** eingesetzt werden
  - <u>Beispiel:</u> Morse-Code nutzt unterschiedliche Anzahl an Strichen und Punkten zur Darstellung eines Zeichens, je nach Häufigkeit in (englischen) Texten

    Z.B.: ‚E‘ = ‚•‘ ; ‚T‘ = ‚–‘ ; ‚M‘ = ‚– –‘; ‚Q‘ = ‚– – • –‘; …
- Zur **Datenübertragung** sollen Übertragungsfehler erkannt und/oder korrigiert werden
  - <u>Beispiel:</u> Parity-Bit zur einfachen Fehlererkennung
- Zur **Geheimhaltung** soll Information verschlüsselt werden
  - <u>Beispiel:</u> „Ceasar Cypher“, d.h. einfaches Vertauschen von Codewörtern

Zweierpotenzen

- Mit 2 Bits können 4 Möglichkeiten dargestellt werden: „00“, „01“, „10“, „11“
  - Mit jedem weiteren Bit verdoppeln sich die Möglichkeiten
- Allgemein gilt: Mit n Stellen können 2^n Codewörter gebildet werden
  - Die Anzahl an Stellen wird auch als **Wortbreite** bezeichnet

- Die Zweierpotenz für 0 bis 10:

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Die Werte können leicht hergeleitet werden:

- Für n=1 ist $2^n=2$, alle weiteren Werte ergeben sich durch verdoppeln
- Häufig verwendet wird:
  - $2^8=256$ (8 Bit sind 1 Byte)
  - $2^{10} \approx 1000$

Für höhere Zweierpotenzen kann der Exponent aufgeteilt werden: $2^{m+n} = 2^m \bullet 2^n$

- **Beispiel:** $2^{16} = 2^{10+6} = 2^{10} \bullet 2^6 = 2^m \bullet 2^n \approx 1000 \bullet 64 = 64\,000$

Zahlendarstellung Dualzahl

- Die Stellen einer Dualzahl haben eine feste **Stellenwertigkeit**:
  - Eine N-stellige Zahl hat die Stellen $N\text{-}1$ bis 0; die Stelle 0 ist LSB
    - o LSB: „Least Significant Bit" (niedrigstwertiges Bit)
    - o MSB: „Most Significant Bit" (höchstwertiges Bit)
  - Die Stelle n hat die Wertigkeit 2^n
  - Beispiel: 8 bit Dualzahl $0010\ 0110_2$
    - o Stelle 5, 2 und 1 entsprechen den Werten 32, 4 und 2 und ergeben die Dezimalzahl 38_{10}

- Der Wertebereich ist $[0\ ;\ 2^N\text{-}1]$
  - Beispiel: 8 bit Dualzahlen haben den Wertebereich [0;255]

- Umwandlung von Dezimalzahlen ins Dualzahlen:
  - Fortwährende Division des Betrags durch 2
  - Die Divisionsreste ergeben die Binärzahl, angefangen vom LSB

Beispiel: $41_{10} = 101001_2$

41:2 = 20 Rest 1 (LSB)
20:2 = 10 Rest 0
10:2 = 5 Rest 0
5:2 = 2 Rest 1
2:2 = 1 Rest 0
1:2 = 0 Rest 1

Zahlendarstellung Zweierkomplement

- Darstellung im Zweierkomplement (2C):
  - Repräsentation positiver und negativer Zahlen
- **Feste Stellenwertigkeit** wie bei Dualzahlen
  - Die Stelle n hat (wie bei Dualzahlen) die Wertigkeit 2^n
  - Ausnahme: Die Stelle N-1 ist das Vorzeichen und hat die Wertigkeit -2^{N-1}
  - Beispiel: 8 bit Zweierkomplementzahl $1010\ 0110_{2C}$
    - Stelle 7 zeigt an, dass die Zahl negativ ist und hat den Wert -128
    - Stelle 5, 2 und 1 entsprechen den Werten 32, 4 und 2
    - Insgesamt ergibt sich die Dezimalzahl -90_{10}
- Der Wertebereich ist $[-2^{N-1}\ ;\ 2^{N-1}-1]$
  - Der Wertebereich ist unsymmetrisch (wegen der Null)
  - Beispiel: 8 bit Zweierkomplementzahlen haben den Wertebereich [-128;127]

Achtung: In der Digitaltechnik (und Informatik) wird immer ab 0 gezählt

- Aufpassen bei sprachlichen Missverständnissen: Stelle 1 ist die zweite Stelle

Umwandlung zum Zweierkomplement

Für positive Zahl

- Erweiterung um das Vorzeichenbit ‚0'
  ➔ Eine n bit Dualzahl benötigt im Zweierkomplement n+1 bit

Für negative Zahl

- Bestimmung der korrespondierenden **positiven Dualzahl**
- **Invertierung aller Stellen** der positiven Dualzahl und **Addition von 1**
  - Beispiel: Dezimalzahl -38_{10}
    - o Dualzahl: $0010\ 0110_2$ ($+38_{10}$ = positive Dualzahl)
    - o Invertierung: 1101 1001
    - o Addition von 1: $1101\ 1010_{2C}$ ← Ergebnis

Rückwandlung aus dem Zweierkomplement

Wertebestimmung einer Zahl im Zweierkomplement

Auswertung des Vorzeichens

- **‚0‘:** Positive Zahl
  - Wert entsprechend der Dualzahl
- **‚1‘:** Negative Zahl
  - Rückwandlung in Dualzahl:
    - Subtraktion von 1 und Invertierung
  - Wert ist negativer Wert der Dualzahl

Vereinfachte Berechnung für negative Zahl

- Subtraktion von 1 und Invertierung ist „lästig“ wegen Subtraktion
- Einfacher: Erst Invertierung, dann wieder Addition
  - Beispiel: Zweierkomplement -38_{10} (siehe oben)
    - Zweierkompl.: $1101\ 1010_{2C}$
    - Invertierung: $0010\ 0101$
    - Addition von 1: $0010\ 0110_2$ ← Ergebnis: 38_{10}

Rechenoperationen im Zweierkomplement

- Addition:
  - Stellenweise Addition mit Übertrag in nächste Stelle
  - Bereichsüberschreitung, falls beide Summanden gleiches Vorzeichen haben und die Summe ein anderes Vorzeichen ergibt
    - Bereichsüberschreitung kann Überlauf („Overflow“) oder Unterlauf („Underflow“) sein
- Subtraktion:
  - Bildung des Komplements und Addition
- Multiplikation und Division:
  - Stellenweise Berechnung, ähnlich der Dezimalrechnung
  - Besondere Behandlung der Vorzeichenstellen
  - Details in Literatur

Achtung: Für alle Rechenoperationen dürfen nur gleiche Zahlendarstellungen miteinander kombiniert werden

- Ansonsten Dualzahlen in Zweierkomplement umwandeln

Wertebereiche und Wortbreiten

- Bei Addition zweier Zahlen kann das Ergebnis den Wertebereich der Summanden überschreiten
- Die Summe muss darum normalerweise eine größere Wortbreite haben
  - Beispiel: Zwei 8 bit Zahlen (Wertebereich [0;255]) können den Wertebereich [0;510] ergeben, benötigen also 9 bit
- Alternative: Die Addition hat eine Überlaufbegrenzung
  - Wird in der Signalverarbeitung verwendet, z.B. Multimedia-Befehle einer CPU

- Erweiterung der Wortbreite:
  - Dualzahlen: Vordere Stellen werden mit ‚0‘ aufgefüllt
  - Zweierkomplement: Vordere Stellen werden mit Vorzeichen (MSB) aufgefüllt
- Bei der Addition kann dadurch kein Überlauf entstehen
- Achtung: Bei Zweierkomplement kann ein **Übertrag** entstehen, der **entfällt**

  ➔ Beispiel auf nächster Seite

Beispiel: Überlauf im Zweierkomplement

- Addition zweier 8 bit Zahlen im Zweierkomplement
- Ergebnis muss 9 bit Zahl sein

- Addition von: -38 ($1101\ 1010_{2C}$) + 43 ($0010\ 1011_{2C}$)
- Erweiterung der Summanden auf 9 Bit durch Auffüllen mit Vorzeichen (MSB)
  - -38 = **1** $1101\ 1010_{2C}$
  - 43 = **0** $0010\ 1011_{2C}$
  - Der vordere Überlauf entfällt!

vorderer Übertrag entfällt

```
      1   1 1 0 1   1 0 1 0
      0   0 0 1 0   1 0 1 1
    1 1   1 1 1 1   0 1 0      ← Übertrag
    ------------------------
      0   0 0 0 0   0 1 0 1
```

Begründung:

a) Das ist eine Rechenregel ☺

b) Die Summanden könnten theoretisch unendlich fortgesetzt werden

```
… 1 1 1 1   1 1 0 1   1 0 1 0
… 0 0 0 0   0 0 1 0   1 0 1 1
… 1 1 1 1   1 1 1 1   0 1 0
  ----------------------------
… 0 0 0 0   0 0 0 0   0 1 0 1
```

Anwendungen für Dualzahl und Zweierkomplement

Anwendungen der Signalverarbeitung nutzen fast immer Dualzahl und Zweierkomplement

- PC Grafik: Drei Anteile, Rot, Grün, Blau, jeweils 8stellige Dualzahl (8 bit)
- CD Audio: 16stellige Zweierkomplementzahl (16 bit)

Grund

- Der Wertebereich ist beschränkt und vorab bekannt
- Bei Überschreiten des Wertebereichs wird auf den Maximal-, Minimalwert begrenzt

Für Anwendungen mit großem, oft unbekannten Wertebereich ist eine **Gleitkommadarstellung** sinnvoll

- Eine Zahl wird aufgeteilt in Vorzeichen, Wert, Multiplikationsfaktor
  - Ähnlich der Darstellung im Taschenrechner: - 3.4563 E -17
- Anwendung
  - Allgemeine Arithmetik in CPU und Signalprozessor

ASCII-Code

- Ein Beispiel für einen allgemeinen Code ist der **ASCII-Code**
  - Buchstaben, Ziffern, Zeichen und Steuerbefehle werden durch eine 7 bit Dualzahl codiert, z.B.:
    - ‚A‘ = 0x41
    - ‚B‘ = 0x42
    - ‚a‘ = 0x61
    - ‚=‘ = 0x3D
    - ‚TAB‘ = 0x09
    - ‚BEL‘ = 0x07
- Da keine Umlaute und internationalen Sonderzeichen (z.B. ‚¿‘)dargestellt werden können, gibt es verschiedene Erweiterungen auf 8 bit
- **Unicode** ist ein über die ASCII-Zeichen herausgehender Standard, der nicht nur Symbole die westlichen Sprachen umfasst, sondern für „alle lebenden Sprachen“
  - Es existieren Unicode Symbole u.a. für Arabisch, Chinesisch („traditionell“ und „vereinfacht“), Hindi, Hebräisch, Persisch

BCD-Code

- Im **Binary Coded Decimal** werden die 10 Ziffern durch einen 4 bit Code dargestellt
  - Die Codewörter entsprechen der Dualzahl, wobei die Codewörter für die Zahlenwerte 10 bis 15 nicht verwendet werden
  - 0 = 0000
  - 1 = 0001
  - …
  - 9 = 1001

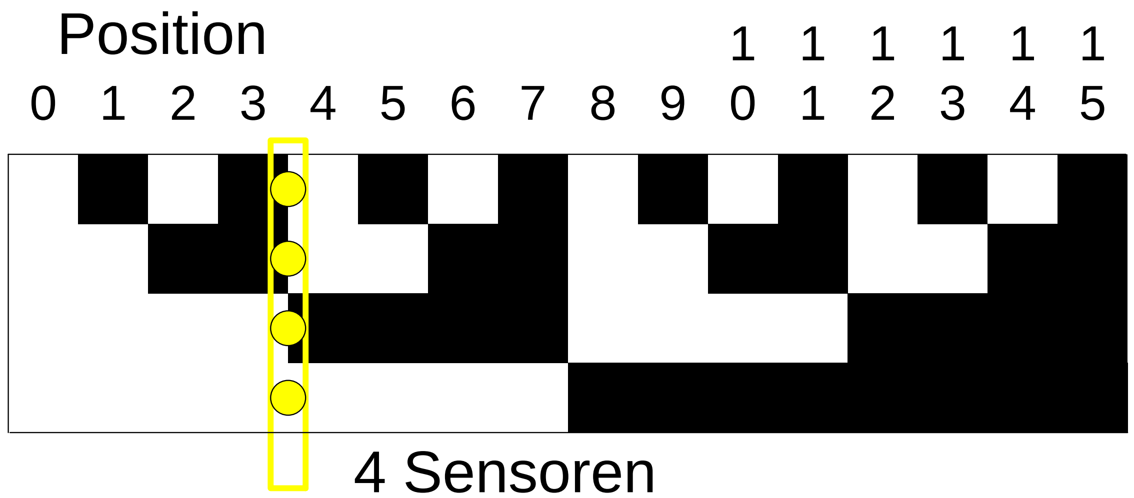
- BCD wird angewendet, wenn eine Anwendung mit Dezimalwerten arbeiten soll
  - Beispiel: Multimeter mit Dezimalanzeige
- Die **Rechnung** mit BCD-Werten ist aufwändiger als für Dualzahlen
- Die **Dezimalanzeige** von BCD-codierten Zahlen ist hingegen einfach

Gray-Code

- Der Gray-Code ist eine spezielle Darstellung zur **Erfassung** von Codewörtern

Beispiel für Problem

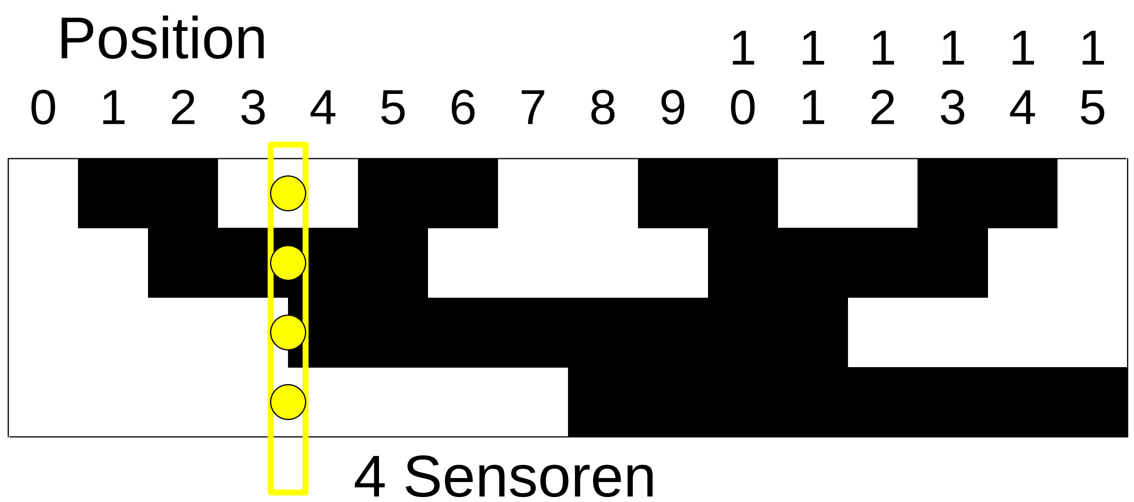
- Eine Maschine fährt auf einer horizontalen Bahn
- Durch 4 Sensoren soll die Position in 16 Schritten erfasst werden



- Im Dualcode (Bild) können **Ablesefehler** an den Übergängen auftreten
- Im Bild soll„0011" oder „0100"abgelesen werden
  - Aber es wird eventuell „0111" oder „0000" oder ???? erfasst
  - Für „0111" wird im Übergang Bit 2 schon, Bit 0 und 1 noch als ‚1' erfasst

Gray-Code (II)

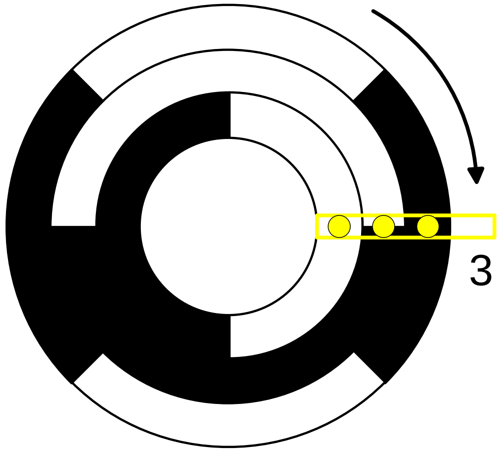
- Im **Gray-Code** unterscheiden sich benachbarte Codewörter immer nur an einer Stelle
  - Die Anzahl unterschiedlicher Stellen wird als **Hamming-Distanz** bezeichnet
  - Im Gray-Code haben benachbarte Codewörter somit immer die Hamming-Distanz eins



  - Die Stellen des Gray-Codes haben keine Wertigkeit
  - Für Rechenoperationen kann **nach der Erfassung** eine Umwandlung in den Dualcode erfolgen

Gray-Code (III)

- Der Gray-Code ist zyklisch, d.h. auch erstes und letztes Codewort haben die Hamming-Distanz Eins
  - Auch für Rotationsmessung geeignet
  - Bild zeigt Beispiel mit 3 bit



Bildung des Gray-Code

- Ein Algorithmus zur Wandlung Dualzahl *D(n-1:0)* nach Gray-Code *G(n-1:0)* lautet:

```
G[n-1]=D[n-1]
for i=n-2 to i=0
  { G[i]=D[i+1] xor D[i] }
```

- Ein Algorithmus zur Wandlung Gray-Code *G(n-1:0)* nach Dualzahl *D(n-1:0)* lautet:

```
D[n-1]=G[n-1]
for i=n-2 to i=0
  { D[i]=D[i+1] xor G[i] }
```