Kapitel 1

Das digitale Signal

1.1 Das digitale Signal

Ein **digitales Signal** ist sowohl wert- als auch zeitdiskret. Es kann endlich viele Zustände annehmen und sind von der Zeit abhängig. Ein binäre Signal kann nur 2 Zustände annehmen, während ein digitales Signal kann N Zustände annehmen. Muss also Informationenals mit mehr als zwei Werte fliessen, so muss dies durch sequentielles Aneinanderreihen von zwei-wertigen Einzelsignalen erfolgen.

1.2 Binäre Signale

Binäre Signale werden durch Spannungen abgebildet. Die Aussagenlogik stellt diese binäre Signale durch trueöder false", die Schaltungstechnik durch "highöder flowünd die Digitaltechnik durch "1öder "0".

1.3 Vorteile der Digitaltechnik

Folgende sind **Vorteile der Digitaltechnik**:

(1) Digitale Signale lassen sich einfacher fehlerfrei übertragen und verarbeiten.

(2) Keine Fortsetzungsfehler bei der Verarbeitung und Übertragung wie bei analogen Techniken.

(3) Digitale Information kann man einfacher fehlerfrei abspeichern.

1.4 Nachteile der Digitaltechnik

Folgende sind **Nachteile der Digitaltechnik**:

(1) Informationsverlust bei der Umwandlung von analogen Signalen in digitale Signale.

(2) Rundungsfehler bei der Verarbeitung mit ungenügender Wortlänge.

(3) Langsamer als analoge Verarbeitung. Vgl. HF-Technik, wo analog gearbeitet wird.

(4) Höherer Schaltungsaufwand als bei analoger Schaltungstechnik. Dank Miniaturisierung tritt dieser Nachteil immer mehr in den Hintergrund.

Kapitel 2

Digitale Darstellung von Signalen

Damit Informationen von Systemen verarbeitet weden können, müssen diese in geeigneter Form kodiert sein. Im Folgenden werden also verschiedene Methoden aufgezeigt, wie Informationen für digitale Verarbeitungssysteme abgebildet werden können.

2.1 Zahlencodes

2.1.1 Vorzeichenlose Ganzzahlen

Für die **Kodierung der natürlichen Zahlen inklusive 0** wird der vorzeichenlose Binärcode eingesetzt. Dieser ist ein Stellenwertsystem, was bedeutet, dass jede Ziffer gemäss ihrer Position in der Nummer eine andere **Wertigkeit** hat. Für den vorzeichenlosen Binärcode lässt sich dieser Sachverhalt gemäss folgende Gleichung beschreiben, wobei i die Position der Ziffer von rechts, z der Zifferwert und N die Anzahl Stellen sind.

$$\boxed{Z = \sum_{i=0}^{N-1} \left(z_i \cdot 2^i\right)} \tag{2.1}$$

2.1.2 Von der Binär- ins Dezimalsystem

Der dezimalen Wert der vorzeichenlosen Binärzahl `1011` kann berechnet werden, indem man die Produkte der Ziffern mit ihren zugehörigen Wertigkeiten aufsummiert.

$$Z = [1011]_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = [11]_{10}$$

2.1.3 Von der Dezimal- ins Binärsystem

Die Umwandlung von Dezimalzahlen in einer vorzeichenlose Binärzahlen geschieht durch eine iterative Division durch die Zahlenbasis des Zielformats. Dieser Vorgang wird so lange wiederholt, bis das Divisionsergebnis 0 ist.

$$[11]_{10} : 2 = \underbrace{5 \bmod 1}_{1,2^0}, \quad 5 : 2 = \underbrace{2 \bmod 1}_{1 \Rightarrow 2^1}, \quad 2 : 2 = \underbrace{1 \bmod 0}_{0 \Rightarrow 2^2}, \quad 1 : 2 = \underbrace{0 \bmod 1}_{1 \Rightarrow 2^2}$$

2.1.4 Wertebereich

Rechenwerke arbeiten mit fixen Wortlängen, also mit einer festen Anzahl von Ziffern. Die Wortlänge einer binären Zahl entspricht der Anzahl Ziffern, also **Bits**, aus der diese Zahl zusammengesetzt ist. Der Wertebereich von digitalen Rechenwerken ist begrenzt.

Der Werteberiech einer vorzeichenlose Binärzahl kann folgendermassen berechnet werden, wobei N die Anzahl Stellen oder Bits sind. Binärzahlen sind `nibble`, `byte`, `word` und `double word`.

$$\boxed{\text{Wertebereich} = 0 \ldots (2^N - 1)} \tag{2.2}$$

- (i) Ein `nibble` besitzt 4 Bits und hat einen Wertebereich von 0 bis 15
- (ii) Ein `byte` besitzt 8 Bits und hat einen Wertebereich von 0 bis 255
- (iii) Ein `word` besitzt 16 Bits und hat einen Wertebereich von 0 bis 65'636
- (iv) Ein `double word` besitzt 32 Bits und hat einen Wertebereich von 0 bis 4'294'967'296

2.1.5 Die Problematik des Überlaufs

In den digitalen Rechenwerken ist die Anzahl Stellen von binären zahlen begrenzt und damit auch der darstellbare Zahlenraum. Ist die letzte darstellbare zahl erreicht und wird zu dieser 1 hinzuaddiert, findet wegen des Fehlens einer weiteren nötigen Ziffer ein **Überlauf** statt. Das Resultat also ist eine 0. Den selben Mechanismus ist beim Subtrahieren von 0-1 zu beobachten. Hier findet ein Unterlauf statt und anstelle des erwarteten Resultats 1"wird die grösste darstellbareZahl ausgegeben. Bei Zahlensystemen mit einer begrenzten Anzahl Ziffern, findet man keinen linearen Zahlenstrahl, sondern einen **zyklischen Zahlenkreis**.

2.1.6 Der Hexadezimalcode

Beim HExadezimalcode beträgt die Zahlenbasis 16. Mit einer einzelnen Ziffer lässt sich damit den Wertebereich 0 bis 15 abdecken. Für die Werte 0 bis 9 wie beim Dezimalsystem und für die Werte 10 bis 15 die Buchstaben A bis F. Dieser Code wird häufig zur Darstellung von vorzeichenlosen binären Zahlen verwendet, wobei sich hiermit jeweils 4 Bits zu einer Ziffer zusammenfassen lassen und damit eine kürzere Schreibweise möglich ist.

Dezimal	Hexadezimal	Binär
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

(2.3)

Die **Umwandlung von Binär- nach Hexadezimal** gestaltet sich sehr einfach, indem man 4er Gruppen von Bits her nimmt und dann gruppenweise umwandelt. Analog funktioniert die **Umwandlung von Hexadezimal nach Binär**

$$[\texttt{10100110}]_2 = [\texttt{1010'0110}]_2 = [\texttt{A'6}]_{16} = \texttt{0xA6} \tag{2.4}$$

$$[\texttt{0x31AB}]_{16} = [\texttt{31AB}]_{16} = [\texttt{0011'0001'1010'1011}]_2 \tag{2.5}$$

2.1.7 Die vorzeichenlose binäre Addition

Die Addition von Binärzahlen geschieht Ziffern-weise von rechts nach links unter Berücksichtigung des Übertrags der Summe der jeweils vorhergehenden Ziffer. So gilt für die Einzelsumme einer jeden Ziffer `i`, wobei a_i der Summand 1, b_i der Summand 2, c_i der Übertrag der vorhergehenden Ziffer und c_{i+1} der Übertrag sind.

$$\boxed{\begin{aligned} sum_i &= \begin{cases} 1 & \text{wenn} \quad c_i + a_i + b_1 \quad \text{ungerade} \\ 0 & \text{sonst} \end{cases} \\ c_{i+1} &= \begin{cases} 1 & \text{wenn} \quad (c_i + a_i + b_1) > 1 \\ 0 & \text{sonst} \end{cases} \end{aligned}} \tag{2.6}$$

Das Summenbit ist also 1, wenn die Summe der Summandenbits und des Übertragsbits ungerade ist. Ansonsten ist das Summenbit 0. Das Übertragsbit für die nächste Ziffer ist 1, wenn die Summe der Summandenbits und des Übertragsbits grösser als 1 ist.

$$\underbrace{[\texttt{1100}]_2}_{\text{Summand 1}} + \underbrace{[\texttt{1010}]_2}_{\text{Summand 2}} = \underbrace{[\texttt{10000}]_2}_{\text{Übertrag}} = \underbrace{[\texttt{10110}]_2}_{\text{Summe}} \tag{2.7}$$

2.1.8 Die vorzeichenlose binäre Subtraktion

Die Subtraktion von Binärzahlen geschieht Ziffern-weise von rechts nach links unter Berücksichtigung des Übertrags der Differenz der jeweils vorhergehenden Ziffer. So gilt für die EInzeldifferenz einer jeden Ziffer `i`, wobei a_i der Minuend, b_i der Subtrahend, c_i der Übertrag der vorhergehenden Ziffer und c_{i+1} der Übertrag

$$\boxed{\begin{aligned} diff_i &= \begin{cases} 1 & \text{wenn} \quad a_i - (a_1 + b_1) \quad \text{ungerade} \\ 0 & \text{sonst} \end{cases} \\ c_{i+1} &= \begin{cases} 1 & \text{wenn} \quad (a_i - (b_1 + c_1)) < 0 \\ 0 & \text{sonst} \end{cases} \end{aligned}} \tag{2.8}$$

Das jeweilige Bit der Differenz ist also 1, wenn Minuend-(Subtrahend+Übetrag) ungerade ist, sonst istes 0. Das Übertragsbit (borrow) für die nächste Stelle wird gesetzt, wenn das Resultat der aktuellen Stelle negativ ist.

$$\underbrace{[1100]_2}_{\text{Minuend}} - \underbrace{[0110]_2}_{\text{Subtrahend}} = \underbrace{[1100]_2}_{\text{Borrow}} = \underbrace{[0110]_2}_{\text{Differenz}} \tag{2.9}$$

Alternativ kann die Subtraktion mittels Addition durchgeführt werden. DAbei wird zum Minuenden das **Zweierkomplement des Subtrahenden** addiert.

$$\underbrace{[1100]_2}_{\text{Minuend}} - \underbrace{[0110]_2}_{\text{Subtrahend}} \implies \underbrace{[1010]_2}_{\text{Zweierkomplement}} \implies [1100]_2 + [1010]_2 = \underbrace{[10000]_2}_{\text{Übertrag}} = \underbrace{[0110]_2}_{\text{Differenz}} \tag{2.10}$$

2.1.9 Vorzeichenbehaftete Zahlen

Es gibt mehrere Möglichkeiten, vorzeichenbehaftete Zahlen darzustellen. Im Folgenden wird lediglich die Zweierkomplement-Darstellung behandelt, da die anderen Systeme nur schwach verbreitet sind. Im Speziellen sind hier die Betrag-Vorzeichen-Darstellung und Bias-Darstellung zu nennen.

2.2 Fehlererkennende und Fehlerkorrigierende Codes

2.3 Zeichencodes