5.3 Wandlerkennwerte

5.3.1 Statische Kennwerte

Auflösung (Resolution)
Die Auflösung betrifft die geringste Wertänderung auf der digitalen Seite, mit anderen Worten: den Unterschied zwischen zwei unmittelbar benachbarten Digitalwerten. Ein Wandler, der für n Bits ausgelegt ist, kann mit 2^n Binärwerten (Codes) arbeiten, und zwar typischerweise im Bereich von 0 bis 2^n-1. Dabei gibt es 2^n-1 Wertübergänge. Beim A-D-Wandler bezieht sich die Auflösung auf die eingangsseitige Spannungsänderung, der erforderlich ist, um den Übergang von einem Binärwert zum nächsten zu veranlassen, beim D-A-Wandler ist es die Ausgangsspannungsänderung, die sich infolge eines solchen Übergangs ergibt (Tabelle 5.3.1). Die Auflösung ergibt sich ausschließlich aus der Auslegung des Digitalteils. Es ist keine Aussage über Genauigkeit und Linearität.

Zwei wichtige Fachbegriffe: LSB und FSR:

- LSB = Least Significant Bit = die niedrigstwertge Bitposition,
- FS = Full Scale. Der jeweilige Höchstwert. Der analoge Höchstwert (Analog FS) entspricht typischerweise der Referenzspannung.
- FSR = Full Scale Range = der gesamte Wertebereich.

Es sind zwei Wertebereiche zu unterscheiden:

- der analoge (Analog FSR): der Bereich von der geringsten bis zur höchsten analogen Spannung (kontinuierlich),
- der digitale (Digital FSR): der Bereich aller Digitalwerte (diskret). Mit n Bits kann man 2n Werte darstellen. Die einfachste Codierung: als natürliche (vorzeichenlose) Binärzahlen. Wertebereich einer Binärzahl aus n Bits: von 0 bis 2^n-1 (000...00 bis 111...11 oder 00...0H bis FF...FH).

Die volkstümliche Auflösungsangabe
Ein n-Bit-Wandler hat eine Auflösung von n Bits oder 2^{-n}.

Die exakte Auflösungsangabe
Die Auflösung wird als Wert der niedrigstwertigen Bitposition (LSB) angegeben. Sie wird auf den gesamten Wertebereich (FSR) bezogen. 1 LSB entspricht der Spannungsänderung, die eine Änderung in der niedrigstwertigen Bitposition hervorruft (A-D) oder die sich durch eine solche Änderung ergibt (D-A).

Was ist ein LSB (1)?
Ein n-Bit-Wandler unterteilt den gesamten Wertebereich in 2^n-1 gleichgroße Intervalle. Deshalb gilt:

$$1\ \mathrm{LSB} = \frac{\mathrm{FS}}{2^n - 1}$$

Was ist ein LSB (2)? - die übliche Praxis
Jeder Binärwert entspricht im Grunde einem Verhältnis zwischen der jeweiligen (analogen) Spannung und der festen Bezugsspannung (Referenzspannung). Es liegt nahe, folgende Entsprechungen anzusetzen:

- analoge Spannung = 0 V entspricht Binärwert Null,
- analoge Spannung = Referenzspannung entspricht dem maximalen Binärwert.

Um bequem (soll heißen: mit Zweierpotenzen) rechnen zu können, setzt man:

$$1\ \mathrm{LSB} = \frac{\mathrm{FS}}{2^n}$$

und definiert:

Binärer Höchstwert (Digital FS)= Höchstwert den analogen Spanung (Analog FS = Referenzspannung) - 1 LSB.

Die diskreten Spannungen im Wandler (vgl. die Widerstandsketten) sind typischerweise nach Zweierpotenzen abgestuft ($^1/_2$, $^1/_4$, $^1/_8$, $^1/_{16}$ usw.). Gängige Referenzspannungen: 1,024 V; 1,25 V; 2,048 V; 2,5 V; 4,096 V; 5,0 V; 8,192 V; 10,0 V.

Beispiel: 4,096 V; 12 Bits. 1 LSB = 4,096 V : 4096 = 1 mV. Der binäre Höchstwert entspricht 4,095 V.

Wie weit kann man den Wertebereich ausnutzen? - Eine Faustregel:
Ein n-Bit-Wandler hat einen ausnutzbaren Wertebereich (Dynamikbereich) von rund 6n dB.

Beispiel: n = 8; Dynamikbereich 1 : $10^{\frac{48}{20}} \approx 1 : 250$

Wir merken uns:
Geht es um Genauigkeit, kann man die Wandler nicht bis zum Anschlag ausnutzen (Stichwort: ENOB = Effective Number of Bits).

Bits	Binärwerte	Auflösung in % vom Endwert (% Full Scale)	Wert der niedrigstwertigen Bitposition (LSB)		
			Bereich = 20 V	Bereich = 5 V	Bereich = 2 V
8	256	0,39%	78.1mV	19.5mV	7.81mV
10	1024	0,098%	19.5mV	4.88mV	1.95mV
12	4096	0,0244%	4.88mV	1.22mV	488µV
14	16 384	0,0061% = 61 ppm	1.22mV	305µV	122µV
16	65 536	0,0015% = 15 ppm	305µV	76.3µV	30.5µV
18	262 144	0,000381% = 3,8 ppm	76.3µV	19.1µV	7.63µV
20	1 048 576	0,000095% = 0,95 ppm	19.1µV	4.78µV	1.91µV

Tabelle 5.3.1 Typische Auflösungsangaben

Genauigkeit (Accuracy)

Die Genauigkeit ist eine Pauschalangabe, die sich auf Abweichungen gegenüber dem Verhalten eines idealen Wandlers bezieht (Abbildungen 5.3.1 bis 5.3.3).

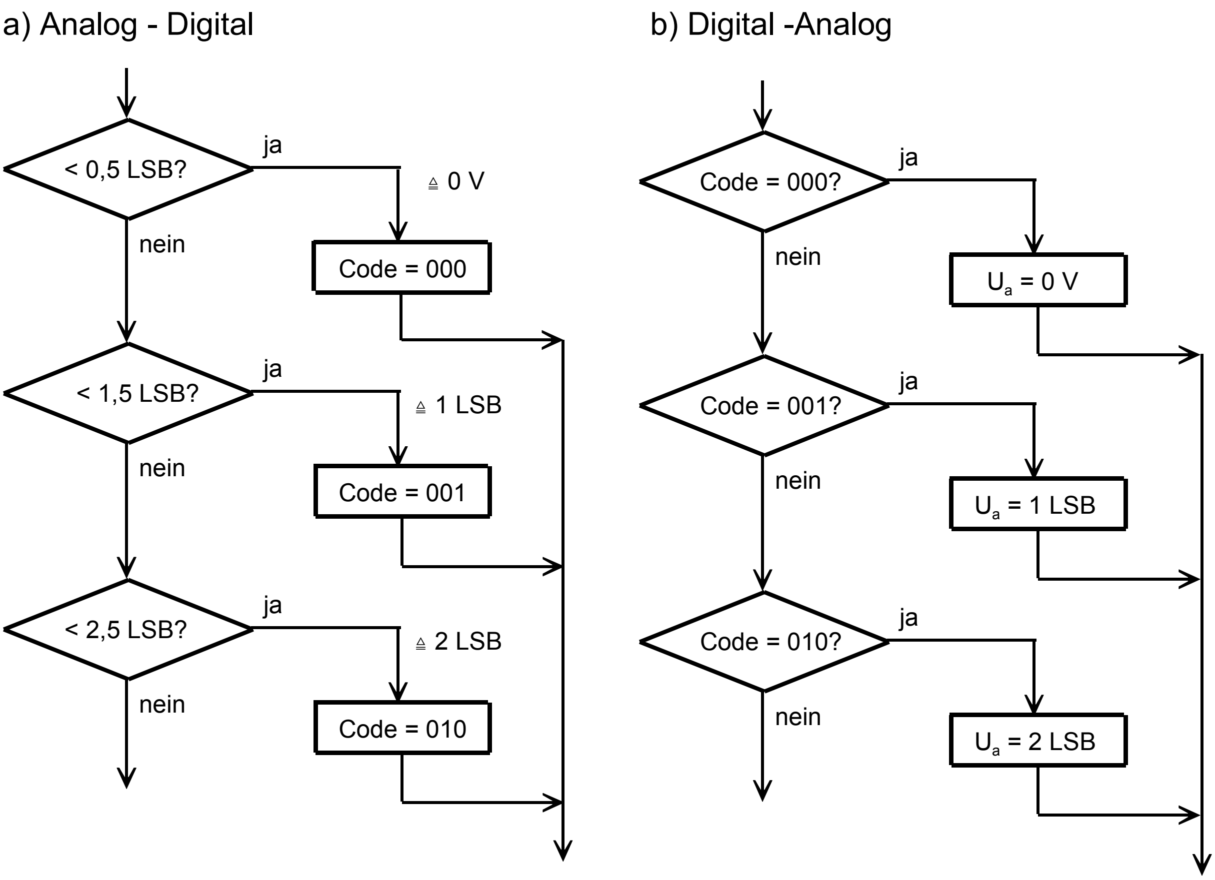


Abbildung 5.3.1 Zur grundsätzlichen Arbeitsweise der Wandler

Erklärung:

a) Analog-Digital-Wandlung. Es gibt einen kontinuierlichen analogen Spannungsbereich, aber nur 2^n diskrete Binärwerte. Beim Wandeln muß also gerundet werden. Hier runden wir nach gesundem Volksempfinden. Die Digitalwerte entsprechen 0 V, 1 LSB, 2 LSB usw. (diskrete Spannungswerte). Ist die analoge Spannung kleiner als die Hälfte des jeweils nächsten LSB-Wertes, so wird abgerundet. Andernfalls wird aufgerundet. Die Übertragungsfunktion wird somit zur Treppenkurve (Abbildung 5.3.2). Beim idealen Wandler sind alle Treppenstufen gleich breit (nämlich 1 LSB), und die Gerade, die die ideale Übertragungskennlinie symbolisiert, geht mitten hindurch - die diskreten Spannungswerte (1 LSB, 2 LSB usw.) liegen in der Mitte der jeweiligen Treppenstufe (Midstep Values). Jeder Treppenstufe entspricht einer der diskreten Spannungswerte ± ½ LSB (Rundung).

b) Digital-Analog-Wandlung. Jeder Binärwert veranlaßt, daß der Wandler eine entspechende Analogspannung ausgibt. Am Ausgang erscheinen also nur diskrete Spannungswerte (0 V, 1 LSB, 2 LSB usw.). Die Übergangsfunktion wird somit zu einer Punktfolge die auf einer Geraden (der idealen Kennlinie) liegt (Abbildung 5.3.3).

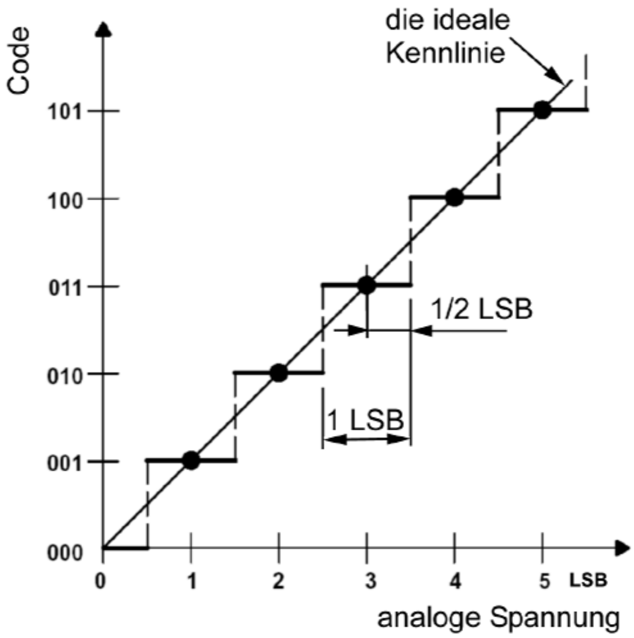


Abbildung 5.3.2 Die Übertragungskennline eines idealen A-D-Wandlers (nach: Texas Instruments)

Wir merken uns: Alle Analog-Digital-Wandler sind mit einem unvermeidlichen Digitalisierungsfehler behaftet, der der halben Auflösung entspricht (±½ LSB). In der Praxis ist der Digitalisierungsfehler typischerweise höher (± 1 LSB und mehr).

Faustregel: Soll das gesamte System eine bestimmte Genauigkeit haben, sind Komponenten mit jeweils 5...10mal größerer Genauigkeit einzusetzen.

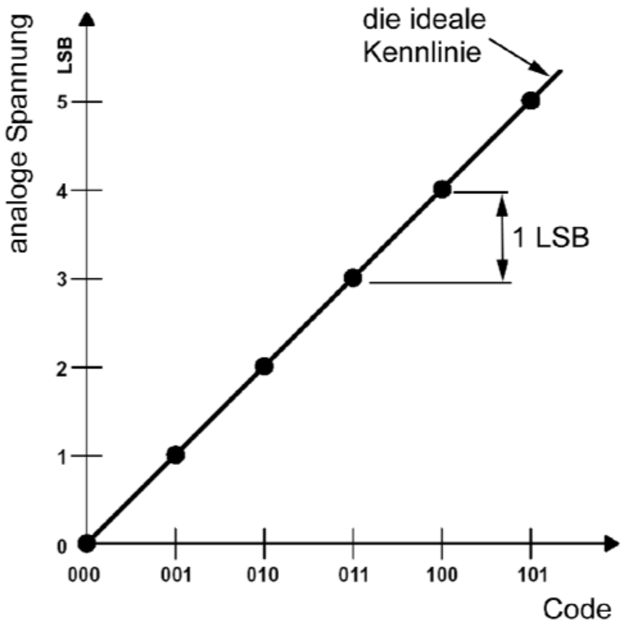


Abbildung 5.3.3 Die Übertragungskennline eines idealen D-A-Wandlers (nach: Texas Instruments)

Die wesentlichen statischen Kennwerte, die die Genauigkeit näher kennzeichnen, sind: Offsetfehler, Verstärkungsfehler, differentielle Nichtlinearität und integrale Nichtlinearität. Sie werden in LSBs, als Spannungswert oder in Prozent vom Bereichsendwert (%FS) angegeben. Der Gesamtfehler (Total Error) ergibt sich aus der Überlagerung der Einzelfehler.

Offsetfehler
Beim A-D-Wandler gibt der Offsetfehler an, welche Eingangsspannung noch einen Digitalwert = 0 hervorruft (Abbildung 5.3.4), beim D-A-Wandler, welche Ausgangsspannung bei einem Digitalwert = 0 abgegeben wird (Abbildung 5.3.5).

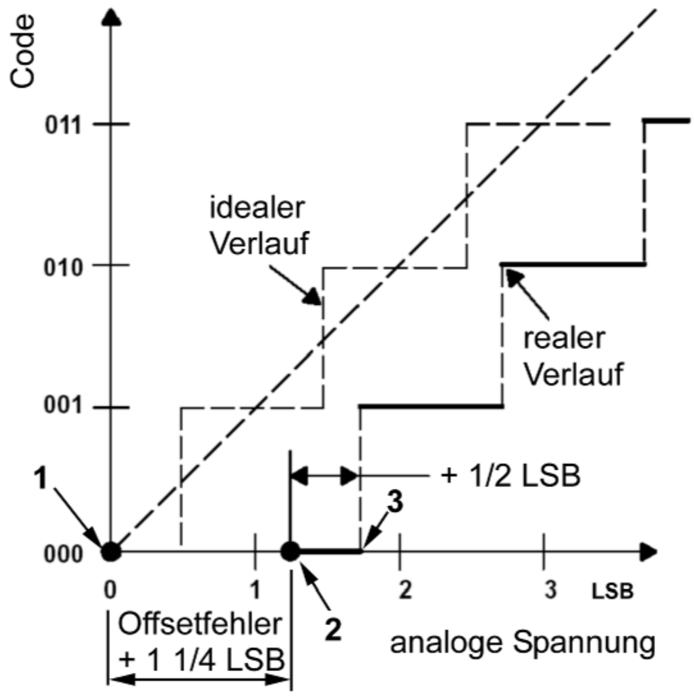


Abbildung 5.3.4 A-D-Wandler: Übertragungskennlinie mit Offsetfehler (nach: Texas Instruments)

Erklärung:
Der Offsetfehler kennzeichnet die Verschiebung der Übertragungskennlinie auf der x-Achse. 1 - Kennlinienbeginn ohne Offsetfehler (Punkt 0, 0); 2 - Kennlinienbeginn mit Offsetfehler; 3 - hier schaltet der Wandler auf den nächsten binären Code um (Aufrundung). Der Punkt 3 kennzeichnet somit die höchste Eingangsspannung, bei der noch ein Binärcode Null geliefert wird. Um den Offsetfehler zu bestimmen, wird der schlimmste Rundungsfehler abgezogen, also ½ LSB zurückgerechnet (Midstep Value). Im Beispiel: Punkt 3 entspricht ca. 1¾ LSB. Abzüglich ½ LSB ergibt sich ein Offsetfehler von 1¼ LSB.

Hinweis:
Der Offsetfehler betrifft alle Codes gleichermaßen. Er kann oft kompensiert werden (Trimmung).

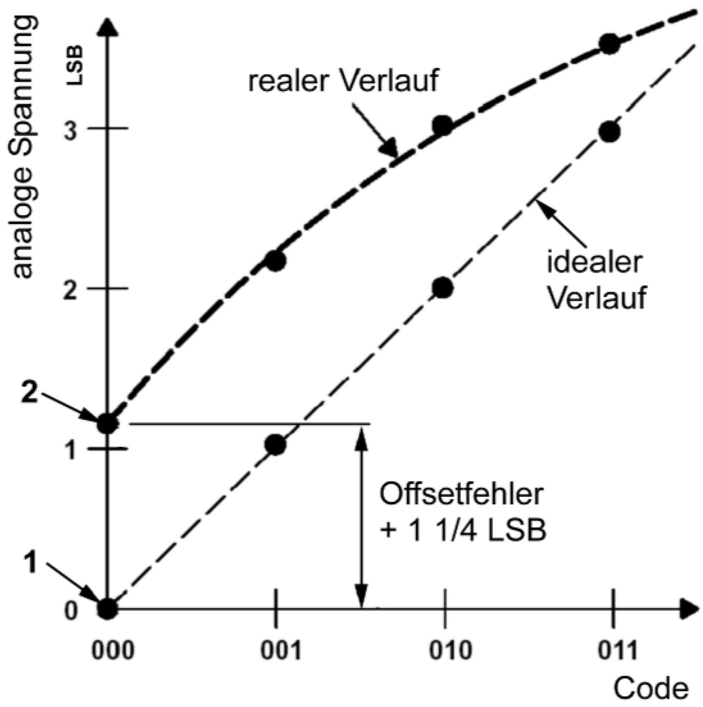


Abbildung 5.3.5 D-A-Wandler: Übertragungskennlinie mit Offsetfehler (nach: Texas Instruments)

Erklärung:
Der Offsetfehler kennzeichnet die Verschiebung der Übertragungskennlinie auf der y-Achse. 1 - Kennlinienbeginn ohne Offsetfehler (Punkt 0, 0); 2 - Kennlinienbeginn mit Offsetfehler (hier: um 1¼ LSB versetzt). Liegt der Binärwert Null an, so gibt der Wandler im Beispiel eine Spannung von ca. 1¼ LSB ab.

Verstärkungsfehler (Gain Error)
Der Offsetfehler betrifft den Anfang der Übertragungskennline, der Verstärkungsfehler deren Ende, nämlich die Abweichung vom idealen Verlauf am Bereichsendwert. Beim A-D-Wandler gibt der Verstärkungsfehler den Unterschied zwischen dem Nennwert und dem tatsächlichen Wert der Eingangsspannung an, die den höchsten Digitalwert hervorruft (Abbildung 5.3.6), beim D-A-Wandler die Abweichung der Ausgangsspannung vom Nennwert, wenn der höchste Digitalwert gewandelt wird (Abbildung 5.3.7). Dabei wird ein Offsetfehler von Null angenommen.

Hinweis:
Der Verstärkungsfehler kennzeichnet eine vom Idealfall abweichende Steigung der Übertragungskennlinie. Davon sind alle Codes in gleicher Weise anteilig (d. h. mit soundsoviel % Abweichung) betroffen. Diese Abweichung kann oft kompensiert werden (Trimmung).

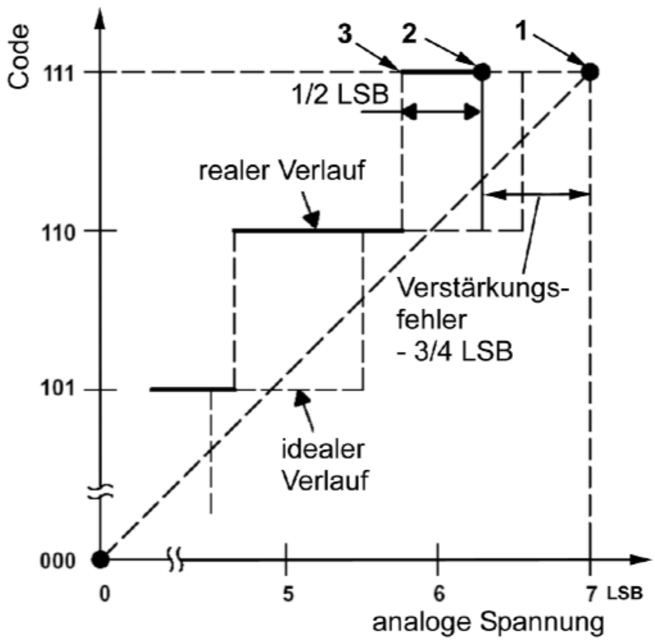


Abbildung 5.3.6 A-D-Wandler: Übertragungskennlinie mit Verstärkungsfehler (nach: Texas Instruments)

Erklärung:

1 - Kennlinienende ohne Verstärkungsfehler; 2 - Kennlinienende mit Verstärkungsfehler; 3 - hier schaltet der Wandler auf den höchsten binären Code um (Aufrundung). Der Punkt 3 kennzeichnet somit die geringste Eingangsspannung, bei der schon de höchste Binärcode (111) geliefert wird. Um den Vestärkungsfehler zu bestimmen, wird der schlimmste Rundungsfehler aaddiert, also ½ LSB hinzugerechnet (Midstep Value). Im Beispiel: Punkt 3 entspricht ca. 5¾ LSB. Zuzüglich ½ LSB ergibt sich ein Midstep Value von 6¼ LSB. Der Idealwert: 7 LSB. Der Verstärkungsfehler beträgt somit - ¾ LSB.

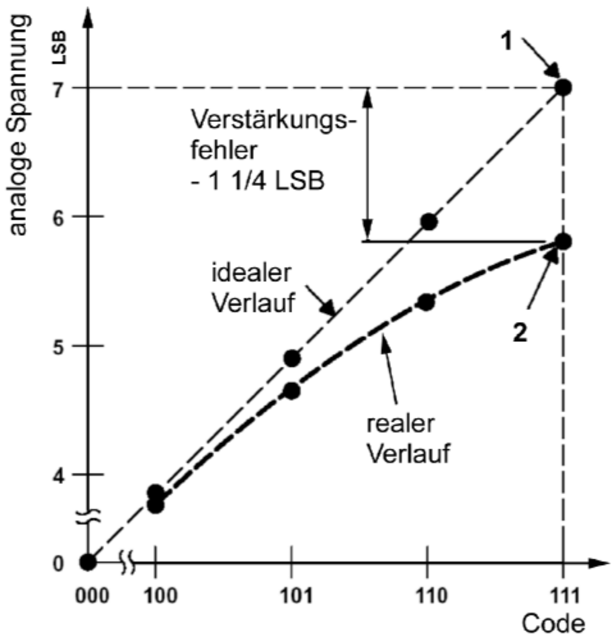


Abbildung 5.3.7 A-D-Wandler: Übertragungskennlinie mit Verstärkungsfehler (nach: Texas Instruments)

Erklärung zu Abbildung 5.3.7:
1 - Kennlinienende ohne Verstärkungsfehler; 2 - Kennlinienende mit Verstärkungsfehler (hier: um 1¼ LSB nach unten versetzt). Liegt der höchste Binärwert (111) an, so gibt der Wandler im Beispiel eine Spannung von ca. 5¾ LSB ab.

Differentielle Nichtlinearität (DNL)
Die differentielle Nichtlinearität beschreibt die Abweichungen vom idealen Verlauf der Übertragungsfunktion. Beim A-D-Wandler geht es um die Breite der Treppenstufen (Abbildung 5.3.8), beim D-A-Wandler um die Abstände zwischen den aufeinanderfolgenden diskreten Spannungen (Abbildung 5.3.9). Die Datenblattangabe betrifft die maximale dieser Abweichungen. Beim A-D-Wandler gibt die differentielle Nichtlinearität den größten Spannungsunterschied an, der erforderlich ist, um den Binärwert um Eins zu ändern, beim D-A-Wandler den größten Spannungsunterschied, der sich bei Änderung des Binärwerts um Eins ergibt.

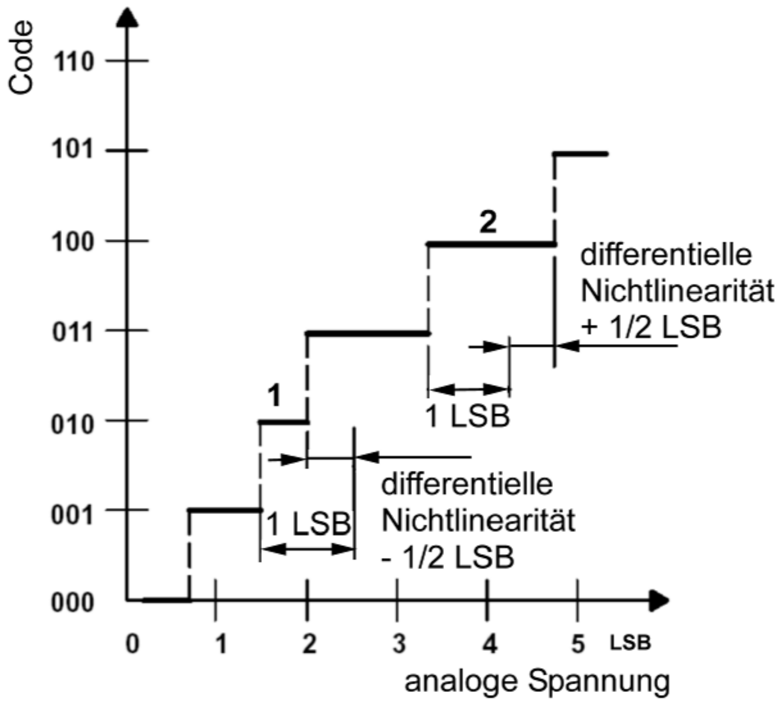


Abbildung 5.3.8 A-D-Wandler: Übertragungskennlinie mit differentieller Nichtlinearität (nach: Texas Instruments)

Erklärung:
Im Idealfall hat jede Treppenstufe eine Breite von 1 LSB. 1 - Treppenstufe zu schmal; Wandler schaltet zu früh von 010 auf 011 (nämlich schon bei 2 LSB anstatt bei 2½). Fehler = - ½ LSB. 2 - Treppenstufen zu breit. Fehler = ½ LSB.

Fehlende Binärwerte (Missing Codes)
So etwas kann bei A-D-Wandlern vorkommen. Eine differentielle Nichtlinearität von -1 LSB bedeutet, daß Binärwerte übergangen werden (also beim Durchfahren des Eingangsspannungsbereichs gar nicht erscheinen).

Hinweis:
Manche Wandler sind für „No missing codes" über die gesamte Auflösung spezifiziert, manche nur für eine eingeschränkte Auflösung (z. B. ein 16-Bit-Wandler für „No missing codes to 15 bits"). Solche Angaben (auch: DNL = - 1 LSB o. ä.) deuten darauf hin, daß man eigentlich nur mit der halben Auflösung sinnvoll arbeiten kann.

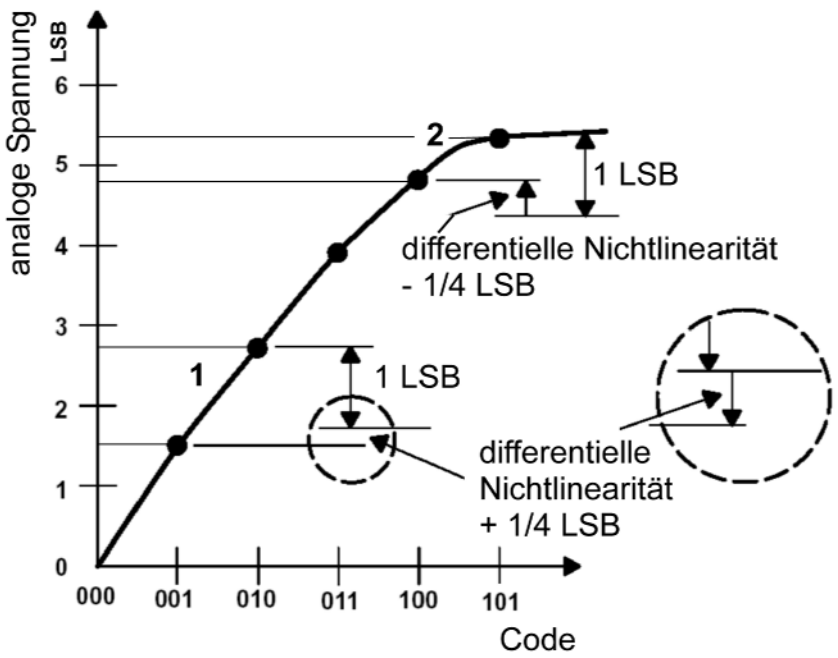


Abbildung 5.3.9 D-A-Wandler: Übertragungskennlinie mit differentieller Nichtlinearität (nach: Texas Instruments)

Erklärung:
Im Idealfall haben die aufeinanderfolgenden diskreten Spannungen einen Abstand von 1 LSB.
1 - zwischen den Binärwerten 001 und 010 ist der Spannungssprung zu groß. Fehler = ¼ LSB.
2 zwischen den Binärwerten 100 und 101 ist der Spannungssprung zu gering. Fehler = - ¼ LSB.

Integrale Nichtlinearität (INL)
Die integrale Nichtlinearität (auch als statische oder absolute Linearität bezeichnet) beschreibt die Abweichungen von einer geraden Linie zwischen den Endpunkten. Der Datenblattwert ist die maximale Abweichung von dieser Linie . Beim Analog-Digital-Wandler geht es um den Übergang von einer Treppenstufe zur nächsten (Abbildung 5.3.10), beim Digital-Analog-Wandler um die Abweichung jeder einzelnen diskreten Spannung (Abbildung 5.3.11).

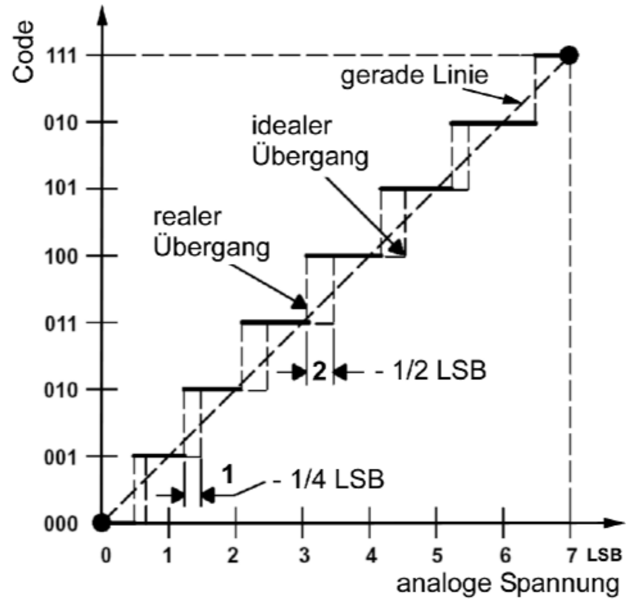


Abbildung 5.3.10 A-D-Wandler: Übertragungskennlinie mit integraler Nichtlinearität (nach: Texas Instruments)

Erklärung:

1 - beim Übergang zwischen 001 und 010 ist die untere Stufe zu schmal. Statt bei 1½ LSB wird schon bei ca. 1¼ LSB umgeschaltet. Fehler: - ¼ LSB. 2 - Übergang zwischen 011 und 100. Fehler: - ½ LSB.

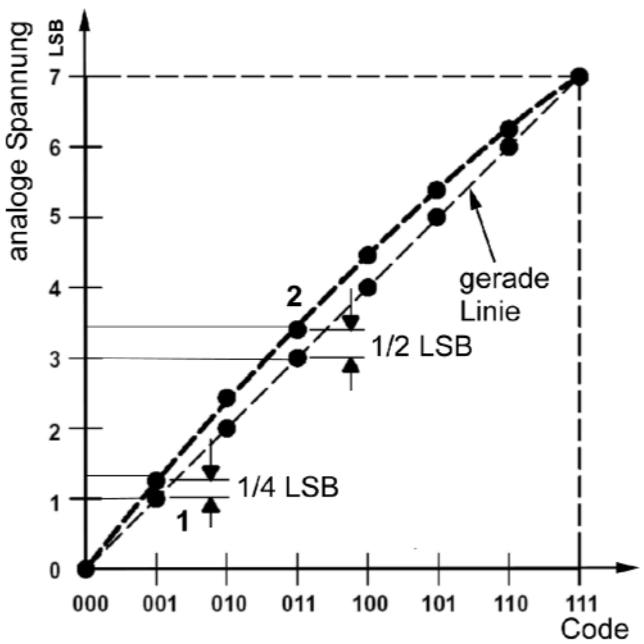


Abbildung 5.3.11 D-A-Wandler: Übertragungskennlinie mit integraler Nichtlinearität (nach: Texas Instruments)

Erklärung:

1 - Binärwert 001 ergibt eine Ausgangsspannung von 1¼ LSB (statt 1 LSB). Fehler: ¼ LSB. 2 - Binärwert 011 ergibt eine Ausgangsspannung von 3½ LSB (statt 3 LSB). Fehler: ½ LSB.

Gesamt- oder Absolutfehler (Total/Absolute Error)

Der Gesamtfehler betrifft die maximale Abweichung zwischen der tatsächlichen analogen Spannung und dem jeweiligen diskreten Idealwert (Midstep Value, also 0 V, 1 LSB, 2 LSB usw.; Abbildungen 5.3.12, 5.3.13)). In dieser Fehlerangabe sind zusammengefaßt: Offsetfehler, Verstärkungsfehler, differentielle Nichtlinearität und integrale Nichtlinearität. Beim Analog-Digital-Wandler kommt der Quantisierungsfehler hinzu.

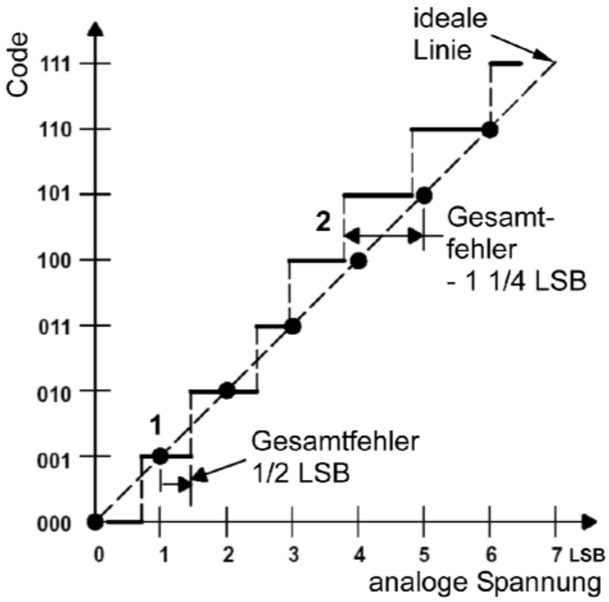


Abbildung 5.3.12 A-D-Wandler: Übertragungskennlinie mit Gesamtfehler (nach: Texas Instruments)

Erklärung:

1 - Gesamtfehler bei Binärwert 001: ½ LSB. 2 - Gesamtfehler bei Binärwert 101: - 1¼ LSB.

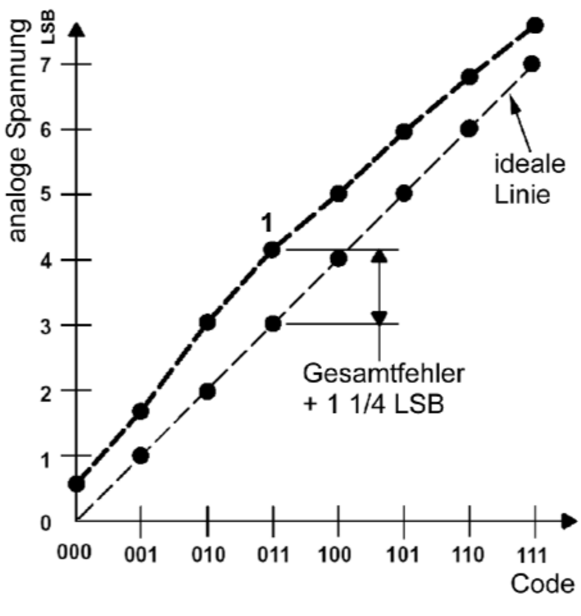


Abbildung 5.3.13 D-A-Wandler: Übertragungskennlinie mit Gesamtfehler (nach: Texas Instruments)

Digitalisierungs- oder Quantisierungsfehler (Quantizing Error)
Diese Fehlerangabe betrifft Analog-Digital-Wandler.Sie kennzeichnet die maximale Abweichung der Übertragungsfunktion von der idealen (geraden) Linie. Dabei bezieht man sich auf einen ansonsten idealen Wandler (kein Offsetfehler, Verstärkungsfehler usw.). Der kleinstmögliche Digitalisierungsfehler: ± ½ LSB (vgl. Abbildung 5.3.2).

Hinweis:
Der Digital-Analog-Wandler kennt keinen Quantisierungsfehler im eigentlichen Sinne, da an seinem Ausgang nur diskrete Spannungswerte erscheinen. Die Abweichungen vom jeweiligen Idealwert werden typischerweise durch andere Fehlerangaben gekennzeichnet (z. B. Nichtlinearität (DNL/INL) und Endwertfehler).

Endwertfehler ((Full) Scale Error)
Diese Fehlerangabe betrifft die Abweichung des realen Spannungswertes vom idealen bei Vorliegen des höchsten Binärwertes (111...11). Beim A-D-Wandler geht es um den Spannungswert, der das Umschalten auf den höchsten Binärwert veranlaßt, beim D-A-Wandler um den Spannungswert, der bei Anliegen des höchsten Binärwertes ausgegeben wird.

Endwertfehler können oftmals kompensiert werden (Trimmung), z. B. durch Einstellen der Referenzspannung oder (D-A-Wandler) der ausgangsseitigen Verstärkung.

Hysteresefehler (Hysteresis Error)
Diese Fehlerangabe betrifft Analog-Digital-Wandler. Sie kennzeichnet die Abhängigkeit des Binärwertes von der Richtung der eingangsseitigen Spannungsänderung (Anstieg oder Abfall). Eine gewisse Hysterese ist unvermeidlich (infolge der Comparatoren); sie sollte aber deutlich unter ½ LSB liegen.

Monotonizität (Monotonicity)
Eine monotone Übertragungsfunktion hat keine Richtungsänderung (Abfall bei ansonsten anstiegender, Anstieg bei ansonsten abfallender Kurve); mit anderen Worten, das Vorzeichen ihrer Ableitung ändert sich nicht. Ist diese Bedingung nicht erfüllt, kann es Ärger geben (vor allem in regelungstechnischen Anwendungen).

Anforderungen an den A-D-Wandler: Kein abnehmender Binärwert bei zunehmender Eingangsspannung.

Anforderung an D-A-Wandler: keine abnehmende Ausgangsspannung bei zunehmendem Binärwert. Gegenbeispiel: Abbildung 5.3.14.

Eine nichtmonotone Übertragungsfunktion ergibt sich dann, wenn die differentielle Nichtlinearität 1 LSB überschreitet.

Eine Nichtlinearität von höchstens ±½ LSB gewährleistet, daß die Übertragungsfunktion monoton ist. Gelegentlich wird die Monotonizität nur bis zu einer bestimmten Bitzahl garantiert (wenn es auf Genauigkeit ankommt, sind die verbleibenen (niederwertigen) Bitpositionen nicht ausnutzbar).

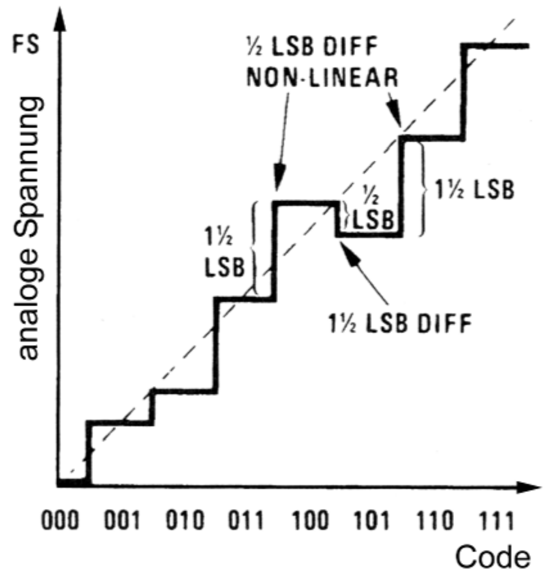


Abbildung 5.3.14 Beispiel einer nichtmonotonen Übertragungsfunktion (nach: National Semiconductor)

5.3.2 Dynamische Kennwerte im Überblick

Dynamische Kennwerte könnem im Zeitbereich oder im Frequenzbereich spezifiziert sein.

Elementare Zusammenhänge zwischen Zeit- und Frequenzbereich

Impulsantwort: $U_a(t) = U_0(1 - e^{-\frac{t}{\tau}})$; τ = RC (Zeitkonstante)

$$\text{Bandbreite} = \frac{0{,}35}{\text{Anstiegszeit}}$$

Anstiegszeit (Rise Time) $= 2{,}2 \cdot \tau$

Beruhigungszeit (Settling Time) bis zu einer Abweichung ε vom Endwert $= - \tau \cdot \ln \varepsilon$

Achtung:
Diese Zusammenhänge gelten für Systeme 1. Ordnung. Bei hohen Geschwindigkeiten und/oder hoher Auflösung (Richtwert: ab 10 Bits) sind die Effekte 2. Ordnung nicht mehr vernachlässigbar. Faustformeln gelten dann nur sehr näherungsweise.

Signalverläufe im Frequenzbereich

Jeder periodische (sich in festen Abständen ständig wiederholende) Signalverlauf kann in eine Folgen von Sinusschwingungen aufgelöst werden (Fourier-Zerlegung; Näheres in Abschnitt 5.5). Hierbei ergibt sich folgenden Sinusschwingungen:

- die Grundwelle oder Grundschwingung (Fundamental Wave F). Deren Frequenz entspricht dem Kehrwert der Periodendauer des ursprünglichen Signals.
- mehrere (im mathematischen Sinne unendlich viele) Harmonische (Harmonics H) bzw. Oberwellen oder Oberrchwingungen. Deren Frequenz entspricht jeweils einem Vielfachen der Grundwelle (2., 3. 4. usw. Harmonische).

Hinweis:
Die Grundwelle ist die 1. Harmonische. Die zweite Harmonische (= 1. Oberwelle/Oberschwingung) hat die doppelte Frequenz der Grundwelle usw.

Die Signaldarstellung als Frequenzspektrum
Da es bekannt ist daß es sich um Sinusschwingungen handelt, kann man darauf verzichten, deren Verlauf wiederzugeben. Ein übersichtlicheres Bild ergibt sich, wenn man auf der waagerechten Achse die Frequenz und auf der senkrechten die Amplitude darstellt (Frequenzspektrum bzw. Spektraldarstellung; Abbildung 5.3.15).

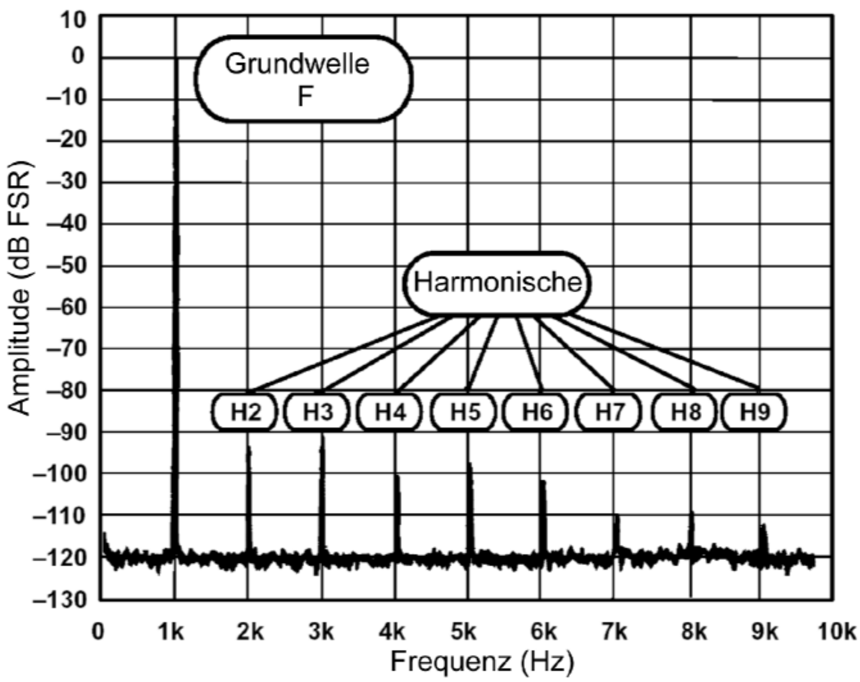


Abbildung 5.3.15 Beispiel einer Spektraldarstellung (nach: Texas Instruments)

Erklärung:
Im Beispiel hat die Grundwelle eine Frequenz von 1 kHz und eine Amplitude von 0dB, bezogen auf den Bereichsendwert (FSR), die 2. Harmonische hat eine Frequenz von 2 kHz und eine Amplitude von -90 dB, bezogen auf den Bereichsendwert usw.

Aufgabe:
Welche „richtigen“ Amplituden haben die Grundwelle und die 2. Harmonische, bezogen auf den Bereichsendwert (FSR)?

0 dB = 100 = 1. Also Amplitude Grundwelle = FSR. - 90 dB = $1/10^{90/20} \approx 3{,}16 \cdot 10^{-5}$. Also 2. Harmonische = 0,0000316 FSR. Es handelt sich um eine nahezu reine Sinusschwingung, die nur von geringen Störungen/Verzerrungen überlagert ist.

Die ins Digitale gewandelte reine Sinusschwingung
Im Digitalen kann jeder analoge (= kontinuierliche) Signalverlauf nur näherungsweise wiedergegeben werden (Abbildung 5.3.16):

- die Analog-Digital-Wandlung ist grundsätzlich mit einem Quantisierungsfehler behaftet (≥ ½ LSB),
- aus dem kontinuierlichen Signalverlauf kann zweck Wandlung nur eine bestimmte Anzahl an Stichproben entnomen werden (Signalabtastung; Näheres in Abschnitt 5.5).

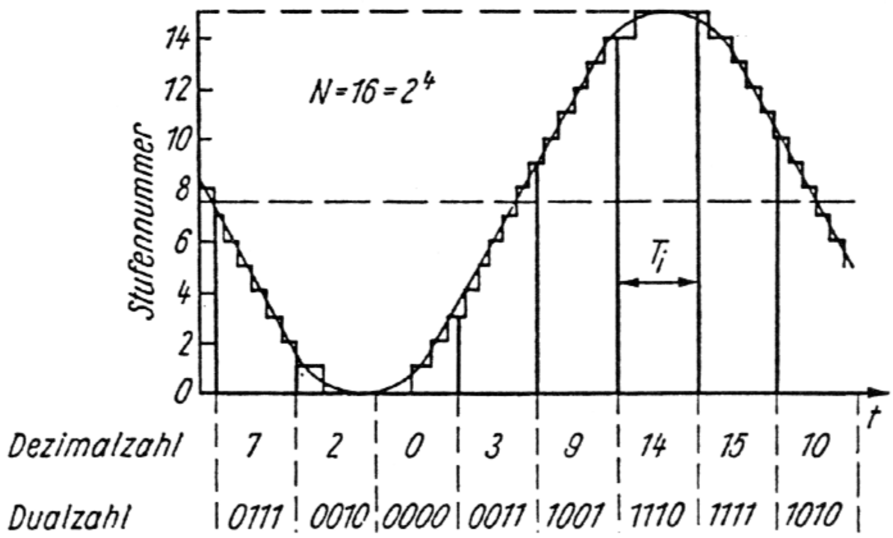


Abbildung 5.3.16 So wird eine Sinusschwingung im Digitalen wiedergegeben

Das Frequenzspektrum eines solchen digitalisierten Signalverlaufs läßt sich rechnerisch bestimmen (Fast Fourier Transformation FFT). Es ist durch Stör- bzw. Rauschanteile (Noise) gekennzeichnet, die den gesamten Frequenzbereich belegen (Abbildung 5.3.17).

Amplitudenspektrum und Leistungsspektrum
Die Diagramme sehen ähnlich aus, unterscheiden sich aber in der Interpretation der senkrechten Achse:

- Amplitudenspektrum: Spannungswerte. dB-Rechnung gemäß 20 · lg Spannungsverhältnis.
- Leistungsspektrum: Leistungswerte. dB-Rechnung gemäß 10 · lg Leistungsverhältnis.

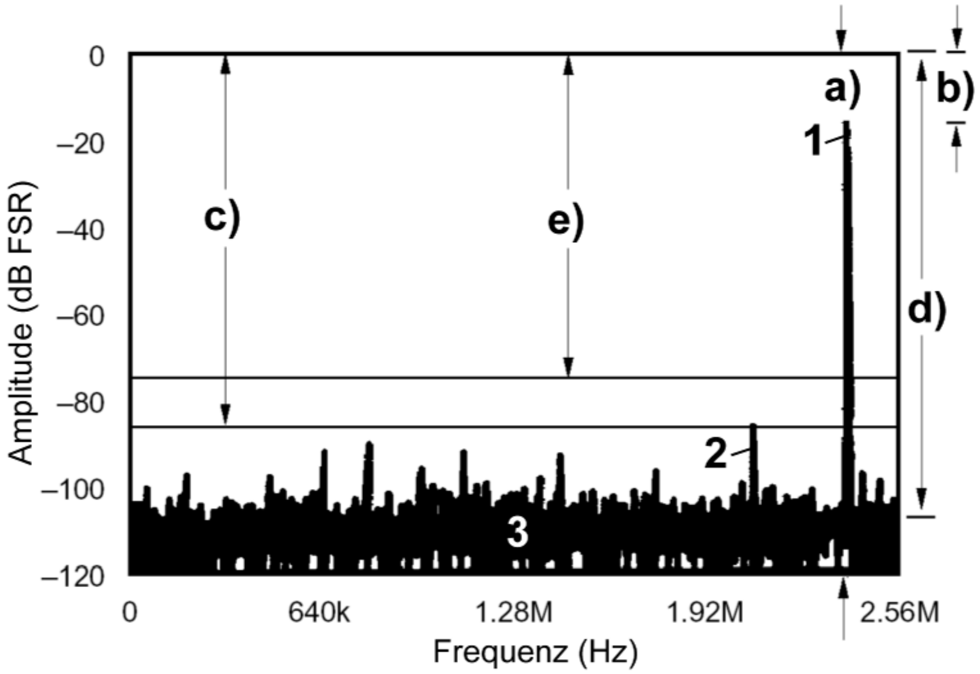


Abbildung 5.3.17 Das Frequenzspektrum einer digitalisierten Sinusschwingung (nach: Texas Instruments)

Erklärung:
1 - die Grundwelle. Im Beispiel mit einer Frequenz von 2,35 MHz. 2 - gelegentliche Störspitzen (Spurious Noise); 3 - das unvermeidliche Grundrauschen (Noise Floor). a) - Amplitude der Grundwelle. Im Idealfall entspricht sie dem gesamten Werte- bzw. Aussteuerbereich (FSR). b) - Aussteuerungsreserve (Headroom); c) - Dynamikbereich ohne Störspitzen (SFDR); d) - Durchschnittspegel des Grundrauschens (Average Noise Floor) e) - Störspannungsabstand (SNR).

Headroom (Aussteuerungsreserve)
In der Praxis erreicht die Grundwelle nicht die volle Amplitude, bleibt also unterhalb FSR. Das ist notwendig, um Signalverfälschungen durch Übersteuern des Wandlers zu vermeiden. Bei FFT-Berechnungen arbeitet man typischerweise mit einem Headroom von - 0,5 dB (Amplitude der Grundwelle ca. 0,94 FSR). Zwecks Veranschaulichung ist der Headroom in Abbildung 5.3.17 übertrieben dargestellt.

SFDR = Spurious Free Dynamic Range (Dynamikbereich ohne Störspitzen)
Der Abstand (in dB) von der Amplitude der Grundwelle zum höchsten Pegel der Störspitzen (diese müssen nicht unbedingt etwas mit den Oberwellen des eigentlichen Signals zu tun haben).

Average Noise Floor (Durchschnittspegel des Grundrauschens)
Er ergibt sich aus dem eigentlichen Rauschen des Wandlers (SNR) und aus einem Anteil, den der Rechenvorgang selbst einbringt (FFT Floor; Abbildung 5.3.18). Der FFT-seitige Anteil:

$$6{,}02n + 1{,}76 + 10 \cdot \lg (m/2); \qquad n = \text{Auflösung},\ m = \text{FFT-Punkte}$$

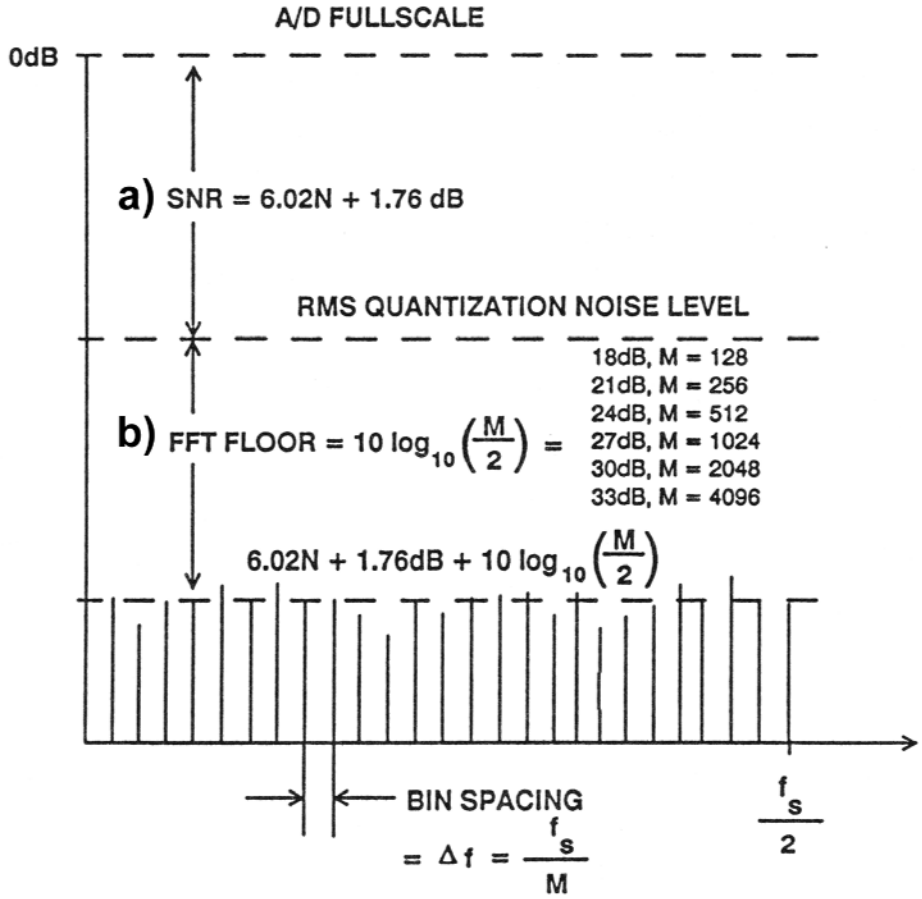


Abbildung 5.3.18 Elementare Rauschanteile. a) Störabstand; b) FFT-Grundrauschen (nach: Analog Devices)

SNR = Signal-to-Noise Ratio (Störabstand)
Die Grundwelle ist das Signal. Weitere Harmonische (Oberwellen) dürften eigentlich gar nicht vorkommen. Wenn sie vorkommen, gelten sie als Störungen. Es liegt somit nahe, das Leistungs-Verhältnis Grundwelle : Oberwellen als Kennwert zu definieren:

$$SNR = \frac{\text{Effektivwert der Grundwelle}}{\text{quadratischer Mittelwert aller weiteren Harmonischen}} = \frac{F_{RMS}}{H_{RMS}}$$

$$SNR = \frac{F_{RMS}}{\sqrt{H_2^2 + H_3^2 + H_4^2 + \ldots}}$$

Einschränkungen:

- ein eventueller Gleichspannungsanteil wird nicht berücksichtigt,
- die weiteren Harmonischen (Oberwellen) werden nur bis zur halben Abtastfrequenz ($f_s/_2$) berücksichtigt (Nyquist-Bandbreite).

SNR und SFDR

- SNR: der Störabstand hängt vor allem von der Anhazhl der Bits ab (Quantisierungsrauschen),
- SFDR: Störspitzen entstehen vor allem infolge von Nichtlinearitäten im Wandler. Somit ist dieser Dynamikbereich weitgehend unabhängig von der Bitanzahl (Abbildung 5.3.19).

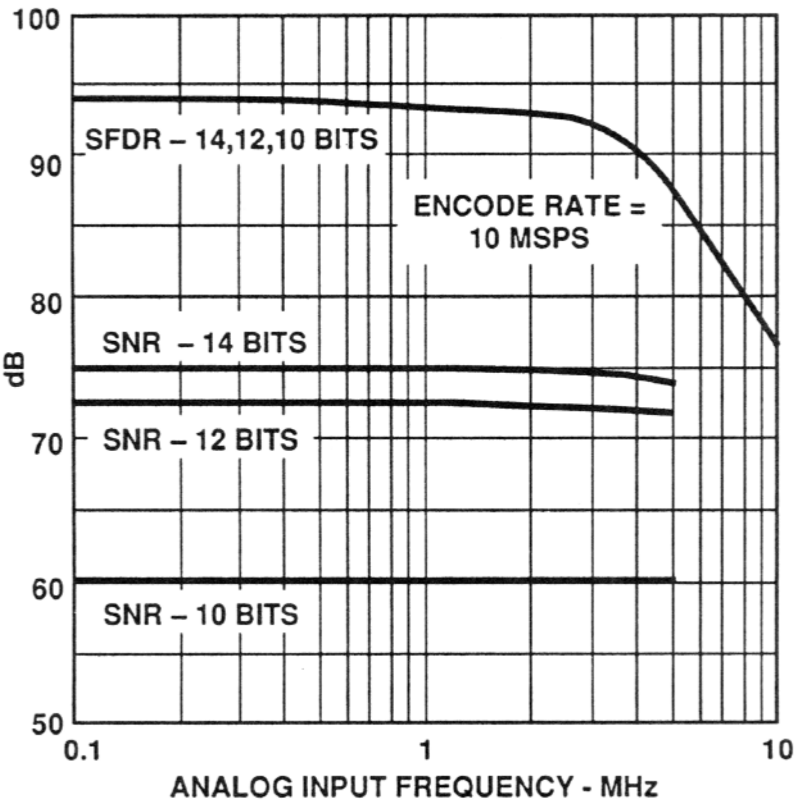


Abbildung 5.3.19 SNR und SFDR anhand eines Beispiels (nach: Analog Devices)

Erklärung:
Es handelt sich um einen 14-Bit-Wandler, wobei 14, 12 oder 10 Bits ausgenutzt werden. Das Weglassen von Bits verschlechtert den Störabstand (SNR), beeinflußt aber den Dynamikbereich (SFDR) praktisch nicht. Mit zunehmender Signalfrequenz werden aber beide Kennwerte schlechter.

Quantisierungsrauschen eines idealen Wandlers mit n Bits Auflösung
Im Bereich von Gleichspannung (DC) bis zur halben Abtastfrequenz $f_s/_2$ (Nyquist-Bandbreite) gilt:

$$U_{\text{QUANT. NOISE}} = \frac{\text{LSB}}{\sqrt{12}} \approx 0{,}289\ \text{LSB}$$

Theoretischer Störabstand eines Analog-Digital-Wandlers mit n Bits Auflösung
Die Effektivwertberechnung auf Grundlag der idealen Wandlerkennlinie (Quantisierungsfehler ± ½ LSB) ergibt:

$$\mathrm{SNR[dB]} = 10 \cdot \lg \frac{\frac{A^2}{2}}{\frac{q^2}{12}}$$

A = Spitzenwert der Sinusspannung. FSR = 2 A. q = 1 LSB. Wir substituieren:

$$q = 1\ \mathrm{LSB} = \frac{2A}{2^n} = \frac{A}{2^{n-1}}$$

$$\mathrm{SNR[dB]} = 10 \cdot \lg \frac{\frac{A^2}{2}}{\frac{A^2}{3 \cdot 2^{2n}}} = 10 \cdot \lg \frac{2^{2n} \cdot 3}{2} = 10 \cdot (2n \lg 2 + \lg \frac{3}{2})$$

$$\mathrm{SNR\ [dB]} = 6{,}02n + 1{,}76$$

Beispielwerte in den Tabellen 5.3.2 und 5.3.3.

Faustformel: $\mathrm{SNR\ [dB]} \approx 6n$

Jedes weitere Bit Auflösung bringt ca. 6 dB zusätzlichen Störabstand.

Berechnung der mindestens erforderlichen Auflösung bei vorgegebenem Störabstand:

$$n = \frac{\mathrm{SNR[dB]} - 1{,}76}{6{,}02}$$

Achtung:
Die Formeln gelten nur für sinusförmige Signale und unter der Annahme, daß es keine Nichtlinearitäten gibt (DNL = 0, INL = 0). Die Nichtlinearität sollte ≤ ½ LSB sein (sonst gefahr von Missning Codes oder gar Nichtmonotoniziät). Der Grenzfall: Reduktion der nutzbaren Auflösung um 1 Bit infolge fehlender (Missing) Codes. Das entspricht einer Verminderung des Störabstandes um 6 dB.

Störabstand (SNR) für Wandler mit ½ LSB Linearitätsfehler (ungünstigster Fall):

$$\mathrm{SNR\ [dB]} = 6{,}02n + 1{,}76 - 6 = 6{,}02n - 4{,}24$$

Bits (n)	SNR (dB)
4	22.32
8	46.4
10	58.44
12	70.48
14	82.52
16	94.56
18	106.6
20	118.64
22	130.68
24	142.72

Tabelle 5.3.2 Der Störspannungsabstand (SBR) in Abhängigkeit von der Auflösung (nach: Texas Instruments)

Auflösung (N Bits)	1 LSB = q	% FS	ppm FS	dB FS (6N)	Quantisierungsrauschen $q/\sqrt{12}$	theoret. Störabstand SNR (dB)
6	32mV	1.56	15625	36	9.2mV	37.9
8	8mV	0.39	3906	48	2.3mV	50.0
10	2mV	0.098	977	60	580µV	62.0
12	500µV	0.024	244	72	144µV	74.0
14	125µV	0.0061	61	84	36µV	86.0
16	31µV	0.0015	15	96	13µV	98.1

Tabelle 5.3.3 Kennwerte typischer A-D-Wandler. In den Beispielen ist FSR = 2,048 V (nach: Analog Devices)

THD = Total Harmonic Distortion (Klirrfaktor)
Der Klirrfaktor kennzeichnet die nichtlinearen Verzerrungen. Er wird angegeben als das Verhältnis des Effektivwertes aller Oberwellen zum Effektivwert der Grundwelle:

$$\mathrm{THD} = \frac{\text{Quadratischer Mittelwert der weiteren Harmonischen}}{\text{Effektivwert der Grundwelle}} = \frac{1}{\mathrm{SNR}}$$

$$\mathrm{THD} = \frac{\sqrt{H_2^2 + H_3^2 + H_4^2 + \ldots}}{F_{RMS}}$$

THD+N = Total Harmonic Distortion + Noise
In diesen Verzerrungskennwert wird das Rauschen eingerechnet:

$$\mathrm{THD} + \mathrm{N} = \frac{\sqrt{H_2^2 + H_3^2 + H_4^2 + \ldots + V_{NOISE}^2}}{F_{RMS}}$$

SINAD = Signal to Noise Ratio with Distortion
SINAD kennzeichnet die Signalqualität anhand der Verzerrungen, die durch Oberwellen und Rauschen verursacht werden:

$$\text{SINAD} = \frac{\text{alle Anteile im Spektrum}}{\text{alle Anteile außer dem Signal (Grundwelle)}}$$

$$\text{SINAD} = \text{SNR} + \text{D} = \frac{\text{Signal} + \text{THD} + \text{N}}{\text{THD} + \text{N}} \text{ ; (D = Distortion)}$$

ENOB = Effective Number of Bits (effektive Bitanzahl)
Ist der Wandler nicht ideal, so ist sein Rauschen größer als das theoretische Minimum. Somit wird die effektive Auflösung geringer; es sind tatsächlich weniger Bits nutzbar, als es die Schaltungsauslegung des Wandlers erwarten läßt.

$$\text{ENOB} = \frac{\text{SINAD[dB]} - 1{,}76}{6{,}02}$$

In den Datenblättern ist oftmals SNR + D angegeben, gelegentlich auch THD+N. Rechengang:

SINAD [dB] = SNR+D [dB] = - (THD+N [dB])

Die Kennwerte sind frequenzabhängig (Abbildungen 5.3.20, 5.3.21).

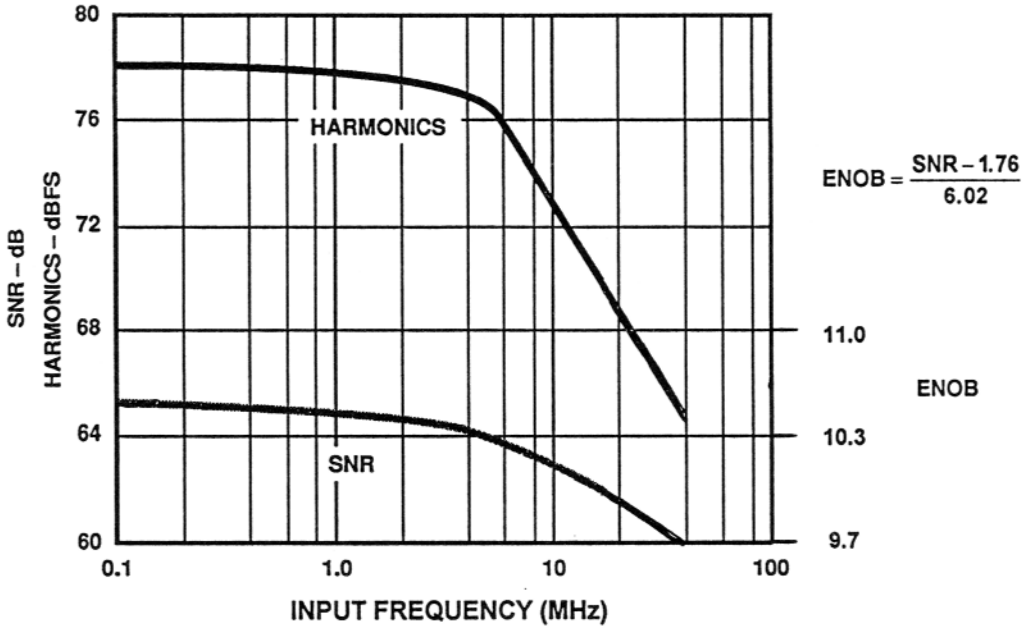


Abbildung 5.3.20 Störabstand (SNR), Klirrfaktor (Harmonics) und effektive Bitanzahl (ENOB) in Abhängigkeit von der Signalfrequenz. 1. Beispiel (nach: Analog Devices)

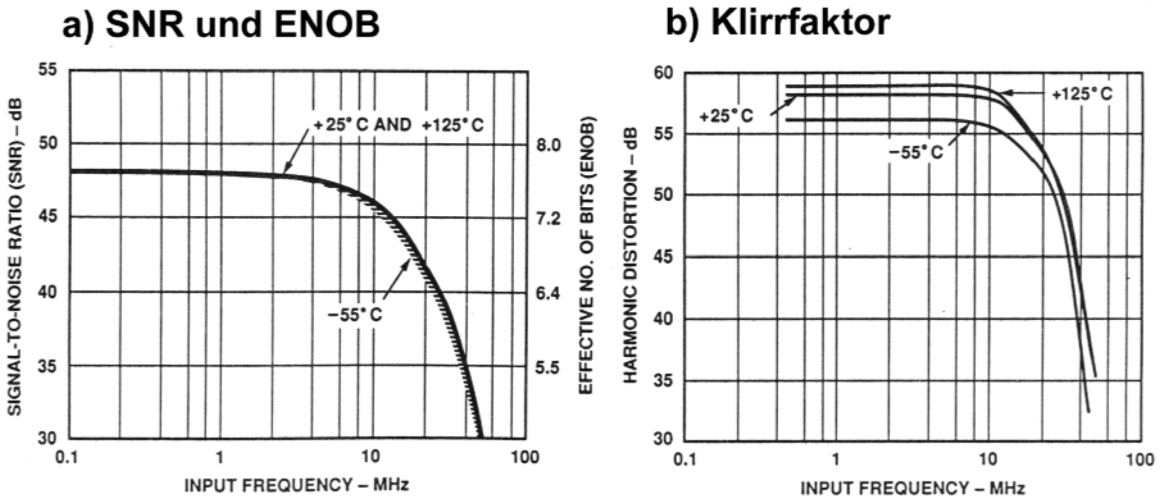
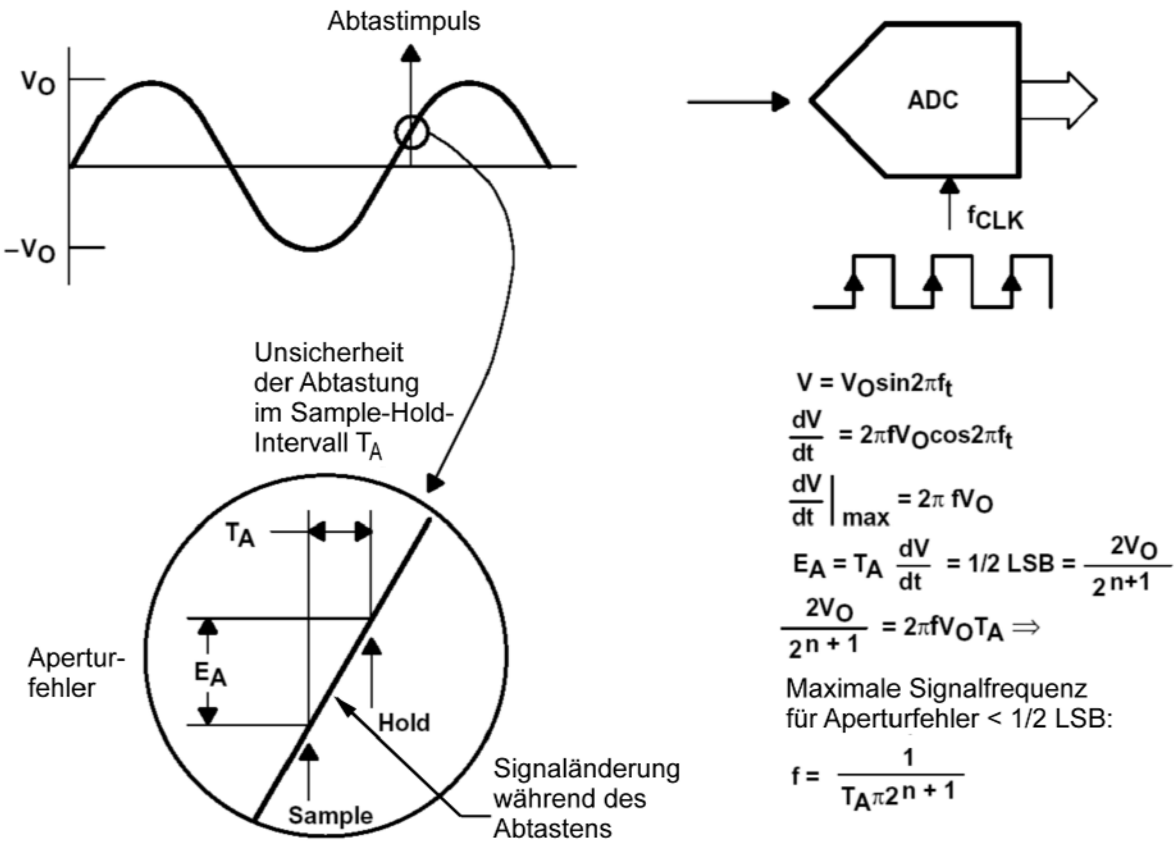


Abbildung 5.3.21 Störabstand (SNR), effektive Bitanzahl (ENOB) und Klirrfaktor in Abhängigkeit von der Signalfrequenz. 2. Beispiel (nach: Analog Devices)

Aperturfehler

Aperaturfehler entstehen beim Abtasten des Signals. Zwischen dem Beginn des Abtastens (Sample) und dem Halten des abgetasteten Wertes (Hold) kann sich der Signalverlauf ändern (Abbildung 5.3.22).



$$V = V_O \sin 2\pi f_t$$

$$\frac{dV}{dt} = 2\pi f V_O \cos 2\pi f_t$$

$$\left.\frac{dV}{dt}\right|_{max} = 2\pi\, f V_O$$

$$E_A = T_A \frac{dV}{dt} = 1/2\ \text{LSB} = \frac{2V_O}{2^{n+1}}$$

$$\frac{2V_O}{2^{n+1}} = 2\pi f V_O T_A \Rightarrow$$

Maximale Signalfrequenz für Aperturfehler < 1/2 LSB:

$$f = \frac{1}{T_A \pi 2^{n+1}}$$

Abbildung 5.3.22 Aperturfehler (nach: Texas Instruments)

Um den Aperturfehler gering zu halten, muß man die Abtastzeit (d. h. die Breite des Sample-Hold-Intervalls) verkürzen (Abbildung 5.3.23). Ggf. ist ein Abtast-und Halteglied vorzusehen.

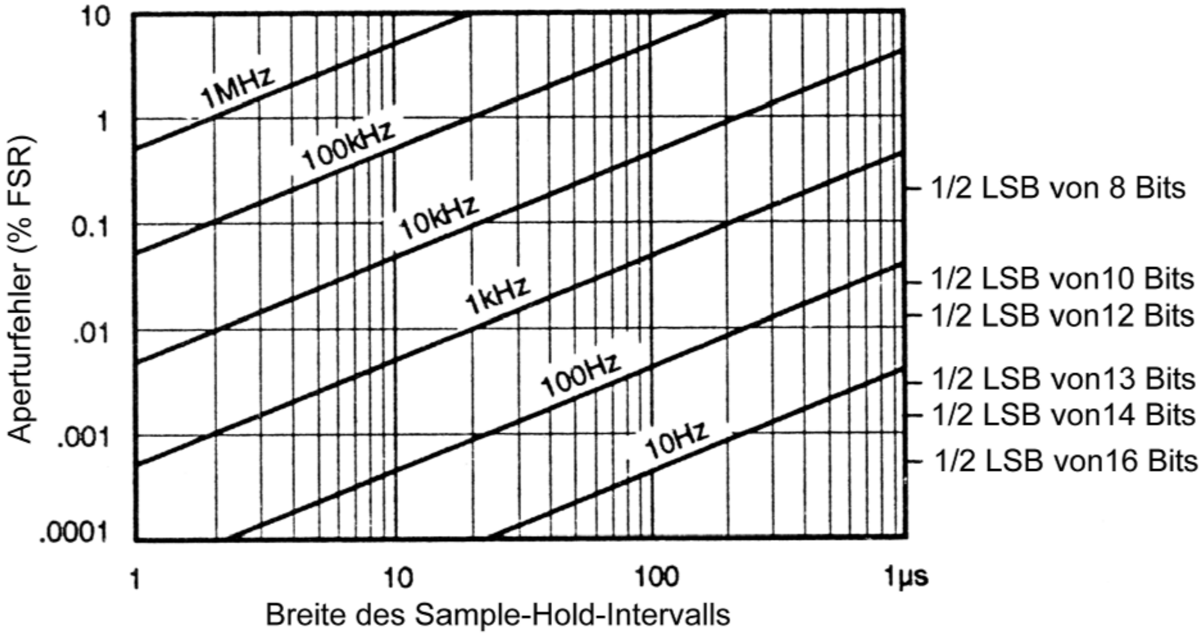


Abbildung 5.3.23 Der Aperturfehler in Abhängigkeit von der Breite des Abtastintervalls (Sample - Hold; nach: Burr-Brown)

Taktunsicherheit (Clock Jitter)

Die Taktfrequenz ist nie vollkommen konstant. Geringfügige Schwankungen führen dazu, daß sich die Abstände der Abtastzeitpunkte verändern und somit weitere Fehler entstehen (Abbildung 5.3.24).

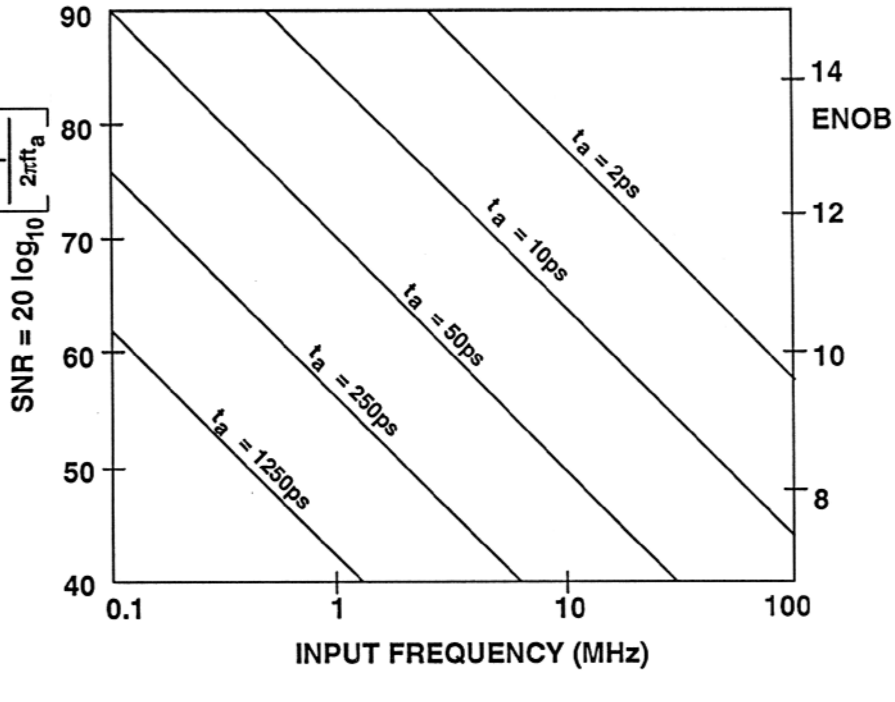


Abbildung 5.3.24 Die Wirkung geringfügiger Schwankungen der Taktperiode auf SNR und ENOB am Beispiel eines 14-Bit-Wandlers (nach: Analog Devices)

5.4 Binäre Codes der Signalwandlung

Hinweis:
Die weiter unten folgende Tabelle 5.4.2 veranschaulicht die einzelnen Codes am Beispiel eines 4-Bit-Wandlers mit einem Aussteuerbereich (Full Scale Range) von 10 V. Im unipolaren Fall (nur eine Polarität) ist - FS (Bereichsanfang) = 0 V und + FS (Bereichsende) = + 10 V, im bipolaren Fall ist - Fs = - 5 V und + Fs = + 5 V.

USB = Unipolar Straight Binary
Das einfachste Prinzip (Tabelle 5.4.1). Darstellung der Zahlenwerte mit natürlichen (vorzeichenlosen) Binärzahlen. Weil das Vorzeichen fehlt, nur für Spannungen einer Polarität nutzbar. Binärwert steigt mit zunehmender Spannung (von 000...00 bis 111...11). Durchfahren des Spannungsbereichs von 0 V bis FS entspricht dem binären Vorwärtszählen.

CSB = Complementary Straight Binary
Wie USB, aber mit Einerkomplementdarstellung auf Grundlage natürlicher (vorzeichenloser) Binärzahlen (Tabelle 5.4.2). Bitweise Negation von USB. Weil das Vorzeichen fehlt, nur für Spannungen einer Polarität nutzbar. Binärwert fällt mit zunehmender Spannung (von 111...11 bis 000...00). Durchfahren des Spannungsbereichs von 0 V bis FS entspricht dem binären Rückwärtszählen.

BOB = Bipolar Offset Binary
Für bipolare Spannungen. Spannungswert 0 wird durch eine Offsetangabe in bezug auf Binärwert 0 dargestellt (Tabelle 5.4.3). Binärwert steigt mit (im arithmetischen Sinne) zunehmender Spannung. Die am meisten negative Spannung (- FS) entspricht 000...00. 0 V entsprechen typischerweise 100...00 (Offset 2^{n-1}). Die am meisten positive Spannung (+ FS) entspricht 111...11. Das höchstwertige Bit (MSB) kann als Vorzeichenangabe interpretiert werden (0 = negativ, 1 = positiv). Durchfahren des Spannungsbereichs von -FS über 0 V bis + FS entspricht dem binären Vorwärtszählen.

COB = Complementary Offset Binary
Wie BOB, aber mit Einerkomplementdarstellung auf Grundlage natürlicher (vorzeichenloser) Binärzahlen (Tabelle 5.4.4). Bitweise Negation von BOB. Für bipolare Spannungen. Spannungswert 0 wird durch eine Offsetangabe in bezug auf Binärwert 0 dargestellt. Binärwert fällt mit (im arithmetischen Sinne) zunehmender Spannung. Die am meisten negative Spannung (- FS) entspricht 111...11. 0 V entsprechen typischerweise 011...11. Die am meisten positive Spannung (+ FS) entspricht 000...00. Das höchstwertige Bit (MSB) kann als Vorzeichenangabe interpretiert werden (0 = positiv, 1 = negativ). Durchfahren des Spannungsbereichs von -FS über 0 V bis + FS entspricht dem binären Rückwärtszählen.

BTC = Binary Two's Complement
Zweierkomplementdarstellung mit vorzeichenbehafteten (ganzen) Binärzahlen (Tabelle 5.4.5). Für bipolare Spannungen. Spannungswert 0 entspricht 000...00. Das höchstwertige Bit (MSB) ist das Vorzeichen (0 = negativ, 1 = positiv). Im positiven Bereich steigt der Binärwert mit zunehmender Spannung (von 0 V bis + FS gemäß 000...00, 000...01 usw. bis 111...11). Im negativen Bereich sinkt der Binärwert mit abnehmender Spannung (von 0 V bis - FS gemäß 000...00, 111...11 usw. bis 100...00). - FS = 100...00; - 1 = 111...11; + 1 0 000...01; + FS =

011...11. Durchfahren des Spannungsbereichs von - FS bis 0 V entspricht dem binären Vorwärtszählen bei stets gesetztem MSB, Durchfahren des Spannungsbereichs von 0 V bis + FS entspricht dem binären Vorwärtszählen bei stets gelöschtem MSB. BTC ist die typische Zahlendarstellung der Mikrocontroller.

CTC = Complementary Two's Complement
Wie BTC, aber bitweise invertiert (Tabelle 5.4.6). Für bipolare Spannungen. Spannungswert 0 entspricht 111...11. Das höchstwertige Bit (MSB) ist das Vorzeichen (0 = positiv, 1 = negativ). Im positiven Bereich sinkt der Binärwert mit zunehmender Spannung (von 0 V bis + FS gemäß 111...11, 111...10 usw. bis 100...00). Im negativen Bereich steigt der Binärwert mit abnehmender Spannung (von 0 V bis - FS gemäß 111...11, 000...00 usw. bis 011...11). -FS = 011...11; - 1 = 000...01; + 1 = 111...11; + FS = 100...00. Durchfahren des Spannungsbereichs von - FS bis 0 V dem binären Rückwärtszählen bei stets gelöschtem MSB, Durchfahren des Spannungsbereichs von 0 V bis + FS entspricht dem binären Rückwärtszählen bei stets gesetztem MSB.

Wertewandlung durch bitweises Invertieren:

- USB => CSB und CSB => USB,
- BOB => COB und COB => BOB,
- BTC => CTC und CTC => BTC.

Wertewandlung durch Invertieren des Vorzeichenbits (MSB):

- BOB => BTC und BTC => BOB,
- COB => CTC und CTC => COB.

Die am meisten verwendeten Codes: USB und BOB (Tabelle 5.4.1).

DEFINITION	OUTPUT DIGITAL CODE	USB CODE	BOB*) CODE
	MSB LSB		
+Full Scale	111....11	$+V_{FSR} - 1/2LSB$	$\frac{+V_{FSR}}{2} - 1/2LSB$
Mid Scale	100....00	$+V_{FSR}/2$	Zero
–Full Scale	000....00	+1/2LSB	$\frac{-V_{FSR}}{2} + 1/2LSB$
1 LSB		$\frac{V_{FSR}}{2^n}$	$\frac{\pm V_{FSR}}{2^n}$
*): BTC = BOB mit invertiertem MSB (Vorzeichen)			

Tabelle 5.4.1 Die am meisten verwendeten Codes (nach: Teaxs Instruments)

a) USB

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
Zero	0000		0.000	0.3125
+1 V_{LSB}	0001	0.3125	0.625	0.9375
	0010	0.9375	1.250	1.5625
	0011	1.5625	1.875	2.1875
1/4 FSR	0100	2.1875	2.500	2.8125
	0101	2.8125	3.125	3.4375
	0110	3.4375	3.750	4.0625
	0111	4.0625	4.375	4.6875
1/2 FSR	1000	4.6875	5.000	5.3125
	1001	5.3125	5.625	5.9375
	1010	5.9375	6.250	6.5625
	1011	6.5625	6.875	7.1875
3/4 FSR	1100	7.1875	7.500	7.8125
	1101	7.8125	8.125	8.4375
	1110	8.4375	8.750	9.0625
+FS	1111	9.0625	9.375	

b) CSB

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
Zero	1111		0.000	0.3125
+1V_{LSB}	1110	0.3125	0.625	0.9375
	1101	0.9375	1.250	1.5625
	1100	1.5625	1.875	2.1875
1/4 FSR	1011	2.1875	2.500	2.8125
	1010	2.8125	3.125	3.4375
	1001	3.4375	3.750	4.0625
	1000	4.0625	4.375	4.6875
1/2 FSR	0111	4.6875	5.000	5.3125
	0110	5.3125	5.625	5.9375
	0101	5.9375	6.250	6.5625
	0100	6.5625	6.875	7.1875
3/4 FSR	0011	7.1875	7.500	7.8125
	0010	7.8125	8.125	8.4375
	0001	8.4375	8.750	9.0625
+FS	0000	9.0625	9.375	

c) BOB

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
–FS	0000		–5.000	–4.6875
	0001	–4.6875	–4.375	–4.0625
	0010	–4.0625	–3.750	–3.4375
	0011	–3.4375	–3.125	–2.8125
1/2 –FS	0100	–2.8125	–2.500	–2.1875
	0101	–2.1875	–1.875	–1.5625
	0110	–1.5625	–1.250	–0.9375
BPZ – 1V_{LSB}	0111	–0.9375	–0.625	–0.3125
BPZ	1000	–0.3125	0.000	+0.3125
BPZ + 1V_{LSB}	1001	+0.3125	+0.625	+0.9375
	1010	+0.9375	+1.250	+1.5625
	1011	+1.5625	+1.875	+2.1875
1/2 +FS	1100	+2.1875	+2.500	+2.8125
	1101	+2.8125	+3.125	+3.4375
	1110	+3.4375	+3.750	+4.0625
+FS	1111	+4.0625	+4.375	

d) COB

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
–FS	1111		–5.000	–4.6875
	1110	–4.6875	–4.375	–4.0625
	1101	–4.0625	–3.750	–3.4375
	1100	–3.4375	–3.125	–2.8125
1/2 –FS	1011	–2.8125	–2.500	–2.1875
	1010	–2.1875	–1.875	–1.5625
	1001	–1.5625	–1.250	–0.9375
BPZ – 1V_{LSB}	1000	–0.9375	–0.625	–0.3125
BPZ	0111	–0.3125	0.000	+0.3125
BPZ + 1V_{LSB}	0110	+0.3125	+0.625	+0.9375
	0101	+0.9375	+1.250	+1.5625
	0100	+1.5625	+1.875	+2.1875
1/2 +FS	0011	+2.1875	+2.500	+2.8125
	0010	+2.8125	+3.125	+3.4375
	0001	+3.4375	+3.750	+4.0625
+FS	0000	+4.0625	+4.375	

e) BTC

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
–FS	1000		–5.000	–4.6875
	1001	–4.6875	–4.375	–4.0625
	1010	–4.0625	–3.750	–3.4375
	1011	–3.4375	–3.125	–2.8125
1/2 –FS	1100	–2.8125	–2.500	–2.1875
	1101	–2.1875	–1.875	–1.5625
	1110	–1.5625	–1.250	–0.9375
BPZ – 1V_{LSB}	1111	–0.9375	–0.625	–0.3125
BPZ	0000	–0.3125	0.000	+0.3125
BPZ + 1V_{LSB}	0001	+0.3125	+0.625	+0.9375
	0010	+0.9375	+1.250	+1.5625
	0011	+1.5625	+1.875	+2.1875
1/2 +FS	0100	+2.1875	+2.500	+2.8125
	0101	+2.8125	+3.125	+3.4375
	0110	+3.4375	+3.750	+4.0625
+FS	0111	+4.0625	+4.375	

f) CTC

MNEMONIC	DIGITAL CODE	V_{t-}	V_{CODE}	V_{t+}
–FS	0111		–5.000	–4.6875
	0110	–4.6875	–4.375	–4.0625
	0101	–4.0625	–3.750	–3.4375
	0100	–3.4375	–3.125	–2.8125
1/2 –FS	0011	–2.8125	–2.500	–2.1875
	0010	–2.1875	–1.875	–1.5625
	0001	–1.5625	–1.250	–0.9375
BPZ – 1V_{LSB}	0000	–0.9375	–0.625	–0.3125
BPZ	1111	–0.3125	0.000	+0.3125
BPZ + 1V_{LSB}	1110	+0.3125	+0.625	+0.9375
	1101	+0.9375	+1.250	+1.5625
	1100	+1.5625	+1.875	+2.1875
1/2 +FS	1011	+2.1875	+2.500	+2.8125
	1010	+2.8125	+3.125	+3.4375
	1001	+3.4375	+3.750	+4.0625
+FS	1000	+4.0625	+4.375	

Tabelle 5.4.2 Codebeispiele (nach: Burr-Brown)

Wandler und Mikrocontroller
Das Hausformat der heutigen Mikrocontroller und Prozessoren ist die Zweierkomplementdarstellung.

Eingabe USB: Je nach Verarbeitungsbreite und anzuwendenden Operationen ggf. in Zweierkomplement wandeln. Ggf. Nullerweiterung.

Eingabe CSB: Alle Bits invertieren. Dann weiter wie USB.

Eingabe BOB: Vorzeichenbit invertieren, alles andere so lassen.

Eingabe COB: Alle Bits invertieren. Dann weiter wie BOB.

Eingabe BTC: An sich nichts zu tun. Ggf. vorzeichengerechte Erweiterung.

Eingabe CTC: Alle Bits invertieren. Dann weiter wie BTC.

Ausgabe USB: Nichts besonderes zu tun.

Ausgabe CSB: Alle Bits invertieren. Dann weiter wie USB.

Ausgabe BOB: Vorzeichenbit (gemäß Stellenzahl des Wandlers) invertieren, alles andere so lassen.

Ausgabe COB: Alle Bits invertieren. Dann weiter wie BOB.

Ausgabe BTC: Nichts besonderes zu tun.

Ausgabe CTC: Alle Bits invertieren. Dann weiter wie BTC.

Typische Stellenzahlen von Wandlern: 6 - 8 - 10 - 12 -14 -16 - 20 - 24 Bits.

Typische Verarbeitungsbreiten von Controllern: 8 - 16 - 32 Bits.

Gelegentlich ist es sinnvoll, niedrigstwertige Bits des Wandles gar nicht zu verwenden (Stichworte: ENOB, Monotonizitätsfehler).

Die zu den benutzten Wandlerbits jeweils passende Verarbeitungsbreite wählen (ggf. aufrunden).

Natürliche Binärzahlen
Natürliche Zahlen sind vorzeichenlos[*)]. Der niedrigste Wert ist Null. Der gesamteWertebereich einer natürlichen Binärzahl x aus n Bits (Abbildung 5.4.1) ist gegeben durch:

$$0 \leq x \leq 2^n - 1$$

Beispiel: eine natürliche Binärzahl aus 8 Bits (n = 8):

- niedrigster Wert = 0 = 0000 0000B = 0H,
- höchster Wert = 2^8-1 = 255 = 1111 1111B = FFH.

*): engl. Natural bzw. Unsigned Numbers. Das Vorzeichen - hier feht es - heißt Sign.

15	0
Wert	

Abbildung 5.4.1 Beispiel einer natürlichen (vorzeichenlosen) Binärzahl (16 Bits)

Elementare Formate
In Tabelle 5.4.3 sind Formate natürlicher Binärzahlen angegeben, die heutzutage gebräuchlich sind (die Zahlen belegen gemäß ihrer Länge Bytes, Worte usw.).

Länge		**Größter Wert**	
in Bits	**in Bytes**		
8	1	2^8 - 1 =	255
16	2	2^{16} - 1 =	65 535
32	4	2^{32} - 1 =	4 294 967 295 (4G - 1)
64	8	2^{64} - 1 =	18,4 · 10^{18} (18,4 Trillionen*)

*): 10^9 = 1 Milliarde (im Englischen: 1 Billion); 10^{18} = 1 Trillion (im Englischen: 1 Quintillion); 18,4 Trillionen = 18 Milliarden Milliarden. Ganz genau: 2^{64} - 1 = 18 446 744 073 709 551 615

Tabelle 5.4.3 Natürliche Binärzahlen als elementare Datentypen

In der Praxis kommen auch natürliche Binärzahlen abweichender Länge vor (z. B. 14 Bits, 20 Bits usw.). Will man solche Zahlen in Programmen verarbeiten, so müssen sie *rechtsbündig* aufbereitet werden. Dazu muß das niedrigstwertige Bit in Bit 0 des niedrigstwertigen Bytes stehen. Von der höchsten Stelle an bis zum Ende des jeweiligen Verarbeitungs-Formates (z. B. 32 Bits) sind Nullen aufzufüllen (Nullerweiterung; Zero Extend).

Ganze Binärzahlen

Ganze Zahlen*) (Abbildung 5.4.2) haben ein Vorzeichen (+ oder -). Das Vorzeichen belegt ein Bit, und zwar jeweils das höchstwertige (Most Significant Bit MSB).

*): Fachbegriff: Integer-Zahlen oder kurz Integers. Auch: Signed Numbers.

15	14 ... 0
S	Wert

Abbildung 5.4.2 Beispiel einer ganzen Binärzahl (16 Bits). S = Vorzeichen (Sign)

Zahlendarstellung
In allen modernen Architekturen wird die Zweierkomplementdarstellung verwendet. Das Vorzeichen wird folgendermaßen codiert:

- 0: positiv (+),
- 1: negativ (-).

Die verbleibenden Bits repräsentieren den Wert der Zahl, allerdings *nicht* unabhängig vom Vorzeichen.

Eine positive Zahl (einschließlich der Zahl 0) wird genau so dargestellt wie eine natürliche Binärzahl. Da die höchstwertige Stelle vom Vorzeichenbit (hier = 0) belegt ist, hat eine n Bits lange ganze Binärzahl x einen positiven Wertebereich von $0 \leq x \leq 2^{n-1} - 1$.

Eine negative Zahl hat in der höchstwertigen Stelle (Vorzeichenbit) eine Eins. Der Zahlenwert wird als Zweierkomplement angegeben ($-x \hat{=} 2^n - x$).

Der gesamte Wertebereich einer ganzen Binärzahl z aus n Bits ist gegeben durch:

$$-(2^{n-1}) \leq x \leq 2^{n-1} - 1.$$

Beispiel: eine natürliche Binärzahl aus 8 Bits (n = 8):

- niedrigster Wert (kleinste negative Zahl) = $-(2^7) = -128$ = 1000 0000B = 80H,
- Wert - 1 (größte negative Zahl) = 1111 1111B = FFH,
- Wert 0 = 0000 0000B = 00H,
- Wert + 1 (kleinste positive Zahl) = 0000 0001B = 01H,
- höchster Wert (größte positive Zahl) = $2^7 - 1 = 127$ = 0111 1111B = 7FH.

Kleiner/größer - eine Wiederholung aus der Elementarmathematik
Eine Zahl ist um so größer, je näher sie an plus Unendlich ($+\infty$) liegt; sie ist um so kleiner, je näher sie an minus Unendlich ($-\infty$) liegt (im besonderen ist eine negative Zahl um so kleiner, "je negativer" sie ist). Beispiele: - 8 ist kleiner als - 3 ($-8 < -3$) und - 2 ist kleiner als + 2. Umkehrung: - 3 ist größer als - 8 ($-3 > -8$), + 2 ist größer als - 2 ($+2 > -2$). $-\infty$ ist kleiner als $+\infty$. Jede negative Zahl ist kleiner als Null.

Elementare Formate
In Tabelle 5.4.4 sind Formate ganzer Binärzahlen angegeben, die heutzutage gebräuchlich sind (die Zahlen belegen gemäß ihrer Länge Bytes, Worte usw.).

Länge		Größte Werte	
in Bits	in Bytes	negativ	positiv
8	1	$-2^7 =$ - 128	$2^7-1 =$ 127
16	2	$-2^{15} =$ - 32 768	$2^{15}-1 =$ 32 767
32	4	$-2^{31} =$ - 2 147 483 648	$2^{31}-1 =$ 2 147 483 647
64	8	$-2^{63} =$ $\approx -9{,}2 \cdot 10^{18}$	$2^{63}-1 =$ $\approx 9{,}2 \cdot 10^{18}$

Ganz genau: $-2^{63} = -9\,223\,372\,036\,854\,775\,808$; $2^{63} - 1 = 9\,223\,372\,036\,854\,775\,807$

Tabelle 5.4.4 Ganze Binärzahlen (Integer-Zahlen) als elementare Datentypen

Ganze und natürliche Binärzahlen beim Addieren und Subtrahieren

Addition und Subtraktion laufen für natürliche und ganze Binärzahlen gleichermaßen ab; die Unterscheidung kommt einzig dadurch zustande, wie Operanden und Resultat interpretiert werden (gleiche Operandenbitmuster liefern das gleiche Ergebnisbitmuster, gleichgültig ob das höchstwertige Bit als Vorzeichen oder als Ziffernstelle interpretiert wird). Verschiedene Additions- und Subtraktionsbefehle für natürliche und ganze Binärzahlen sind deshalb nicht notwendig, wohl aber verschiedene Multiplikations- und Divisionsbefehle.

Vergleichen zweier Binärzahlen

Wir vergleichen 2 Binärzahlen A, B miteinander, indem wir sie voneinander subtrahieren und bestimmte Bedingungen auswerten. Diese Bedingungen werden in den üblichen Prozessoren als Flagbits bzw. Bedingungscodes gespeichert (Tabelle 5.4.5). Sie können zwecks Programmverzweigung abgefragt werden (in JMP- bzw. BRANCH-Befehlen). Aus den Tabellen 5.4.6 und 5.4.7 sind die typischen Verzweigungsbedingungen ersichtlich.

Typische Bezeichnung	Benennung	Bedeutung
ZF	Zero Flag	Ergebnis = 0
CF	Carry Flag	Addition: Ausgangsübertrag = 1, Subtraktion: Auslegung 1: Ausgangsübertrag = 1*) , Auslegung 2: Ausgangsübertrag = 0*)
OF	Overflow Flag	Overflow = 1
SF	Sign Flag (auch: Negative Flag)	Vorzeichen (= höchstwertige Bitposition) = 1. Wert ist negativ

*): siehe Text in Anschluß an Tabelle 5.4.6

Tabelle 5.4.5 Flagbits

.

Vergleichen natürlicher Binärzahlen. Rechengang: A - B			
Vergleichsaussage	**Bedingung[1)]**	**Flagbits[2)]**	**typische Bezeichnung[*3)]**
A = B	Ergebnis = 0 (sowie Ausgangsübertrag)	ZF = 1	Equal
A ≠ B	Ergebnis ≠ 0	ZF = 0	Not Equal
Auslegung 1: CF = Ausgangsübertrag der Zweierkomplementrechnung[2)]			
A < B	kein Ausgangsübertrag (≙ Borgen)	CF = 0	Below
A > B	Ergebnis ≠ 0 und Ausgangsübertrag (≙ kein Borgen)	$\overline{ZF} \cdot CF = 1$ bzw. $ZF \vee \overline{CF} = 0$	Above
A ≤ B	Ergebnis = 0 oder kein Ausgangsübertrag (≙ Borgen)	$ZF \vee \overline{CF} = 1$	Below or Equal
A ≥ B	Ausgangsübertrag (≙ kein Borgen)	CF = 1	Above or Equal
Auslegung 2: CF = invertierter Ausgangsübertrag der Zweierkomplementrechnung[2)]			
A < B	kein Ausgangsübertrag (≙ Borgen ≙ CF = 1)	CF = 1	Below
A > B	Ergebnis ≠ 0 und Ausgangsübertrag (≙ kein Borgen≙ CF = 0)	$\overline{ZF} \cdot \overline{CF} = 1$ bzw. $ZF \vee CF = 0$	Above
A ≤ B	Ergebnis = 0 oder kein Ausgangsübertrag (≙ Borgen ≙ CF = 1)	$ZF \vee CF = 1$	Below or Equal
A ≥ B	Ausgangsübertrag (≙ kein Borgen≙ CF = 0)	CF = 0	Above or Equal

1): Zweierkomplement-Arithmetik; 2) siehe den folgenden Text; 3): in Befehlsbeschreibungen

Tabelle 5.4.6 Vergleichen natürlicher (vorzeichenloser) Binärzahlen

Ausgangsübertrag und Carry Flag (CF)
Es gibt zwei Auslegungen:

1. die naive Auslegung: das Carry Flag entspricht dem Ausgangsübertrag der Zweierkomplement-Arithmetik. Es schaltet demzufolge gemäß Tabelle 2.4. Das Ergebnis liegt außerhalb des Wertebereichs, wenn beim Addieren CF = 1 gebildet wird und beim Subtrahieren CF = 0. Beispiele: einige der PIC-Mikrocontroller der Fa. Microchip. CF = 0 entspricht einem Borgen von der nächst-höheren Binärstelle.
2. die Vorzugsauslegung: das Carry Flag wird in Abhängigkeit von der jeweiligen Rechenoperation gestellt (viele Controller und Prozessoren sind so ausgelegt: Atmel AVR, Intel x86 usw.):

- Addieren: Carry Flag = Ausgangsübertrag (gemäß Zweierkomplement-Arithmetik),
- Subtrahieren: das Carry Flag soll das Borgen aus der nächst-höheren Binärstelle kennzeichnen. Zu borgen ist aber nur dann, wenn der Minuend kleiner ist als der Subtrahend. (Beim Rechengang A - B also dann, wenn A < B.) Beim Rechnen im Zweierkomplement entsteht aber ein Ausgangsübertrag nur dann, wenn A ≥ B. Das Carry Flag entspricht somit beim Subtrahieren dem invertierten Ausgangsübertrag. Der Vorteil: beim Rechnen mit natürlichen (vorzeichenlosen) Binärzahlen zeigt CF = 1 stets (sowohl beim Addieren als auch beim Subtrahieren) an, daß das Resultat außerhalb des Wertebereichs liegt

Vergleichen ganzer Binärzahlen. Rechengang: A - B			
Vergleichsaussage	**Bedingung**	**Flagbits*)**	**typische Bezeichnung**)**
A = B	Ergebnis = 0	ZF = 1	Equal
A ≠ B	Ergebnis ≠ 0	ZF = 0	Not Equal
A < B	Ergebnis negativ und Überlauf oder Ergebnis positiv und kein Überlauf	SF ≠ OF SF ⊕ OF = 1	Less
A > B	Ergebnis ≠ 0 und Ergebnis negativ und kein Überlauf oder Ergebnis positiv und Überlauf	ZF = 0 und SF = OF ZF ∨ (SF ⊕ OF) = 0	Greater
A ≤ B	Ergebnis = 0 oder Ergebnis negativ und Überlauf oder Ergebnis positiv und kein Überlauf	ZF oder SF ≠ OF ZF ∨ (SF ⊕ OF) = 1	Less or Equal
A ≥ B	Ergebnis negativ und kein Überlauf oder Ergebnis positiv und Überlauf	SF = OF SF ⊕ OF = 0	Greater or Equal

*) siehe Tabelle 5.4.4; **): in Befehlsbeschreibungen

Tabelle 5.4.7 Vergleichen ganzer (vorzeichenbehafteter) Binärzahlen

Erweiterung vorzeichenloser Binärzahlen
Beispiel: 14 Bits USB auf 16 Bits. Durch Einfügen von Nullen in die hinzugekommenen Bitpositionen (Nullerweiterung, Zero Extend (Abbildung 5.4.3)).

Erweiterung vorzeichenbehafteter Binärzahlen
Beispiel: 14 Bits BTC auf 16 Bits. Durch Eintragen des Vorzeichens in alle hinzugekommenen Bitpositionen (Vorzeichenerweiterung, Sign Extend (Abbildung 5.4.4).

								A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0	A	B	C	D	E	F	G	H

Abbildung 5.4.3 Prinzip der Nullerweiterung (8 Bits vorzeichenlos auf 16 Bits)

								S	A	B	C	D	E	F	G
S	S	S	S	S	S	S	S	S	A	B	C	D	E	F	G

Abbildung 5.4.4 Prinzip der Vorzeichenerweiterung (8 Bits ganzzahlig auf 16 Bits). S = Vorzeichen

Gotcha: Stellenwert und Bitnumerierung

Stellenwert
In Binärzahlen hat jedes Bit einen Stellenwert, genau wie jede Ziffer in einer Dezimalzahl. In diesem Sinne spricht man allgemein von nieder- und höherwertigen Bits. Zeichnerisch wird das niedrigstwertige Bit (Least Significant Bit, LSB) ganz rechts dargestellt, das höchstwertige Bit (Most Significant Bit, MSB) hingegen ganz links, in völliger Entsprechung zur üblichen Zahlenschreibweise.

Bitnumerierung
Die einzelnen Bitpositionen des Wanders und des Controllers werden typischerweise fortlaufend numeriert, wobei man meist mit 0 beginnt, gelegentlich mit 1.

Wie hängen Stellenwert und Bitnummer zusammen?
Es gibt zwei Möglichkeiten (Abbildung 5.4.5):

1. Rechtsadressierung (Little Endian)
Das niedrigstwertige Bit hat die niedrigste Bitnummer (zumeist Null). Das höchstwertige Bit eines n Bits langen Wortes hat - wenn die Zählung von Null an beginnt - die Nummer n-1, im Byte (8 Bits) also die Nummer 7, im 32-Bit-Wort die Nummer 31 usw.

2. Linksadressierung (Big Endian)
Das höchstwertige Bit hat die niedrigste Bitnummer (zumeist Null). Das niedrigstwertige Bit eines n Bits langen Wortes hat - wenn die Zählung von Null an beginnt - die Nummer n-1, im Byte (8 Bits) also die Nummer 7, im 32-Bit-Wort die Nummer 31 usw.

Beispiele:

- Little Endian: die Intel-Typen, die PICs, die AVRs usw.,
- Big Endian: die Motorola-Typen, die IBM-Mainframes, PowerPC, JVM (Java Virtual Machine) usw.

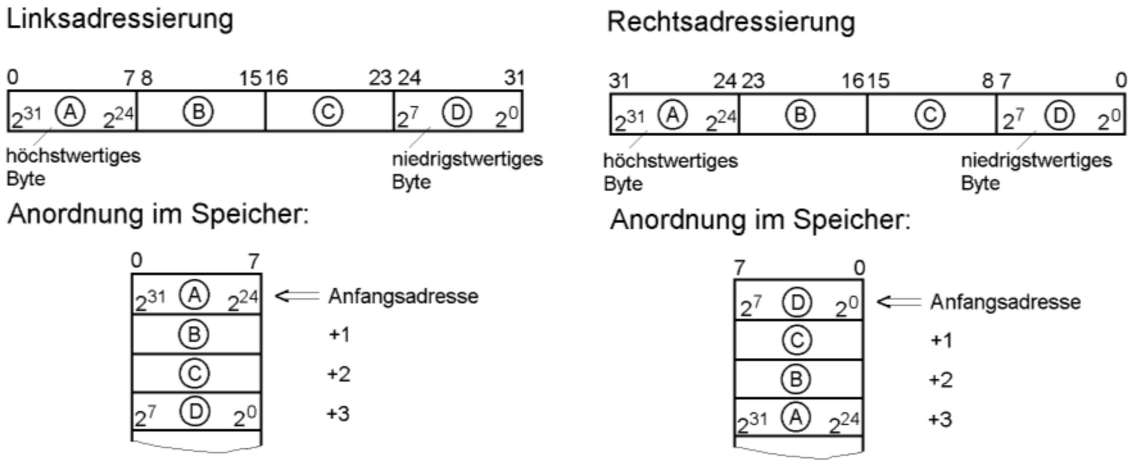


Abbildung 5.4.5 Links- und Rechtsadressierung am Beispiel von 32-Bit-Worten

Wandler mit seriellem Interface
Schieberichtung ansehen. Welches Bit kommt als erstes (A-D) bzw. wird als erstes erwartet (D-A)? *Beispiele:*

- serielle Schnittstelle: LSB zuerst,
- SPI: in vielen Mikrocontrollern umsteuerbar,
- I^2C, Microwire: MSB zuerst.

Wandler mit parallelem Interface

1. Wandler und Controller haben gleiche Adressierungsweise: Datenanschlüsse 1:1 verbinden. Wenn der Wandler über zwei oder mehr Ports angeschlossen werden muß: die mit den niedrigeren Bitpositionen beschalteten Ports betreffen die niederwertigen Binärstellen.

2. Wandler und Controller haben unterschiedliche Adressierungsweise: Datenanschlüsse über Kreuz verbinden (bei einem 8-Bit-Wandler Bit 0 mit Bit 7, Bit 1 mit Bit 6 usw.). Wenn der Wandler über zwei oder mehr Ports angeschlossen werden muß: die mit den niedrigeren Bitpositionen beschalteten Ports betreffen die höherwertigen Binärstellen.

Ggf. muß die Bitreihenfolge mittels Software umgestellt werden (Links- oder Rechtsrotieren gemäß Verarbeitungsbreite - 1; Nutzung entsprechender Spezialbefehle).

Gotcha: Natürliche (vorzeichenlose) Binärzahlen gemäß Verarbeitungsbreite
Fallbeispiel: 8-Bit-Wandler, USB und 8 Bits Verarbeitungsbreite. Geht es nur ums Addieren. Subtrahieren und Vergleichen, ist keine Erweiterung der Verarbeitungsbreite erforderlich.

Achtung:

1. Vergleichsbedingungen gemäß Tabelle 5.4.5,
2. Konstantenangaben. Aufpassen: es kann sein, daß der Compiler oder Assembler nur

ganze Binärzahlen kennt. Im Wertebereich von 0 bis 2^{n-1}-1 funktioniert alles. Um Bitmuster im Wertebereich von 2^{n-1} bis 2^n-1 darzustellen, sind aber negative Zahlen anzugeben. Beispiel (für 8 Bits Verarbeitungbreite): für 128: - 128, für 255: - 1. Um einen Wert n ($n \geq 2^{n-1}$) anzugeben, eine Konstante n - 2^n definieren. Abhilfe: Konstanten hexadezimal oder binär darstellen (mühevoll).

Gotcha: Rechnen mit kurzen Binärzahlen im Zweierkomplement

Viele Prozessoren unterstützen die Verarbeitung kürzerer Operanden (Beispiel: Verknüpfung eines Bytes mit einem 16-Bit-Wort). Dabei wir aber der kürzere Operand typischerweise *vorzeichengerecht* erweitert, unabhängig davon, welche Bedeutung er vom Programmierer erhalten hat. Beispiel: wir rechnen USB-Angaben eines 8-Bit-Wandlers, die ein byte belegen. Wenn wir ein solche Bytes mit 16-Bit- oder 32-Bit-Zahlen verknüpfen wollen und einfach die entsprechenden Befehle hinschreiben, so macht die Vorzeichenerweiterung aus allen Werten über 127 negative Werte! *Abhilfe*: die Angaben des Wandlers werden *vor* dem eigentlichen Rechnen auf die jeweilige Verarbeitungsbreite gebracht (die Erweiterung wird also ausprogrammiert - was nicht besonders schwierig ist; in manchen Architekturen gibt es eigens Befehle zum Laden ohne Vorzeichenerweiterung. Beispiel (IA-32): “Laden mit Nullerweiterung” (MOVZX).

Gotcha: Einfache Rundungsverfahren

1. Löschen niedrigstwertiger Bitpositionen

Eine bestimmte Anzahl der niedrigstwertigen Bits wird zu Null gemacht.

2. Abschneiden niedrigstwertiger Bitpositionen

Eine bestimmte Anzahl der niedrigstwertigen Bits wird abgeschnitten (durch arithmetisches Rechtsverschieben*) der Binärzahl entfernt).

*): mit Einfügen des Vorzeichens in die links freiwerdenden Bitpositionen.

Handelt es sich um positive Zahlen, so bewirken beide Verfahren ein Runden in Richtung Null, handelt es sich um negative, ein Runden in Richtung $-\infty$. Es handelt sich also stets um ein Abrunden. Der Betrag einer positiven Zahl kann hierbei nur kleiner werden, der Betrag einer negativen Zahl nur größer. Ist es erforderlich, daß sich der Betrag stets (unabhängig vom Vorzeichen) im gleichen Sinne (der Verkleinerung) ändert, so sind die Verfahren abzuwandeln:

1. Löschen niedrigstwertiger Bitpositionen

Bei positivem Vorzeichen Nullen einfügen, bei negativem Einsen.

Abschneiden niedrigstwertiger Bitpositionen

eine negative Zahl vor dem Verschieben stets in ihren Betrag wandeln (Zweierkomplementbildung*)) und nach dem Verschieben erneut das Zweierkomplement*) bilden.

*): Zweierkomplement = bitweise Negation + 1. Viele Prozessoren haben entsprechende Befehle.