

Kodierungsverfahren

Bei der Übertragung von binären Daten über eine physikalische Leitung ist eine Kodierung notwendig. Das Zuordnen von Zeichen zu Bedeutungen nennt man Kodierung. Die Dekodierung entnimmt den Zeichen die Bedeutung. Bei der Datenübertragung wird im Allgemeinen von binären Zeichen ausgegangen.

Digitale Signale müssen innerhalb der digitalen Netze den Eigenschaften des jeweils verwendeten Übertragungskanals angepasst werden, um optimale Übertragungseigenschaften zu gewährleisten. Die aus den Endgeräten herrührende Kodierung ist nicht in allen Fällen für eine Übertragung geeignet. Die Netzwerkkarte enthält eine Signalisierungskomponente, die die Daten neu kodiert bzw. dekodiert. Die Anzahl der möglichen Kodierungsverfahren ist sehr groß, da es kein universell für alle Übertragungsprobleme geeignetes Verfahren gibt.

In der Bitübertragungsschicht können durch die Art der Kodierung bereits Fehler bei der Datenübertragung vermieden werden. In den höheren Schichten des OSI-Modells werden die eigentlichen Verfahren zur Fehlererkennung und -behebung eingesetzt.

Die wesentlichen Anforderungen an die Kodierungsverfahren sind:

- eine hohe Übertragungsreichweite
- Gleichspannungsfreiheit ermöglichen
- eine Taktrückgewinnung sichern

Eine direkte binäre Codierung mit 0 Volt für ein 0-Bit und 5 Volt für ein 1-Bit wird selten genutzt, da dies zu Mehrdeutigkeiten führen kann. Sendet eine Station die Bitfolge 0001000, kann das von anderen Stationen fälschlicherweise als 1000000 oder 0100000 interpretiert werden, da sie nicht zwischen einem Sender im Leerlauf und einem 0-Bit unterscheiden können. Zudem besteht das Problem, dass der Empfänger das Signal evtl. mit einer etwas anderen Frequenz abtastet, als der Sender dieses erzeugt hat. Unterschiedliche Taktfrequenzen können insbesondere bei langen Folgen von Nullen oder Einsen (hoher Gleichspannungsanteil) dazu führen, dass Bitfehler aufgrund mangelnder Synchronisation auftreten (=Bitsynchronisation). In den Kodierungsverfahren sollten also häufige Übergänge eine Synchronisation ermöglichen. Durch einen geringen Gleichspannungsanteil wird zudem die zur Verfügung stehende Bandbreite besser ausgenutzt.

Aufgrund von Rauschen werden die Signale bei der Datenübertragung verzerrt. Bei geringen Differenzen zwischen den Spannungspegeln der einzelnen Signale kann es durch das Rauschen zu „Fehlinterpretationen“ durch den Empfänger kommen. Um eine hohe Übertragungsreichweite (Grad für die Stärke des Rauschens) zu gewährleisten, sollten die Kodierungsverfahren daher eindeutige Signalpegel definieren. Bei Kodierungen mit mehreren Spannungspegeln werden mehrere Bits pro Signal übertragen, d.h. die Bitrate ist doppelt oder dreifach so hoch wie die Baudrate. Aufgrund der höheren Empfindlichkeit gegenüber Rauschen sind solche Multilevel-Codes nicht für alle Anwendungen geeignet. Die Bitfehlerwahrscheinlichkeit steigt mit der Zahl der Signalzustände.

Im Folgenden werden einige Kodierungsverfahren vorgestellt:

Non-Return to Zero (NRZ) Code

Für einen NRZ-Code ist charakteristisch, dass das Spannungsniveau während eines Bitintervalls konstant bleibt. Eine 1 wird mit einem High-Signal und eine 0 mit einem Low-Signal definiert. Das Signal besitzt dadurch einen geringen Taktgehalt bei langen, unveränderten Werten und weist dadurch einen hohen Gleichspannungsanteil auf. Als Vorteile sind die einfache Implementierung und die hohe Effizienz zu nennen. NRZ wird beispielsweise im SONET eingesetzt.

Exkurs: Scrambling

Das Scrambling (Verwürfeln) ist eigentlich kein Kodierungsverfahren, sei hier aber als eine Methode vorgestellt, den Nachteil eines hohen Gleichspannungsanteils bzw. eines geringen Taktgehalts von NRZ auszugleichen. Das Scrambling hat allgemein die Funktion das Signal so umzuformen, dass die zur Verfügung stehende Bandbreite möglichst günstig ausgenutzt wird. Ziel des Scramblers ist es, dass keine identischen Signale aufeinander folgen und keine periodischen Signale entstehen (mit Ausnahme der Periodizität durch das Scrambling selbst). Technisch werden die zu übertragenden Daten mit einer festgelegten Signalfolge über ein XOR-Gatter zusammengefasst. Diese festgelegte Signalfolge unterscheidet sich bei den unterschiedlichen Scrambling-Arten und muss sowohl dem Sender als auch dem Empfänger bekannt sein.

Non-Return to Zero Inverted (NRZI) Code

Bei dem NRZI-Code stellt der Sender im Gegensatz zum NRZ-Code eine 1 mit einem Wechsel des Pegels dar. Bei einer 0 verharrt der Signalpegel auf seinem bisherigen Niveau. Dadurch wird das Problem aufeinander folgender Einsen zwar gelöst, jedoch nicht für aufeinander folgende Nullen. NRZI wird beispielsweise bei vielen seriellen Schnittstellen eingesetzt.

4B/5B- Code

Bei diesem häufig in Kombination mit NRZI eingesetzten Kodierungsverfahren werden in den Bitstrom zusätzliche Bits eingefügt, um lange Folgen von Nullen und Einsen zu vermeiden. Konkret werden in diesem Verfahren 4 Bits der eigentlichen Daten in einem 5-bit-Codewort kodiert, der dann an den Empfänger übertragen wird. Die 5-bit-Codewörter werden so gewählt, dass jedes nicht mehr als eine führende und höchstens zwei nachgestellte Nullen enthält. Dadurch werden maximal drei aufeinander folgende Nullen übertragen. 4B/5B löst also das Problem der aufeinander folgenden Nullen, während NRZI das Problem der aufeinander folgenden Einsen löst. Die Effizienz reduziert sich bei diesem Verfahren um 20%. Der 4B/5B-Code wird beispielsweise bei FDDI eingesetzt.

Manchester-Code

Jeder binäre Wert wird beim Manchester-Code durch einen Phasensprung kodiert. Ein binäres 1-Bit wird gesendet, indem die Spannung im ersten Intervall hoch und im zweiten niedrig gesetzt wird (negative Flanke). Eine binäre Null ist genau das Gegenteil: eine positive Flanke. Durch die erzwungenen Flanken in der Mitte jeder Bitperiode wird die Bitsynchronisation gewährleistet. Elektronisch wird hier der Takt mit den zu übertragenden Daten mit einem XOR-Gatter zusammengefasst. Der Manchester-Code hat daher einen hohen Taktgehalt und ist gleichstromfrei. Er wird im Ethernet eingesetzt und ist daher sehr verbreitet. Die Kodierung hat allerdings den Nachteil, dass sie bei einer Polaritätsvertauschung einen Fehler erzeugt. Bei dem Manchester-Code ist die Bitrate kleiner als die Baudrate, bzw. benötigt die doppelte Bandbreite.

Differential Manchester-Code

Der Nachteil der Manchester-Codierung bei Polaritätsvertauschung einen Fehler zu verursachen, wird hier vermieden. Hier wird ein 1-Bit dadurch übertragen, dass zu Beginn des Intervalls kein Übergang auftritt. Ein 0-Bit wird durch einen Übergang angezeigt. In beiden Fällen gibt es in der Mitte des Intervalls einen Übergang. Diese Kodierung setzt komplexere Geräte voraus, bietet aber eine höhere Rauschfestigkeit.

CMI-Code

Der CMI-Code (Code Mark Inversion) kennzeichnet den binären Wert 0 mit einem Sprung in der Bitmitte. Die binäre 1 wird mit einem alternierenden Amplitudenwert in Abhängig-

keit der Kodierung des letzten 1-Wertes codiert. Dieses Kodierungsverfahren ist gleichstromfrei und mit hohem Taktgehalt versehen.

AMI-Code

Der AMI-Code gibt bei einer 0 ein Null-Level aus, bei einer 1 abwechselnd ein positives oder negatives High. Beim AMI-Code bestehen daher drei Signalzustände, trotzdem ist die Baudrate gleich der Bitrate. Ähnlich wie bei NRZ besteht hier das Problem einer geringen Gleichspannungsfreiheit und einer geringen Taktrückgewinnung bei langen Folgen von Nullen oder Einsen. Dieses Verfahren wird beispielsweise im S_0 -Bus bei ISDN eingesetzt.

HDB3-Code

Dieses Kodierungsverfahren basiert auf dem AMI-Code, löst jedoch das Problem der geringen Gleichspannungsfreiheit bei langen Nullfolgen. Dies geschieht dadurch, dass der Sender maximal vier aufeinander folgende Nullen sendet, eine evtl. vorhandene fünfte Null wird als „falsche“ gesendet. Falsch bedeutet in diesem Fall, dass der Pegel der letzten regulären Eins verwendet wird.

In der folgenden Abbildung werden die verschiedenen Kodierungsverfahren grafisch gegenüber gestellt (vgl. Roppel (2006) „Grundlagen der digitalen Kommunikationstechnik“):

