

### Der Programmierspannungs-Umschalter

Der Programmierspannungs-Umschalter hat die Aufgabe, die Spannung am EPROM-Anschluß  $V_{pp}$  während des Programmiervorgangs auf 24 V und während aller anderen Betriebszustände auf 5 V zu schalten. Er wird hierzu (durch das Betriebsprogramm) über die Portleitung PC7 gesteuert: Bild 47.

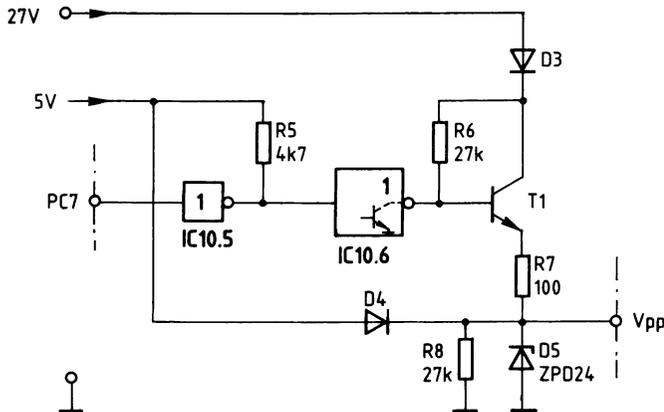


Bild 47: Der Programmierspannungs-Umschalter.

Die Funktion beim Programmieren:

Das Betriebsprogramm gibt an PC7 einen H-Pegel aus. Hierdurch sperrt der Ausgangstransistor von IC10.6. Die 27-V-Spannungsquelle treibt über die Verpolungsschutzdiode D3 und über R6 einen Basisstrom in den Transistor T1. Der Transistor leitet und gibt an seinem Emittter eine Spannung von ca. 25,2 V ab. Der Widerstand R7 begrenzt den Strom für die Z-Diode D5, die an den EPROM-Anschluß  $V_{pp}$  eine Spannung von 24 V abgibt. Die Diode D4 verhindert beim Programmieren einen Stromfluß vom Emittter zur 5-V-Spannungsquelle.

Die Funktion in allen anderen Betriebszuständen:

Das Betriebsprogramm gibt an PC7 einen L-Pegel aus. Der Ausgangstransistor von IC10.6 leitet und verbindet die Basis des Transistors T1 mit 0 V, wodurch T1 sperrt. Über die Diode D4 fließt nun ein Strom in den Arbeitswiderstand R8 und bewirkt am Anschluß  $V_{pp}$  eine Spannung von ca. 4,5 V. Diese Spannung reicht nicht aus, um die Z-Diode in den leitenden Zustand zu steuern.

## Einleitung

Jeder Mikrocomputer enthält neben Prozessor und Speicher auch Eingabe- und Ausgabe-Einheiten, über die der Datenverkehr mit Geräten außerhalb des Mikrocomputers stattfindet. Häufig bezeichnet man Eingabe- und Ausgabe-Einheiten auch als Eingabe- bzw. Ausgabe-Ports oder als Eingabe- bzw. Ausgabe-Schnittstellen, und die Geräte außerhalb des Mikrocomputers nennt man Peripheriegeräte.

Die zu übertragenden Daten können entweder seriell oder parallel übermittelt werden. Bei der seriellen Übertragung werden die einzelnen Bits eines Datenbytes nacheinander über eine einzige Leitung gesendet oder empfangen, während sie bei der parallelen Datenübertragung gleichzeitig über entsprechend viele Leitungen geführt werden.

Innerhalb einer bestimmten Zeit kann mit der parallelen Datenübertragung eine wesentlich größere Datenmenge übermittelt werden als mit der seriellen. Der Aufwand ist jedoch durch die hohe Leitungszahl recht groß. Daher wird die Parallelübertragung vor allem dann eingesetzt, wenn große Datenmengen in kurzer Zeit über geringe Entfernungen übertragen werden müssen. Beispiele hierzu sind der Anschluß von Schnelldruckern oder Floppy-Disk-Datenspeichern an Computersysteme.

Bei großen Entfernungen zwischen Peripheriegerät und Mikrocomputer bevorzugt man die serielle Datenübertragung. Dies ist z. B. im Fernschreibverkehr, beim Bildschirmtext (BTX) und beim Teletext der Fall. Häufig wird die Telefonleitung zur Datenfernübertragung (DFÜ) verwendet. Bestimmte Peripheriegeräte werden allerdings auch bei geringen Entfernungen seriell angesteuert, wie z. B. Kassettenrecorder zur magnetischen Datenaufzeichnung.

In diesem Abschnitt wird eine serielle Ein/Ausgabe-Baugruppe zum MFA-Mikrocomputer (siehe Vorwort) beschrieben, die einen programmierbaren Schnittstellenbaustein mit seriellm Datensender und seriellm Datenempfänger besitzt. Durch die Programmierbarkeit des Schnittstellenbausteins ist die Verarbeitung aller gebräuchlichen Datenübertragungsformate möglich; daher ist die programmierbare Serienschnittstelle universell verwendbar.

Die Erklärung der programmierbaren Serienschnittstelle beschränkt sich im Rahmen dieses Buches auf die asynchrone Betriebsweise. Darüber hinausgehende Informationen müssen den Datenbüchern der Baueinsteher entnommen werden.

---

## Die Grundlagen der seriellen Datenübertragung

Zur seriellen Datenübertragung ist ein serieller Datensender, eine Übertragungsleitung und ein serieller Datenempfänger erforderlich. Da die Daten eines Mikrocomputers auf dem Datenbus in paralleler Form vorliegen, erfolgt im Datensender eine parallel/seriell-Umwandlung. Der Empfänger besitzt eine seriell/parallel-Umwandlung; so entsteht wieder die parallele Datenform. Die Wandlung paralleler Daten in serielle und umgekehrt erfolgt z. B. mit Hilfe von Schieberegistern. Bild 48 zeigt das Prinzip der seriellen Datenübertragung, wobei zur vereinfachten Darstellung von einem 5-Bit-Datenbus ausgegangen wird.

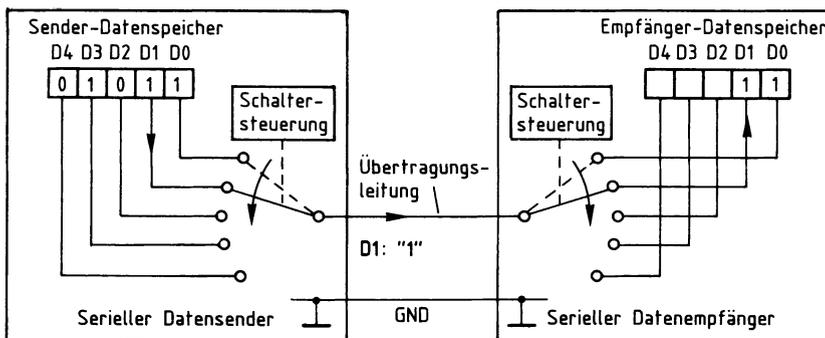


Bild 48: Das Prinzip der seriellen Datenübertragung.

Im seriellen Datensender befindet sich ein Sender-Datenspeicher, der das auszugebende Datenwort 01011 enthält (Bild 48). Es stammt z. B. aus dem Akku eines Prozessors und ist über den Datenbus in den Sender-Datenspeicher gelangt.

Der fünfpolige Umschalter des Datensenders schaltet nacheinander jedes Datenbit für eine gewisse Zeit auf die Übertragungsleitung durch und wechselt dann zum nächsten Bit. Die Übertragung beginnt stets beim niederwertigsten Bit D0. Im Empfänger befindet sich ebenfalls ein Umschalter, über den die ankommenden Datenbits einzeln in den Empfänger-Datenspeicher gelangen. In der gezeichneten Schalterstellung wird gerade das Bit D1 übertragen. Die Umschalter im Sender und Empfänger werden jeweils durch eine Schaltersteuerung betätigt.

Für eine korrekte Datenübertragung müssen folgende Bedingungen erfüllt sein:

Die Schalter müssen gleichzeitig (synchron) umgeschaltet werden. Dies wird durch die gleiche Taktfrequenz der Schaltersteuerungen im Sender und im Empfänger erreicht. Durch die Taktfrequenz wird die sogen. Baudrate bestimmt. Sie gibt die Anzahl der pro Sekunde übertragenen Bits an.

Der Empfänger muß den Beginn der Datenübertragung erkennen können, damit er seinen Schalter in die Anfangsposition stellen kann. Hierzu dient ein zusätzliches Bit, das Startbit, das vor den Datenbits vom Sender ausgegeben wird.

Auf der Sender- und der Empfängerseite muß die gleiche Datenwortbreite (z. B. 5 Bits) vorliegen. Hierdurch kann der Empfänger das Ende einer Datenübertragung feststellen. Zur Trennung zweier Datensendungen wartet der Sender nach der Ausgabe der Datenbits eine gewisse Zeit, bevor er eine neue Übertragung beginnt. Diese Zeit wird durch Stopbits realisiert.

### Die Baudrate

Die oben erwähnte Taktfrequenz bestimmt, wieviel Bits pro Sekunde übertragen werden. Die Einheit der Übertragungsgeschwindigkeit ist 1 Bd (Baud, nach dem französischen Ingenieur Baudot) und bedeutet 1 Bit pro Sekunde. Bei 300 Bd werden z. B. in einer Sekunde 300 Bits übertragen, wobei ein Bit eine zeitliche Dauer von 3,33 ms besitzt. Bild 49 zeigt einige in der Datenübertragungs-Technik häufige Baudraten, die zugehörige zeitliche Länge eines Bits und typische Einsatzgebiete.

Obertragungsgeschwindigkeit (Baudrate) in Bd	Länge eines Bits (Dauer der Übertragung eines Bits) in ms	Anwendungsbeispiele
50	20	TTY (Teletype, Fernschreiber), RTTY (Radio-Teletype, Funkfernschreiber), teilweise BTX (Bildschirmtext) Datenübertragung über Telefonleitungen, Fernkopierer, BTX, Datenspeicherung auf Tonbandkassetten, TELETEXT; Anschluß von Terminals an Computer, innerbetriebliche Datenübertragung, schnelle Drucker und Datenstationen
75	13,3	
110	9,1	
300	3,3	
600	1,67	
1200	0,83	
2400	0,42	
4800	0,21	
9600	0,1	
19200	0,05	

Bild 49: Gebräuchliche Baudraten.

### Das Startbit und die Stopbits

Damit der Datenempfänger den Anfang einer Datenübertragung erkennt, fügt der Datensender ein zusätzliches Bit, das Startbit, in den Datenstrom ein. Es wird vor dem ersten Datenbit ausgesendet und hat stets L-Pegel. Außerdem ist festgelegt, daß die Übertragungsleitung im Ruhezustand, also vor dem Aussenden des Startbits, H-Pegel besitzt.

Nach der Übertragung des Startbits und der Datenbits folgt stets eine Pause, die durch sogen. Stopbits gebildet wird, bevor das nächste Datenwort übertragen wird. Diese Pause hat mindestens eine Dauer von einem Bit, wird aber meistens länger gewählt (1,5 Bits, 2 Bits oder mehr). Die Pause zwischen zwei Datenworten ist zur Trennung der Datensendungen erforderlich. Außerdem kann der Sender in dieser Zeit neue Daten entgegennehmen und der Empfänger die empfangenen Daten weiterverarbeiten (z. B. abspeichern). Startbit und Stopbits werden im Empfänger dabei aus dem Datenstrom entfernt, so daß nur die eigentlichen Datenbits erhalten bleiben.

Bild 50 zeigt das Impulsdiagramm bei der seriellen Datenübertragung von fünf Datenbits mit Start- und zwei Stopbits. Hierbei wird (wie im Beispiel von Bild 48) der Datenwert 01011 übertragen.

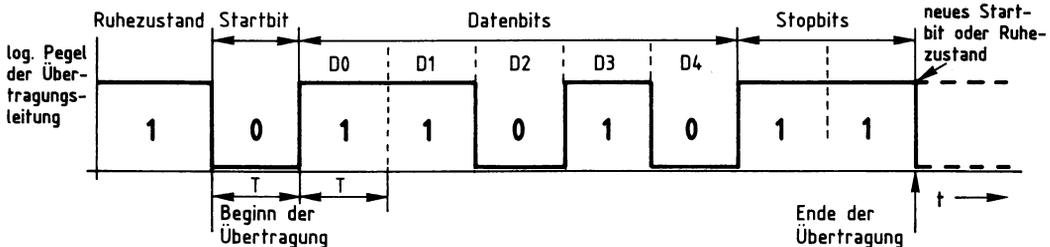


Bild 50: Die serielle Übertragung mit Start- und Stopbits.

Vor dem Beginn der Übertragung, im Ruhezustand, führt die Übertragungsleitung 1-Signal (Bild 50). Der erste 1-0-Wechsel wird vom Empfänger als Beginn der Datenübertragung (Beginn des Startbits) aufgefaßt. Alle Bits besitzen die gleiche Dauer  $T$ , durch welche die Übertragungsgeschwindigkeit (Baudrate) festgelegt ist.

Im Empfänger, der dieselbe Baudrate wie der Sender besitzt, wird also davon ausgegangen, daß die fünf Datenbits nach dem Startbit jeweils im Abstand  $T$  auftreten. Aus Sicherheitsgründen wartet der Empfänger aber nicht nur die Zeit  $T$  ab, bis er das erste Datenbit aufnimmt, sondern noch etwas länger.

Die Abfrage der Übertragungsleitung zur Ermittlung des ersten Datenbits erfolgt  $1,5 * T$  nach der 1-0-Flanke des Startbits und somit in der Bit-Mitte von D0. Hierdurch wird nicht der Bitwechsel, sondern mit Sicherheit stets der stabile Zustand der Übertragungsleitung erfaßt. Alle weiteren Bits werden im Abstand  $T$  betrachtet:

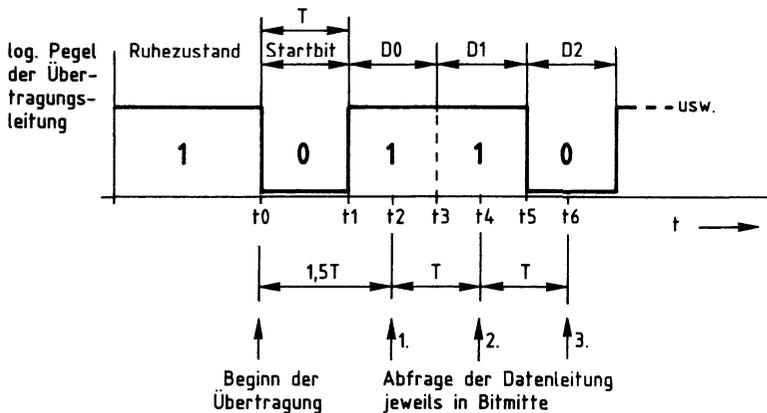


Bild 51: Die Abfrage der Datenleitung in der Bit-Mitte.

Zum Zeitpunkt  $t_0$  tritt das Startbit auf und zum Zeitpunkt  $t_1$  das erste Datenbit.  $1,5$  Perioden nach dem Beginn des Startbits ist zum Zeitpunkt  $t_2$  die Mitte des ersten Datenbits erreicht; nun liest der Empfänger den Bitwert ein. Nach jeweils einer weiteren Taktzeit  $T$  wird immer zur Bit-Mitte jedes folgende Datenbit erfaßt (D1 bei  $t_4$ , D2 bei  $t_6$  usw.). Durch diese Vorgehensweise werden – wie gesagt – die Zeitpunkte, an denen sich die Daten ändern können ( $t_1$ ,  $t_3$ ,  $t_5$  usw.), mit Sicherheit nicht erfaßt, sondern erst die stabilen Signalzustände der Übertragungsleitung.

### Das Paritätsbit

Vielfach soll auf der Empfängerseite kontrolliert werden, ob bei der seriellen Datenübertragung Fehler auftreten. Dies ist mit Hilfe des Paritätsbits (Parität = Gleichheit) möglich. Das Paritätsbit wird (wenn gewünscht) zusätzlich als Prüfbit nach den Datenbits, aber vor den Stopbits ausgesendet. Es besitzt in Abhängigkeit vom Datenwert entweder 1- oder 0-Signal, wobei der Wert des Paritätsbits vom Datensender selbstständig nach bestimmten Kriterien (s. u.) berechnet und automatisch an das letzte Datenbit angefügt wird.

Man unterscheidet zwei Arten, das Paritätsbit zu bilden: Bei der geraden Parität (engl. parity even) nimmt das Paritätsbit den Wert an, der erforderlich ist, um zusammen mit den Datenbits eine gerade Anzahl von 1-Signalen (0, 2, 4 usw.) zu bilden. Wählt man die ungerade Parität (engl. parity odd), so nimmt das Paritätsbit denjenigen Wert an, der zum Erzielen einer ungeraden Anzahl von 1-Signalen (1, 3, 5 usw.) erforderlich ist. Bild 52 zeigt ein Beispiel zur geraden Parität.

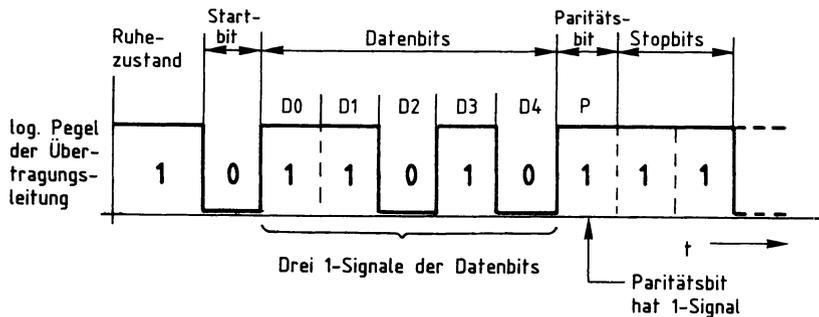


Bild 52: Die gerade Parität (Beispiel 1).

In Bild 52 wird der Datenwert 01011, der dreimal 1-Signal besitzt, mit gerader Parität übertragen. Das Paritätsbit nimmt in diesem Beispiel 1-Signal an, damit insgesamt vier 1-Signale (eine gerade Zahl) auftreten.

Im Datenempfänger wird aus den fünf Datenbits selbständig ein Paritätsbit berechnet und mit dem empfangenen Paritätsbit verglichen. Sind beide gleich, kann mit hoher Wahrscheinlichkeit von einer korrekten Datenübertragung ausgegangen werden. Mit Hilfe des Paritätsbits lassen sich allerdings nicht alle Übertragungsfehler (z. B. Doppelbitfehler) erkennen.

Bild 53 zeigt als weiteres Beispiel zur geraden Parität die Übertragung des Datenwertes 01010. Da hier zwei 1-Signale (eine gerade Zahl) vorkommen, tritt das Paritätsbit diesmal mit 0-Signal auf, damit die gerade Gesamtzahl der 1-Signale erhalten bleibt.

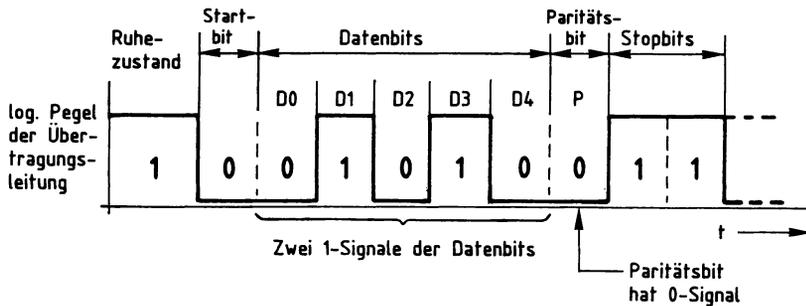


Bild 53: Die gerade Parität (Beispiel 2).

Für die ungerade Parität ist kein Beispiel gezeigt. Hierbei gilt sinngemäß, daß durch das Paritätsbit insgesamt eine ungerade Bitzahl erzeugt wird. Es ist gleich, ob bei der Überwachung der seriellen Datenübertragung die gerade oder die ungerade Parität gewählt wird. Beachtet werden muß lediglich, daß die Empfängerseite diejenige Parität zu überprüfen hat, die in der Senderseite gebildet wird.

### Das Übertragungsformat und die Überwachung von Stopbits

Sender und Empfänger einer seriellen Datenübertragungsstrecke müssen stets das gleiche Datenübertragungsformat besitzen. Hierunter versteht man die Baudrate, die Anzahl der Datenbits und, wenn sie vorgenommen wird, die Paritätsprüfung. Auch die Anzahl der Stopbits gehört dazu; sie sollte gleich sein, jedoch darf der Sender eine größere Anzahl von Stopbits aussenden, als im Empfänger erwartet wird. Zusätzliche Stopbits des Senders werden im Empfänger nicht bemerkt, sondern als Ruhezustand der Übertragungsleitung aufgefaßt. Nicht vorkommen darf, daß der Empfänger z. B. zwei Stopbits erwartet, der Sender aber nur ein Stopbit ausgibt.

Zur zusätzlichen Fehlerkontrolle überprüfen viele serielle Datenempfänger neben dem Paritätsbit auch das Datenübertragungsformat, das auch als Übertragungsrahmen bezeichnet wird. Durch das auf Sender- und Empfängerseite einheitliche Übertragungsformat kann der Empfänger das Auftreten des ersten Stopbits vorhersehen und dieses Bit überprüfen. Es muß zu einem bestimmten Zeitpunkt stets mit 1-Signal auftreten. Ist dies nicht der Fall, liegt ein Fehler im Datenübertragungsformat bzw. im Übertragungsrahmen vor. Ein derartiger Fehler wird oft als frame error (frame = engl. Rahmen), also als Rahmenfehler bezeichnet.

## Synchrone und asynchrone Datenübertragung

Bei der seriellen Datenübertragung wird meist das beschriebene Start/Stop-Verfahren angewendet. Man bezeichnet es im allgemeinen als asynchrones Datenübertragungsverfahren, weil die Synchronisation (der Gleichlauf) zwischen dem Sender und dem Empfänger nur durch das Startbit erfolgt und der Empfängertakt nicht beeinflusst wird.

Auch bei längeren Übertragungen oder nach größeren Übertragungspausen wird beim Asynchronverfahren durch das Startbit eine einwandfreie Synchronisation erzielt. Dieser Gleichlauf braucht immer nur für kurze Zeit zu bestehen, da bei jedem übertragenen Datenwort neu synchronisiert wird. Durch die empfängerseitige Datenabfrage in der Bitmitte bleiben sogar geringe zeitliche Verschiebungen ohne Auswirkungen. Dies erlaubt die Verwendung einfacher Taktgeneratoren im Sender und im Empfänger, die in ihren Taktfrequenzen sogar etwas voneinander abweichen dürfen (1 bis 2 %).

Ein gewisser Nachteil der asynchronen Datenübertragung besteht im Zeitbedarf für die Übertragung der zusätzlichen Bits (Startbit, evtl. Prüfbit und ein bis zwei Stopbits). Bei einem 5-Bit-Datenwort werden so beispielsweise insgesamt 9 Bits übertragen. Dies führt zu einem deutlich geringeren Datendurchsatz.

Dieser Nachteil besteht bei der synchronen Datenübertragung nicht; bei ihr werden die Datenbits direkt nacheinander (ohne Start- und Stopbits) übertragen. Hierzu ist aber während der ganzen Datenübertragung eine exakte Synchronisation zwischen dem Sender und dem Empfänger erforderlich. Sie ist auf verschiedene Weisen erzielbar:

In einigen Fällen wird zusätzlich zu den Daten über eine besondere Leitung der Sendertakt zum Empfänger übertragen, wodurch beide Geräte den gleichen Takt erhalten und synchron arbeiten.

Ist dies nicht möglich, so gibt der Sender zu Beginn der Datenübertragung (und eventuell auch danach in gewissen Zeitabständen) bestimmte Synchronisationsworte aus, die der Empfänger erkennt und mit denen er synchronisiert wird. Tritt in der Datenübertragung eine Pause auf, dann werden vom Sender zum Überbrücken der Pause weiterhin bestimmte Signale ausgegeben, damit die Synchronisation erhalten bleibt.

Die synchrone Datenübertragung arbeitet schneller als die asynchrone, erfordert aber einen höheren Aufwand und ist insgesamt gesehen etwas schwieriger zu handhaben. Da die Synchron-Übertragung in der Mikrocomputertechnik nur wenig verbreitet ist und auch im hier zugrundegelegten Mikrocomputer nicht angewendet wird, bleibt sie außer Betracht.

### Blockschaltbild und Arbeitsweise der Seriell-Baugruppe

Ihre Funktion wird im wesentlichen durch den programmierbaren Schnittstellenbaustein 8251 bestimmt. Die Bezeichnung USART ist eine Abkürzung des Begriffs Universal Synchronous/Asynchronous Receiver/Transmitter. Dies bedeutet Universal synchroner/asynchroner Empfänger (engl. receiver) und Sender (engl. transmitter). Darüber hinaus enthält die Baugruppe einen Adreßvergleicher und einen Baudraten-Teiler. Zunächst wird die grundsätzliche Wirkungsweise der Baugruppe anhand des Blockschaltbildes erklärt: Bild 54.

Die Datenanschlüsse des 8251 sind direkt mit den Datenleitungen D0 bis D7 des Systembus verbunden. Sie führen über den Datenbus-Puffer zum internen Datenbus des Bausteins, an den insgesamt fünf Register mit folgenden Funktionen angeschlossen sind:

Das Betriebsarten- und das Kommando-Register bestimmen gemeinsam die Funktion des seriellen Schnittstellenbausteins. Durch Programmierung (d.h. durch das Einschreiben bestimmter Steuerworte in diese Register) wird z. B. die synchrone oder die asynchrone Betriebsart und die Datenwortlänge der seriellen Datenübertragung (5, 6, 7 oder 8 Bits) festgelegt.

Das Sender-Register gehört zum seriellen Datensender. Es <sup>er</sup>enthält die seriell auszugebenden Daten in paralleler Form über den Datenbus-Puffer und den internen Datenbus. Innerhalb des Datensenders werden das Startbit und die Stopbits in den Datenstrom eingefügt, der am Anschluß TxD (Transmitter Data) in serieller Form ausgegeben wird. Außerdem erfolgt im Datensender die parallel/seriell-Wandlung, wozu die Sendersteuerung vom Baudratenteiler den Sendertakt  $\overline{\text{TxC}}$  (Transmitter Clock) erhält.

Das Empfänger-Register kann über den Anschluß RxD (Receiver Data) einen seriellen Datenstrom empfangen. Innerhalb des Datenempfängers findet die seriell/parallel-Wandlung der empfangenen Daten statt, wobei alle zusätzlichen Bits (Start-, Paritäts- und Stop-Bits) aus dem empfangenen Signal entfernt werden. Vom Baudratenteiler erhält die Empfängersteuerung den Empfängertakt  $\overline{\text{RxC}}$  (Receiver Clock), der in der programmierbaren Serienschnittstelle gleich dem Sendertakt ist. Da Sender- und Empfänger-Steuerung des 8251 getrennte Takteingänge besitzen, könnte beim Sender und beim Empfänger mit unterschiedlichen Baudraten gearbeitet werden, falls dies erforderlich ist.

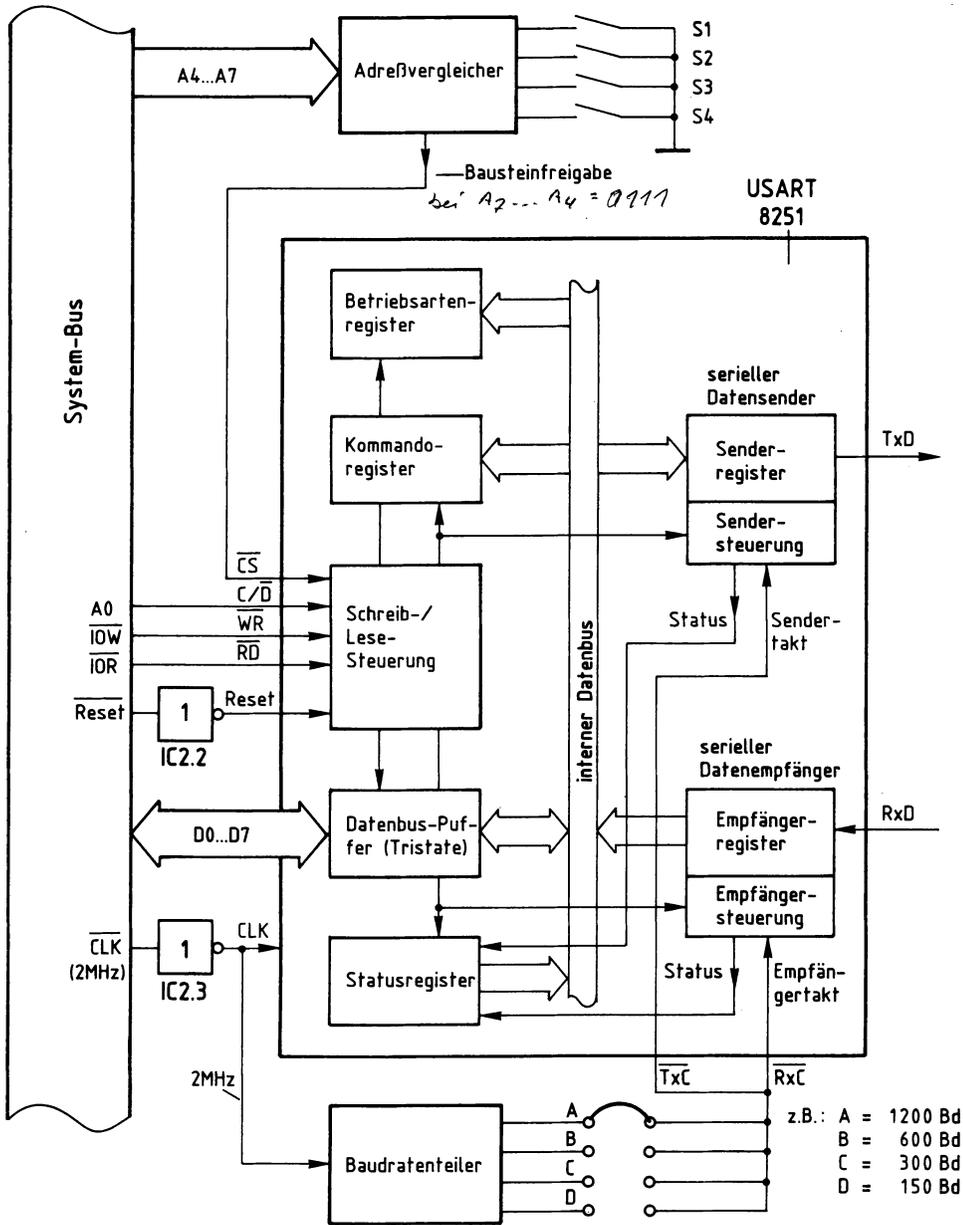
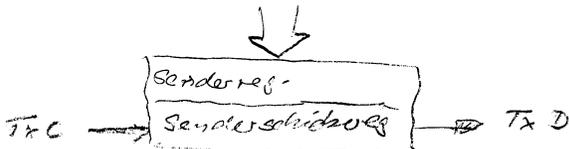


Bild 54: Das Blockschaubild der Seriell-Baugruppe; die in den Bildern 74 und 75 (Seiten 446 ff.) vorge-stellte Variante erlaubt bis zu 9600 Bd.



Das Status-Register (Status = Zustand) erhält sowohl vom seriellen Datensender als auch vom seriellen Datenempfänger Informationen über den Zustand dieser Einheiten. Beispiele hierfür sind das Auftreten von Übertragungsfehlern (parity error oder frame error) oder Meldungen wie »Senderregister leer« und »Empfängerregister hat Zeichen empfangen«. Der Prozessor kann das Statusregister lesen und erkennt hierdurch u. a., ob er ein neues Zeichen zur seriellen Ausgabe an die Baugruppe ausgeben kann oder ob ein seriell empfangenes Zeichen vorliegt und eingelesen werden muß.

Damit der serielle Schnittstellenbaustein Daten vom Prozessor erhalten oder an ihn abgeben kann, muß der Anschluß  $\overline{CS}$  (Chip-Select) über die Leitung Bausteinfreigabe ein L-Signal erhalten (Bild 54). Es stammt vom Adreßvergleichler und wird nur abgegeben, wenn die Baugruppe vom Prozessor angesprochen wird, d. h. wenn die Bitkombination der Adreßleitungen A4 bis A7 gleich ist mit der Bitkombination, die mit den Schaltern S1 bis S4 eingestellt wurde (Baugruppennummer).

Die Auswahl der einzelnen Register beim Datenverkehr mit der CPU erfolgt mit Hilfe der Bausteinanschlüsse  $C/D$ ,  $\overline{WR}$  und  $\overline{RD}$ , auf die später noch weiter eingegangen wird.

Zur zeitlichen Steuerung der internen Vorgänge besitzt der Baustein 8251 die drei Taktanschlüsse  $\overline{RxC}$ ,  $\overline{TxC}$  und CLK (Bild 54).  $\overline{RxC}$  und  $\overline{TxC}$  bestimmen die sender- bzw. empfängerseitige Datenübertragungsrate (Baudrate). Hierzu wird der 2-MHz-Systemtakt einem Baudratenteiler zugeführt, der z. B. Baudraten von 1200 Bd, 600 Bd, 300 Bd und 150 Bd ermöglicht.

An CLK ist ein Taktsignal erforderlich, dessen Frequenz wesentlich höher als die höchste Ausgangsfrequenz des Baudratenteilers ist. Hierzu eignet sich der 2-MHz-Systemtakt, der durch IC2.3 invertiert auf den CLK-Anschluß geführt wird. Diese Invertierung soll das  $\overline{CLK}$ -Signal lediglich auffrischen. IC2.2 invertiert das L-aktive  $\overline{Reset}$ -Signal des Systembus, da der Schnittstellenbaustein 8251 einen H-aktiven Reset-Eingang besitzt. Nach jedem Reset muß der Baustein neu programmiert werden.

### Steuer-Register und Daten-Register

Vergleicht man die Register des Bausteins 8251 miteinander, so kann zwischen Steuerregistern (engl. Control Register) und Datenregistern unterschieden werden. In einige der Register kann der Prozessor Daten einschreiben, andere dagegen nur lesen. Bild 55 zeigt die Aufteilung der Register in diese beiden Gruppen sowie die erforderlichen Signale zu ihrer Ansteuerung.

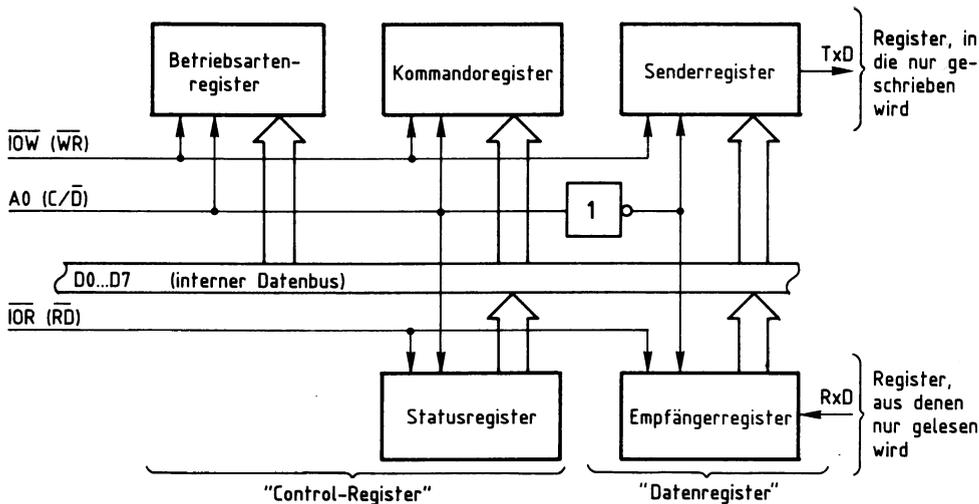


Bild 55: Die Aufteilung der Register des 8251 in Gruppen.

Zur Unterscheidung der Steuer- und der Daten-Register voneinander dient der Bausteinanschluß  $C/\overline{D}$  (Control/Data): Führt  $C/\overline{D}$  H-Pegel, so sind die Steuer-Register, dagegen bei L-Pegel die Datenregister ausgewählt. Aus Bild 55 geht ebenfalls hervor, welche Register vom Prozessor gelesen und welche mit Daten beschrieben werden können: Das Status- und das Empfänger-Register können nur gelesen, und die Betriebsarten-, Kommando- und Sender-Register können nur beschrieben werden.

Mit Hilfe der L-aktiven Steuerleitung  $\overline{IOW}$ , die mit dem Bausteinanschluß  $\overline{WR}$  verbunden ist, schreibt der Prozessor Datenwerte in den Baustein ein.

Zur Unterscheidung, ob bei einem Schreibvorgang das Senderregister (mit den seriell auszugebenden Daten) oder das Betriebsarten- bzw. das Kommando-Register (mit Steuerworten) beschrieben wird, dient der Bausteinanschluß  $C/\overline{D}$ , der mit der Adreßleitung  $A0$  verbunden ist.

Bei L-Signal an  $C/\overline{D}$  (entspricht Data) gelangen die vom Prozessor ausgegebenen Datenworte in das Senderregister. Liegt bei einem Schreibvorgang am Anschluß  $C/\overline{D}$  jedoch H-Pegel vor (entspricht Control), wird entweder das Betriebsarten- oder das Kommandoregister mit einem Steuerwort beschrieben. Diesen Vorgang nennt man Initialisieren des Schnittstellenbausteins; hierdurch wird die Betriebsart programmiert.

Bedingt durch die Vielzahl der Programmiermöglichkeiten besitzt der 8251 nicht nur ein, sondern zwei die Arbeitsweise bestimmende Register. Die Unterscheidung, ob das ausgegebene Steuerwort in das Betriebsartenregister oder in das Kommandoregister gelangt, erfolgt nicht durch die Schaltung, sondern durch folgende Festlegung: Nach einem Kaltstart des Mikrocomputers (Einschalten oder Reset) wird das erste Initialisierungs-Byte stets in das Betriebsarten-Register geschrieben. Das zweite und alle weiteren Steuerbytes gelangen in das Kommandoregister. Beide Register sind unter derselben Adresse ansprechbar; vgl. Seite 429ff.

Mit Hilfe der L-aktiven Steuerleitung  $\overline{IOR}$ , die mit dem Bausteinanschluß  $\overline{RD}$  verbunden ist, löst der Prozessor das Lesen von Daten aus dem Schnittstellenbaustein aus (Bild 55). Der Signalzustand des Anschlusses  $C/\overline{D}$  bestimmt dabei wieder, welches Register gelesen wird: Bei H-Pegel an  $C/\overline{D}$  (entspricht Control) gelangt der Inhalt des Statusregisters auf den Datenbus, bei L-Pegel (entspricht Data) der Inhalt des Empfängerregisters.

### Der Adreßvergleichler und die Baugruppennummer

Der Mikroprozessor steuert die programmierbare Serienschnittstelle wie jede andere Eingabe- und Ausgabe-Einheit an. Da immer nur eine einzige Einheit aktiviert sein darf, müssen alle im System vorhandenen Eingabe- und Ausgabe-Einheiten unterschiedliche Baugruppennummern besitzen. Aus diesem Grund ist die Baugruppennummer der programmierbaren Serienschnittstelle mit Hilfe von Schaltern einstellbar. Ein Adreßvergleichler übernimmt die Aufgabe, die Baugruppe nur dann zu aktivieren, wenn der Prozessor diejenige Adresse aussendet, die der eingestellten Baugruppennummer entspricht. Bild 56 zeigt die Schaltung des Adreßvergleichlers.

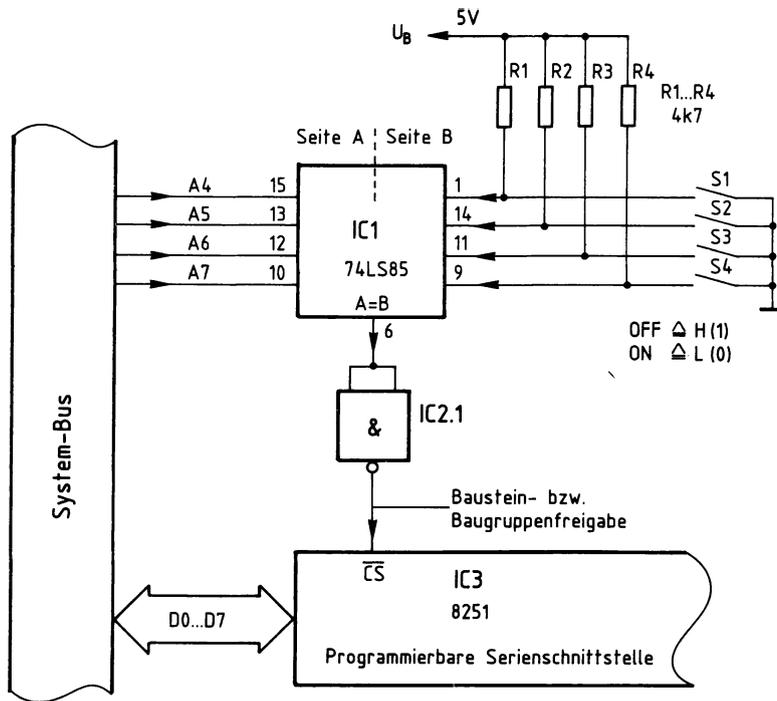


Bild 56: Der Adreßvergleichler und die Bausteinfreigabe.

Nur wenn die Bitkombination auf den Adreßleitungen A4 bis A7 der mit den Schaltern S1 bis S4 eingestellten Bitkombination entspricht, gibt der 4-Bit-Vergleichler IC1 an seinem Ausgang »A=B« ein H-Signal ab (Bild 56). Da der serielle Schnittstellenbaustein 8251 zur Aktivierung jedoch ein L-Signal benötigt, wird das Ausgangssignal von IC1 durch IC2.1 invertiert. Die Baugruppe kann nun Daten empfangen oder an den Prozessor abgeben.

Der Prozessor gibt stets eine vollständige 8-Bit-Adresse an A0 bis A7 aus, wenn er eine Eingabe- oder Ausgabe-Einheit anspricht. Beim hier vorgenommenen Adreßvergleich werden jedoch die Adreßleitungen A0 bis A3 nicht berücksichtigt. Daher steht nicht der volle Adreßbereich von 00H bis FFH mit 256 verschiedenen Adreßwerten zur Verfügung, sondern mit den Schaltern S1 bis S4 können lediglich 16 verschiedene Signalzustände (Bausteinadressen) eingestellt werden. Da diese Schalter den vier höherwertigen Adreß-Bits A4 bis A7 zugeordnet sind, kann die Baugruppennummer nur die hexadezimalen Werte von 0X bis FX annehmen. Das X steht hier für die vier niederwertigen Adreß-Bits A0 bis A3, die beim Adreßvergleich nicht benutzt werden. Die Tabelle in Bild 57 zeigt die Bildung der möglichen Baugruppennummern.



A7	A6	A5	A4	A3	A2	A1	A0 (C/ $\overline{D}$ )	$\overline{IOW}$ ( $\overline{WR}$ )	$\overline{IOR}$ ( $\overline{RD}$ )	Betriebszustand Nr. ....
1	1	1	1	X	X	X	X	1	1	1 (nicht aktiv)
1	1	1	1	X	X	X	0	0	1	2 (Prozessor schreibt Daten in das Senderregister)
1	1	1	1	X	X	X	1	0	1	3 (Prozessor schreibt Daten in das Betriebsarten- oder Kommandoregister)
1	1	1	1	X	X	X	0	1	0	4 (Prozessor liest Daten aus dem Empfängerregister)
1	1	1	1	X	X	X	1	1	0	5 (Prozessor liest das Statusregister)

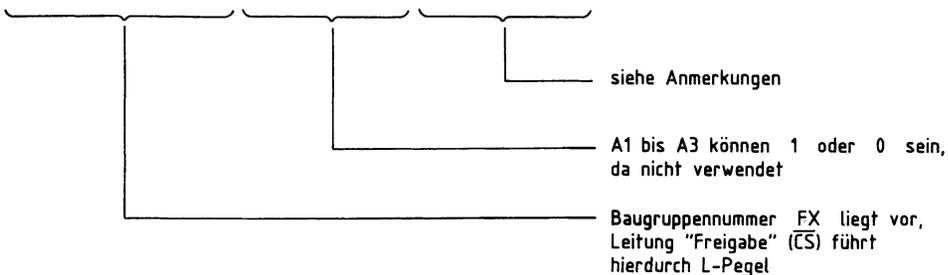


Bild 58: Die Adressierung und die Schreib/Lese-Steuerung in der programmierbaren Serien-Schnittstelle.

Die Adreßleitungen A4 bis A7 besitzen die in Bild 58 angegebenen Signalzustände, die der Baugruppennummer FXH entsprechen. Der serielle Schnittstellenbaustein 8251 erhält L-Pegel am Anschluß  $\overline{CS}$  und ist freigegeben. Da die Adreßleitungen A1 bis A3 nicht angeschlossen sind, kann ihr logischer Pegel 1 oder 0 sein. Dies wird in der Tabelle durch die Bezeichnung X ausgedrückt. Die Betriebszustände werden nachfolgend erläutert.

#### Betriebszustand 1:

Da die Schreibleitung  $\overline{IOW}$  und die Leseleitung  $\overline{IOR}$  H-Pegel (1-Signal) führen, werden weder Daten in die Baugruppe geschrieben, noch werden Daten von der Baugruppe gelesen. Der Signalzustand der Adreßleitung A0 ist daher beliebig. In diesem Betriebszustand verhält sich die programmierbare Serienschnittstelle passiv. Dennoch muß dieser Fall beachtet werden, da er immer auftritt, wenn der Prozessor eine Speicherzeile mit der Adresse XXFXH anspricht.

Betriebszustand 2:

Der Prozessor steuert die Schreibleitung  $\overline{IOW}$  auf L-Pegel. In diesem Betriebszustand schreibt er Daten in die Baugruppe. Da die Adreßleitung A0 (bzw. der Anschluß C/D) einen L-Pegel aufweist, werden die Daten in das Senderegister geschrieben (siehe auch Bild 55). Wegen der nicht benutzten Adreßleitungen A1 bis A3 kann das Senderregister unter folgenden Adressen angesprochen werden: F0H, F2H, F4H, F6H, F8H, FAH, FCH und FEH. Diese Mehrfachadressierung ist durch die einfache Schaltung bedingt, die nicht alle Adreßleitungen zur Adreßbildung verwendet. Von den acht Adreßmöglichkeiten wird hier der Wert F0H verwendet, um Daten an das Senderregister auszugeben.

Betriebszustand 3:

Die Schreibleitung  $\overline{IOW}$  führt weiterhin L-Pegel, aber die Adreßleitung A0 hat H-Pegel. Da bei H-Pegel an A0 eines der Steuer-Register angesprochen wird, gelangen die Prozessordaten jetzt als Steuerworte in das Betriebsarten- oder das Kommandoregister (Bild 55). Die Initialisierung der seriellen Schnittstelle kann unter folgenden Adressen erfolgen: F1H, F3H, F5H, F7H, F9H, FBH, FDH und FFH. Hier wird davon der Wert F1H verwendet.

Beachtet werden muß, daß beide Steuer-Register Steuerworte erhalten müssen, bevor der Baustein betriebsbereit ist. Hierbei ist festgelegt, daß zuerst das Betriebsarten- und danach das Kommandoregister beschrieben wird. Weitere Steuerworte gelangen stets in das Kommandoregister und überschreiben seinen vorherigen Zustand, wodurch die Programmierung des Bausteins, soweit das Kommandoregister Einfluß besitzt, verändert werden kann.

Etwas schwieriger ist es, auch den Inhalt des Betriebsartenregisters zu verändern, da alle Steuerworte, die nach der Initialisierung ausgegeben werden, stets in das Kommandoregister gelangen. Ein innerhalb des Kommandoregisters reserviertes und speziell hierfür vorgesehenes Bit ermöglicht eine Umschaltung auf das Betriebsartenregister auf folgende Weise: Wird dieses Steuerbit des Kommandoregisters gezielt angesprochen, so schreibt der Baustein das darauffolgende Steuerwort in das Betriebsartenregister ein. Diese Vorgehensweise wird Software-Reset genannt und ermöglicht es, den Baustein auch ohne Reset-Signal des Systembus vollständig neu zu programmieren.

Betriebszustand 4:

$\overline{IOW}$  liegt wieder auf H-Pegel, aber die Leseleitung  $\overline{IOR}$  führt L-Pegel. Da die Adreßleitung A0 ebenfalls L-Pegel besitzt, werden in diesem Betriebszustand vom Prozessor die Daten des Empfängerregisters gelesen (siehe auch Bild 55). Das Lesen der Daten ist unter den gleichen Adressen wie im Betriebszustand 2 möglich und erfolgt hier unter der Adresse F0H.

Betriebszustand 5:

$\overline{\text{IOR}}$  führt weiterhin L-Pegel, aber A0 hat H-Pegel angenommen. In diesem Betriebszustand wird vom Prozessor das Statusregister der seriellen Schnittstelle gelesen. Hierbei sind wieder die 8 Adressen des Betriebszustandes 3 möglich, von denen hier der Wert F1H verwendet wird.

Der Betriebszustand, daß  $\overline{\text{IOW}}$  und  $\overline{\text{IOR}}$  beide L-Pegel haben, tritt nicht auf. Liegt eine andere Adresse als die gewählte Baugruppennummer vor, so ist die programmierbare Serienschnittstelle nicht aktiv (wie im Betriebszustand 1).

### Der Baudraten-Teiler

Das an den Anschlüssen  $\overline{\text{RxC}}$  und  $\overline{\text{TxC}}$  des Schnittstellenbausteins angeschlossene TTL-Rechtecksignal bestimmt durch seine Frequenz die Datenübertragungs-Rate (Baudrate). Für vier unterschiedliche Baudraten werden die erforderlichen Taktfrequenzen aus dem 2-MHz-Systemtakt gewonnen (Bild 54). Hierzu dienen mehrere Frequenzteilerstufen: Bild 59.

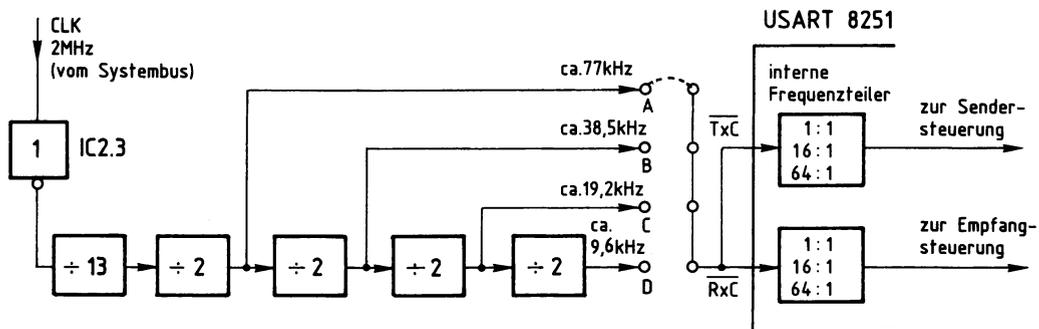


Bild 59: Der Aufbau des Baudratenteilers; vgl. aber die Variante ab Seite 446.

Der Baudratenteiler besteht aus einem 13:1-Frequenzteiler und vier nachgeschalteten 2:1-Teilerstufen. Angesteuert wird er durch den von IC2.3 »aufgefrischten« 2-MHz-Systemtakt. An den vier Ausgängen A, B, C und D treten die angegebenen Frequenzwerte auf. Jeweils eines dieser Signale ist durch eine Brücke auf die miteinander verbundenen Takteingänge  $\overline{\text{RxC}}$  und  $\overline{\text{TxC}}$  des seriellen Schnittstellenbausteins geschaltet. Beachtet werden muß, daß immer nur eine einzige Brücke geschlossen sein darf.

Innerhalb des Schnittstellenbausteins befinden sich hinter den Takteingängen  $\overline{\text{RxC}}$  und  $\overline{\text{TxC}}$  zusätzliche Frequenzteilerstufen (in Bild 54 nicht eingezeichnet), deren Teilungsrate durch die Initialisierung auf die Werte 1:1, 16:1 und 64:1 programmierbar sind. Davon können im hier vorgestellten Ausbau jedoch nur die Teilverhältnisse 16:1 und 64:1 genutzt werden. Das Teilverhältnis von 1:1 ist vom Bausteinhersteller für besondere Anwendungen vorgesehen. Unter Berücksichtigung der internen Teilerfaktoren ergeben sich die in Bild 60 aufgeführten möglichen Baudraten der programmierbaren Serienschnittstelle. Welche der Baudraten benutzt wird, hängt vom jeweiligen Anwendungsfall der programmierbaren Seriellschnittstelle ab.

Lötbrücke bei ...	Frequenzen an $\overline{\text{RxC}}$ und $\overline{\text{TxC}}$	Baudraten bei einem internen Teilverhältnis von	
		16 : 1	64 : 1
A	ca. 77 kHz	4800 Bd	1200 Bd
B	ca. 38,5 kHz	2400 Bd	600 Bd
C	ca. 19,2 kHz	1200 Bd	300 Bd
D	ca. 9,6 kHz	600 Bd	150 Bd

Bild 60: Die möglichen Baudraten der Seriell-Schnittstelle; vgl. aber Seite 446 ff.

### Der vollständige Anschluß des Schnittstellenbausteins 8251

Der serielle Schnittstellenbaustein 8251 besitzt zusätzliche Anschlüsse, auf die bisher nicht eingegangen wurde. Zum Gesamtverständnis der Schaltung und der Programmierung der seriellen Schnittstelle muß man die Aufgaben dieser Anschlüsse kennen, auch wenn sie nicht benötigt werden. In Bild 61 sind alle Signalanschlüsse des Schnittstellenbausteins 8251 dargestellt. Die folgenden Erklärungen berücksichtigen nur den Asynchronbetrieb; Hinweise zum Synchronbetrieb sind den Datenbüchern zu entnehmen.

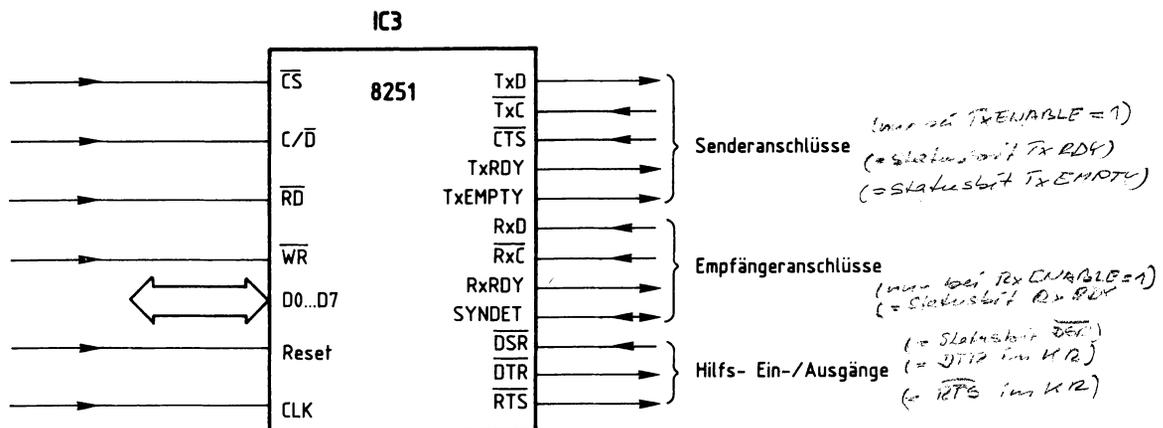


Bild 61: Alle Signalanschlüsse des Bausteins 8251.

Die Anschlüsse auf der linken Seite von Bild 61 führen zum Systembus des Prozessors und wurden schon behandelt. Die Anschlüsse auf der rechten Seite stellen die Verbindung zur Peripherie dar, wobei aber einige davon auch den Mikroprozessor steuern können. Insgesamt kann man zwischen Senderanschlüssen, Empfängeranschlüssen und Hilfs-Ein/Ausgängen unterscheiden.

Senderanschlüsse:

**TxD** (Transmitter Data, Senderdaten)

Hier gibt der Baustein die Senderdaten in serieller Form ab (wurde bereits behandelt).

**$\overline{\text{TxC}}$**  (Transmitter Clock, Sendertakt)

Dieser Anschluß erhält vom Baudratenteiler den Sendertakt (wurde bereits behandelt).

**$\overline{\text{CTS}}$**  (Clear To Send, Senderfreigabe)

L-Pegel an diesem Eingang gibt den seriellen Datensender frei, und H-Pegel sperrt ihn. Falls der Sender beim Anlegen des H-Pegels gerade arbeitet, wird vor dem Sperren noch das Senderregister geleert, d. h. eine gerade stattfindende Datenübertragung wird noch beendet. Diese Art der Steuerung nennt man Hardware-Freigabe des Senders, da sie durch einen logischen Pegel innerhalb der Schaltung erfolgt.

Damit der Sender bei L-Pegel am  $\overline{\text{CTS}}$ -Anschluß arbeiten kann, muß noch eine weitere Bedingung erfüllt sein: Bei der Programmierung des Bausteins muß im Kommandoregister zusätzlich das Steuerbit TxENABLE (Senderfreigabe) gesetzt worden sein. Dies nennt man Software-Freigabe des Datensenders. Sie wird ab Seite 429 näher behandelt.

TxRDY (Transmitter Ready, Sender fertig. Besser: Sender bereit zur Aufnahme des nächsten auszugebenden Zeichens vom Prozessor)

Der Ausgang TxRDY meldet durch H-Pegel, daß das Senderregister leer ist und der Datensender vom Prozessor ein neues Zeichen aufnehmen kann. Ist das Senderregister nicht bereit, ein Zeichen aufzunehmen, führt TxRDY L-Pegel. Das Signal TxRDY kann als Hardware-Statusmeldung (Zustandsmeldung durch einen log. Pegel in der Schaltung) bezeichnet werden, mit der z. B. ein Prozessor-Interrupt ausgelöst werden kann.

Hier wird davon allerdings kein Gebrauch gemacht, da sich der Zustand des Senderregisters außer am Anschluß TxRDY zusätzlich im gleichbenannten Bit TxRDY des Statusregisters ausdrückt. Der Prozessor kann das Statusregister lesen und dieses Bit prüfen. Hierdurch stellt er fest, ob er ein neues Zeichen ausgeben darf oder nicht. Diese Art der Zustandskontrolle nennt man Status-Abfrage durch Software, da hierzu Programmschritte erforderlich sind; vgl. Seite 434 ff.

Im Zusammenhang mit dem Anschluß TxRDY und dem ebenfalls vorhandenen Anschluß TxEMPTY muß folgendes beachtet werden:

Innerhalb des Datensenders befindet sich neben dem Senderregister ein Schieberegister zur parallel/seriell-Wandlung der Zeichen. Nachdem (innerhalb des Datensenders) ein Zeichen vom Senderregister in das Schieberregister übergeben wurde, kann das Senderregister schon wieder ein neues Zeichen übernehmen, obwohl das Schieberregister noch das vorher erhaltene Zeichen ausgibt.  $\Rightarrow \text{TxRDY} = \text{H}$

Die Steuerung dieser Vorgänge erfolgt automatisch. Der Anschluß TxRDY meldet daher nicht die vollständige serielle Ausgabe eines Zeichens, sondern nur die Bereitschaft des Senderregisters, ein neues Zeichen aufzunehmen. Weitere Einzelheiten hierzu müssen den Datenbüchern entnommen werden.

TxEMPTY (Transmitter Empty, Datensender leer)

Der Ausgang TxEMPTY hat H-Pegel, wenn alle Zeichen des Datensenders (Senderregister und Schieberegister) vollständig ausgegeben wurden. Sobald das Senderregister ein neues Zeichen vom Prozessor erhält, nimmt TxEMPTY wieder L-Pegel an. Durch TxEMPTY wird somit das Ende einer Datenübertragung markiert, wenn vom Prozessor keine weiteren Zeichen in das Senderregister übergeben werden und alle vorherigen Zeichen ausgegeben sind.

Der Signalzustand des Ausgangs TxEMPTY kann vom Prozessor über ein Bit des Statusregisters abgefragt werden; vgl. Seite 434 ff.

Empfängeranschlüsse:

RxD (Receiver Data, Empfängerdaten)

An diesem Anschluß empfängt der Schnittstellenbaustein einen seriellen Datenstrom (wurde bereits behandelt).

$\overline{\text{RxC}}$  (Receiver Clock, Empfängertakt)

Dieser Anschluß erhält vom Baudratenteiler den Empfängertakt (wurde bereits behandelt).

RxRDY (Receiver Ready, Empfänger fertig. Besser: Im Empfängerregister liegt ein Zeichen vor, das vom Prozessor gelesen werden kann)

Mit H-Pegel am Anschluß RxRDY signalisiert der Schnittstellenbaustein, daß im Empfängerregister ein neues Zeichen (Datenwort ohne Start-, Stop- und Paritäts-Bits) bereitsteht, das vom Prozessor gelesen (»abgeholt«) werden soll. Liest der Prozessor das Zeichen durch einen IN-Befehl ein, geht RxRDY wieder auf L-Pegel. Damit RxRDY wirksam werden kann, muß der Empfänger aber bei der Programmierung des Bausteins freigegeben worden sein, was durch gezieltes Setzen des Steuerbits RxENABLE (Empfängerfreigabe) im Kommandowort geschieht. Dies wird ab Seite 429 behandelt. *Status-Bit RxRDY = Anzeigung RxRDY*

SYNDET (Synchronisations-Detektor)

Dieser Anschluß hat im Synchronbetrieb eine andere Bedeutung als im Asynchronbetrieb. Er wird in der hier vorgestellten Baugruppe nicht benötigt und daher nicht weiter behandelt.

Hilfs-Ein/Ausgänge:

$\overline{\text{DSR}}$  (Data Set Ready, Betriebsbereitschaft eines anderen Gerätes)

Dieser Eingang läßt sich für beliebige Zwecke verwenden. Sein Signalzustand kann über das Statuswort abgefragt werden. Mit Hilfe des Anschlusses DSR ist der Prozessor z. B. in der Lage, die Betriebsbereitschaft eines Gerätes der Peripherie festzustellen. Hierzu muß dieses Gerät L-Pegel auf die DSR-Leitung legen.

DTR (Data Terminal Ready, Betriebsbereitschaft des eigenen Gerätes)

Dieser Ausgang läßt sich für beliebige Zwecke verwenden. Sein Signalzustand wird bei der Programmierung des Bausteins durch das Kommandowort festgelegt und kann danach auch wieder verändert werden. Durch L-Pegel auf der DTR-Leitung kann der Prozessor einem Peripheriegerät mitteilen, daß die programmierbare Serienschnittstelle betriebsbereit ist.

$\overline{\text{RTS}}$  (Request To Send, Sendeteil einschalten)

Dieser Ausgang läßt sich für beliebige Zwecke verwenden. Sein Signalzustand wird bei der Programmierung des Bausteins durch das Kommandowort festgelegt und kann danach auch wieder verändert werden. Durch L-Pegel auf der  $\overline{\text{RTS}}$ -Leitung kann der Prozessor ein anderes Gerät steuern und z. B. dort den Sendeteil einschalten.

EA Ende

### Der Stromlaufplan der Seriell-Baugruppe

Die Baugruppe (siehe Bild 62) erhält die drei Betriebsspannungen +5 V, +12 V und -12 V. Diese werden durch die Kondensatoren C1 bis C3 abgeblockt. Fast alle Bausteine arbeiten mit +5 V, nur IC7 wird mit  $\pm 12$  V betrieben. IC7 und IC8 sind V-24/TTL- bzw. TTL/V-24-Pegelwandler, die für bestimmte Anwendungsfälle nachbestückt werden können; vgl. Seite 446ff. Bei der seriellen Datenübertragung hat sich ein Spannungspegel durchgesetzt, bei dem die Übertragungsleitung die Werte +12 V und -12 V annimmt, wobei -12 V dem 1-Signal entspricht. Die Bezeichnung »V-24-Pegelwandler« ergibt sich aus dem Spannungshub von insgesamt 24 V.

Die Busanschlüsse RX und TX führen zur CPU-Baugruppe, die ebenfalls einen seriellen Datensender und einen Empfänger mit Pegelwandlern besitzt. Im normalen Betrieb wird der serielle Datenstrom der ASCII-Tastatur und des Video-Interfaces (vgl. Kapitel 4.5) über die Leitungen RX bzw. TX und die Anschlüsse SID (Seriell Input Data) und SOD (Seriell Output Data) der CPU 8085 geführt. Für bestimmte Anwendungen oder zu Übungszwecken können die Daten der Tastatur über RX und IC8 an den Empfänger der programmierbaren Serienschnittstelle gegeben werden. Die Senderdaten der Schnittstelle (TxD) sind über IC7 auf die Leitung TX des Systembus aufschaltbar. Entsprechend müßten dazu die Brücken P3 und P5 bzw. P4 und P6 geschlossen werden. Beachten Sie, daß niemals zwei Datensender auf einer Leitung arbeiten dürfen!

IC1 und IC2.1 erzeugen das erforderliche  $\overline{\text{CS}}$ -Signal für IC3, wenn die richtige Baugruppennummer vorliegt. IC2.2 invertiert das L-aktive Reset-Signal des Systembusses. IC2.3 frischt den 2-MHz-Systemtakt auf.



IC4, IC5.1 und IC5.2 sind so geschaltet, daß IC4 eine Frequenzteilung von 13:1 durchführt (s.o.). IC6 besitzt vier 2:1-Teiler zur Erzeugung der Taktfrequenzen für vier unterschiedliche Baudraten, die mit den Brücken A bis D ausgewählt werden können. Wichtig: Es darf immer nur eine der vier Brücken geschlossen sein.

IC2.4 invertiert das serielle Ausgangssignal des Schnittstellenbausteins. Somit stehen die Signale TxD und  $\overline{\text{TxD}}$  zur Verfügung.

Die Empfängerdaten werden über den Anschluß RxD zur seriellen Schnittstelle geführt. Bei geschlossener Brücke P1 können dem seriellen Empfänger über  $\overline{\text{RxD}}$  invertierte Daten zugeführt werden. In diesem Fall muß der Anschluß RxD freibleiben. Mit den Anschlüssen INTR, RST5.5, RST6.5 und RST7.5 können Interrupts bei der CPU 8085 ausgelöst werden.

### Die Programmierung des seriellen Schnittstellenbausteins 8251

Bevor der serielle Schnittstellenbaustein 8251 Daten aussenden oder empfangen kann, muß er initialisiert, d.h. in seiner Betriebsart programmiert werden. Hierzu gibt der Prozessor zwei Steuerbytes in festgelegter Reihenfolge aus: Das erste (Betriebsarten-Wort genannte) Steuerbyte gelangt in das Betriebsartenregister, das zweite (Kommando-Wort) in das Kommandoregister (Bilder 54 und 55). Beide Register besitzen die gleiche Portadresse. Die erforderlichen Steuerworte können im Prozessor durch den Befehl »MVI A, konst« gebildet und durch »OUT Adr« ausgegeben werden.

#### Das Betriebsarten-Wort

Bild 63 zeigt die Bildung des Betriebsarten-Wortes. Die Benennung der einzelnen Bits ist dem Datenbuch des Baustein-Herstellers entnommen. Sie haben folgende Bedeutung:

B1 (D0) und B2 (D1), Baudraten- und Betriebsarten-Bits:

Mit B1 und B2 wird die Betriebsart und das interne Teilverhältnis des seriellen Schnittstellenbausteins festgelegt.

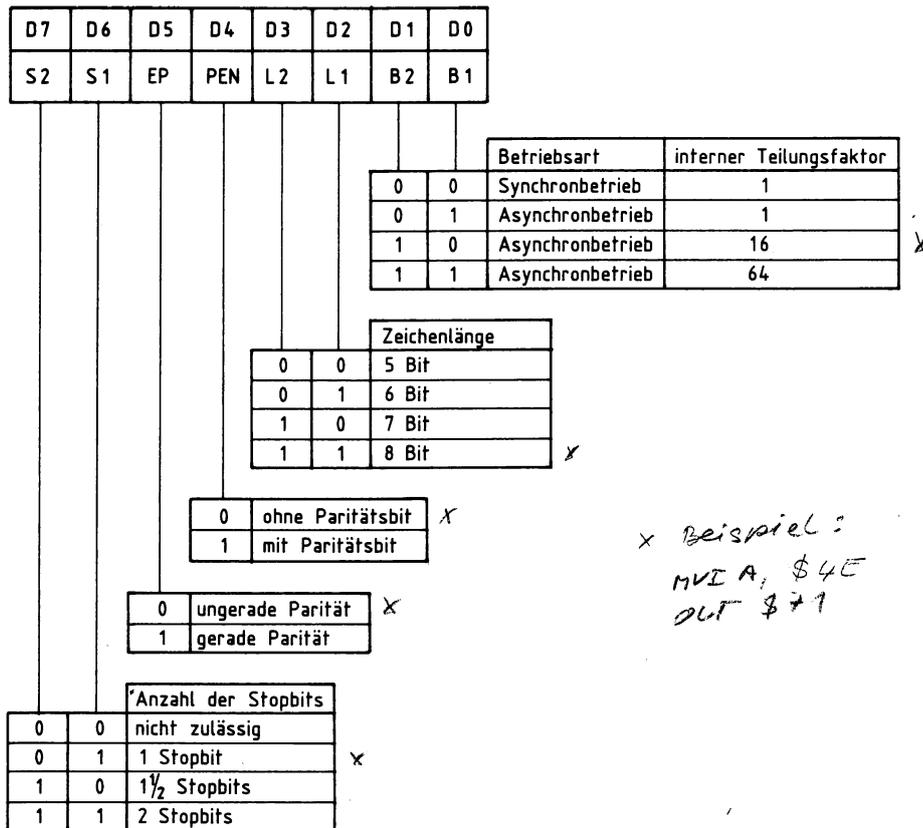


Bild 63: Das Betriebsarten-Wort des Bausteins 8251.

EA ↘ L1 (D2) und L2 (D3), Längen-Bits:

L1 und L2 des Betriebsartenwortes bestimmen die Länge der Datenübertragung, womit die Anzahl der Datenbits innerhalb einer Übertragungssequenz gemeint ist (nicht die Gesamtlänge!). Meist wird eine Übertragung von 7 Bits (z.B. für ASCII-Zeichen) oder 8 Bits (für allgemeine Datenübertragung) gewählt. Eine 5-Bit-Übertragung ist z.B. für den Baudot-Code (5-Bit-Fernschreib-Code) erforderlich.

PEN (D4), Parity Enable, Paritätsprüfung ein- und ausschalten:

Bei L-Signal (0-Pegel) bildet der Sender kein Paritätsbit. Außerdem prüft der Empfänger nicht die Parität empfangener Zeichen. Bei H-Signal fügt der Sender automatisch Paritätsbits in den Datenstrom ein, und der Empfänger überprüft die Parität der empfangenen Zeichen. Das Ergebnis der Paritätsprüfung steht im Statusregister.

EP (D5), Even Parity, gerade Parität:

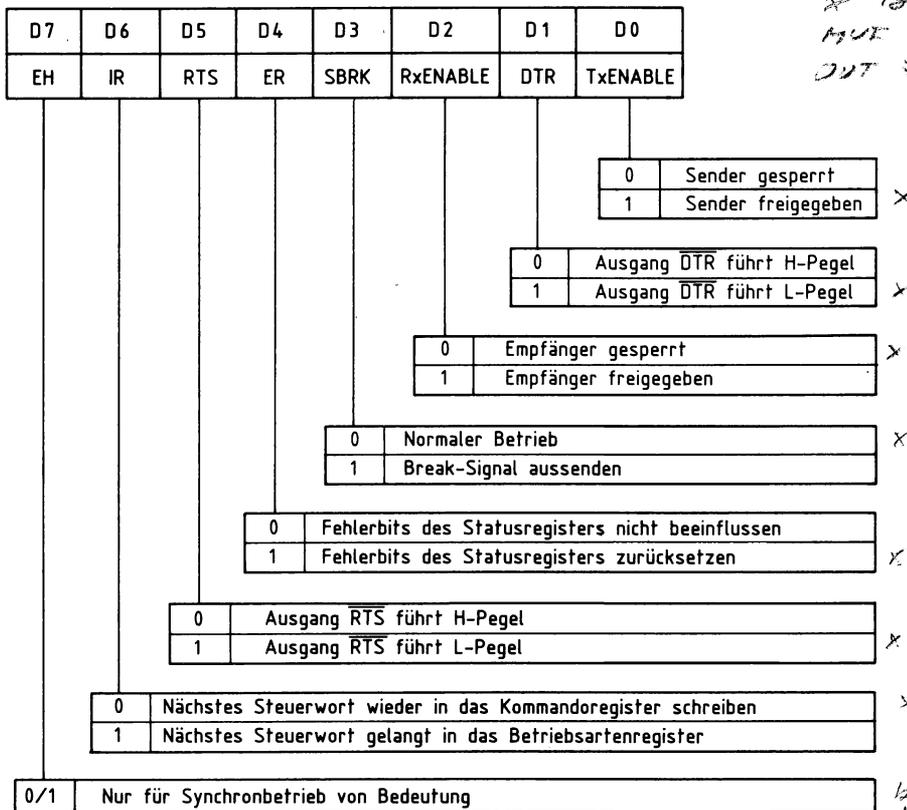
Falls das Paritätsbit mit dem Steuerbit PEN (D4) eingeschaltet wurde bzw. eine Paritätsprüfung erfolgen soll, kann mit D5 zwischen der geraden (bei 1-Signal von D5) und der ungeraden Parität (bei 0-Signal) gewählt werden.

S1 (D6) und S2 (D7), Anzahl der Stopbits:

Im Asynchronbetrieb wird hiermit die Anzahl der Stopbits festgelegt. Im Synchronbetrieb haben diese Bits eine andere Bedeutung. EA-Ende

### Das Kommando-Wort

Das Kommando-Wort, mit dessen Hilfe mehrere Funktionen gesteuert werden können, gelangt nach der Programmierung des Betriebsartenregisters stets in das Kommandoregister.



\* Beispiel  
MUT 9, 337  
DUT 871

beliebig bei Asynchron =  
beispielsweise  
\* 0

Bild 64: Das Kommando-Wort des Bausteins 8251.

EA ↓

Die Bezeichnung und Bedeutung der einzelnen Kommando-Bits:

TxENABLE (D0), Senderfreigabe:

Mit L-Signal wird der serielle Datensender gesperrt und mit H-Signal freigegeben. Zu beachten ist, daß zur Senderfreigabe zusätzlich der Bausteinanschluß  $\overline{\text{CTS}}$  L-Pegel führen muß (Bild 61). Die Steuerung des Kommandobits TxENABLE nennt man »Software-Freigabe«; die des Anschlusses  $\overline{\text{CTS}}$  heißt Hardware-Freigabe des Senders. Die Senderfreigabe wirkt sich auf den Bausteinanschluß »TxRDY« und das gleichbenannte Bit TxRDY des Statusregisters aus.

Bei freigegebenem Datensender tritt am Schnittstellenausgang TxRDY ein H-Pegel auf, wenn der Datensender bereit ist, ein neues Zeichen aufzunehmen. Ist der Sender jedoch gesperrt, bleibt der Ausgang TxRDY auf L-Pegel, auch wenn der Datensender leer ist und ein Zeichen aufnehmen könnte.

Im allgemeinen wird der Ausgang TxRDY dazu verwendet, einen Prozessor-Interrupt auszulösen, durch den ein neues Zeichen in das Senderregister eingeschrieben wird. Hierbei wirkt die Senderfreigabe so, daß bei H-Signal des Bits TxENABLE das Auftreten des Interrupt-auslösenden H-Signals an TxRDY möglich ist. Bei gesperrtem Sender (Kommando-Bit TxENABLE auf L-Pegel) bleibt TxRDY auf L-Pegel, wodurch kein Interrupt ausgelöst werden kann. Hier wird hiervon allerdings kein Gebrauch gemacht, da die Information, daß der Datensender ein neues Zeichen aufnehmen kann, auch im Bit TxRDY des Statusregisters steht. Falls man ohne Interrupts arbeitet, muß der Prozessor dieses Zustandsbit prüfen, um hierdurch festzustellen, ob der Sender frei ist oder nicht. Bei gesperrtem Sender (Bit TxENABLE des Kommandowortes hat L-Pegel) ist das Statusbit TxRDY ständig auf L-Pegel gesetzt und signalisiert hierdurch, daß der Sender blockiert ist.

DTR (D1), Ausgang  $\overline{\text{DTR}}$  (Data Terminal Ready) setzen:

Dieses Bit des Kommandowortes setzt den beliebig zu verwendenden Bausteinanschluß  $\overline{\text{DTR}}$  auf H- oder L-Pegel, wobei für H-Pegel am Ausgang ein L-Pegel des Steuerbits erforderlich ist (Bild 61).

RxENABLE (D2), Empfängerfreigabe:

Mit H-Pegel wird der serielle Datensender freigegeben und mit L-Pegel gesperrt. Das Kommandobit RxENABLE wirkt sich intern auf den Bausteinanschluß RxRDY und auf das gleichbenannte Bit RxRDY des Statusregisters aus.

Bei freigegebenem Empfänger tritt am Schnittstellenausgang RxRDY H-Pegel auf, wenn im Empfängerregister ein Zeichen vorliegt. Führt Bit D2 des Kommandoregisters jedoch L-Pegel, bleibt der Ausgang RxRDY auf L-Pegel, auch wenn Zeichen empfangen wurden.

Im allgemeinen wird der Ausgang RxRDY dazu verwendet, einen Interrupt auszulösen, durch den ein empfangenes Zeichen in den Prozessor eingelesen wird. Hierbei wirkt die Empfängerfreigabe so, daß bei H-Signal des Bits RxENABLE das Auftreten des Interrupt-auslösenden H-Signals an RxRDY möglich ist. Bei gesperrtem Empfänger (Kommando-Bit RxENABLE auf L-Pegel) bleibt RxRDY auf L-Pegel, wodurch kein Interrupt ausgelöst werden kann.

Hier wird hiervon allerdings kein Gebrauch gemacht, da die Information, daß der Empfänger ein neues Zeichen aufgenommen hat, auch im Bit RxRDY des Statusregisters steht. Falls man ohne Interrupts arbeitet, muß der Prozessor dieses Zustandsbit prüfen, um hierdurch festzustellen, ob ein neues Zeichen vorliegt oder nicht. Bei gesperrtem Empfänger (Bit RxENABLE des Kommandowortes hat L-Pegel) ist das Statusbit RxRDY ständig auf L-Pegel gesetzt und signalisiert hierdurch, daß der Empfänger blockiert ist.

SBRK (D3), Send Break (Sende Unterbrechungs-Signal aus):

Dieses Bit wirkt mit dem Bausteinanschluß SYNDT zusammen, der je nach Betriebsart verschiedene Aufgaben besitzt. Informationen hierzu entnehmen Sie bitte den Datenbüchern.

ER (D4), Error Reset (Fehler-Bits des Statusregisters zurücksetzen):

Das Statusregister besitzt drei Bits, mit denen das Auftreten bestimmter Fehler bei der Datenübertragung signalisiert wird. Bei H-Signal des Bits ER werden die Fehler-Bits des Statusregisters zurückgesetzt. *Einmalig zurückgesetzt.*

RTS (D5), Ausgang  $\overline{\text{RTS}}$  (Request To Send) setzen:

Dieses Bit setzt den beliebig zu verwendenden Ausgang  $\overline{\text{RTS}}$  auf H- oder L-Pegel, wobei das Steuerbit für H-Pegel am Ausgang L-Pegel besitzen muß (Bild 61).

IR (D6), Initialization Request (Umschalten auf das Betriebsartenregister):

Wird dieses Bit auf H-Pegel gesetzt, so gelangt das nächste Steuerwort nicht in das Kommandoregister, sondern in das Betriebsartenregister.

Bei der Initialisierung bzw. Programmierung des Bausteins, zu der nach jedem Kaltstart oder Reset-Signal zwei Steuerworte erforderlich sind, gelangt das erste Steuerwort in das Betriebsartenregister und das zweite, sowie alle folgenden, in das Kommandoregister (Bild 55).

Da es zwei Control-Register (Betriebsarten- und Kommandoregister) gibt, aber zur Unterscheidung zwischen Daten- und Control-Registern nur ein Bausteinanschluß ( $C/\bar{D}$ ) vorhanden ist, wurde die Reihenfolge der Programmierung festgelegt. Das Bit IR des Kommandowortes ermöglicht es, nach der Initialisierung, wenn weitere Steuerworte stets in das Kommandoregister gelangen, doch noch, falls nötig, das Betriebsartenregister zu erreichen.

EH (D7), Enter Hunt Mode (Suchen nach Synchronisationszeichen im Synchronbetrieb): Dieses Bit hat im Asynchronbetrieb, auf den sich diese Beschreibung be-  
*EA-Ende* schränkt, keinen Einfluß.

### Das Statusregister

Das Statusregister wird nicht programmiert, sondern kann vom Prozessor nur gelesen werden. Hierdurch erhält er Zustandsmeldungen über den Datensender, den Datenempfänger und einige Bausteinanschlüsse. Außerdem meldet das Statusregister bestimmte Fehler beim seriellen Datenempfang.

Da die Programmierung des Schnittstellenbausteins in gewisser Weise auch das Statusregister beeinflusst, wird es in diesem Zusammenhang behandelt. Bild 65 zeigt den Aufbau des Statusregisters.

6.2 Serielle Ein/Ausgabe, universell programmierbar

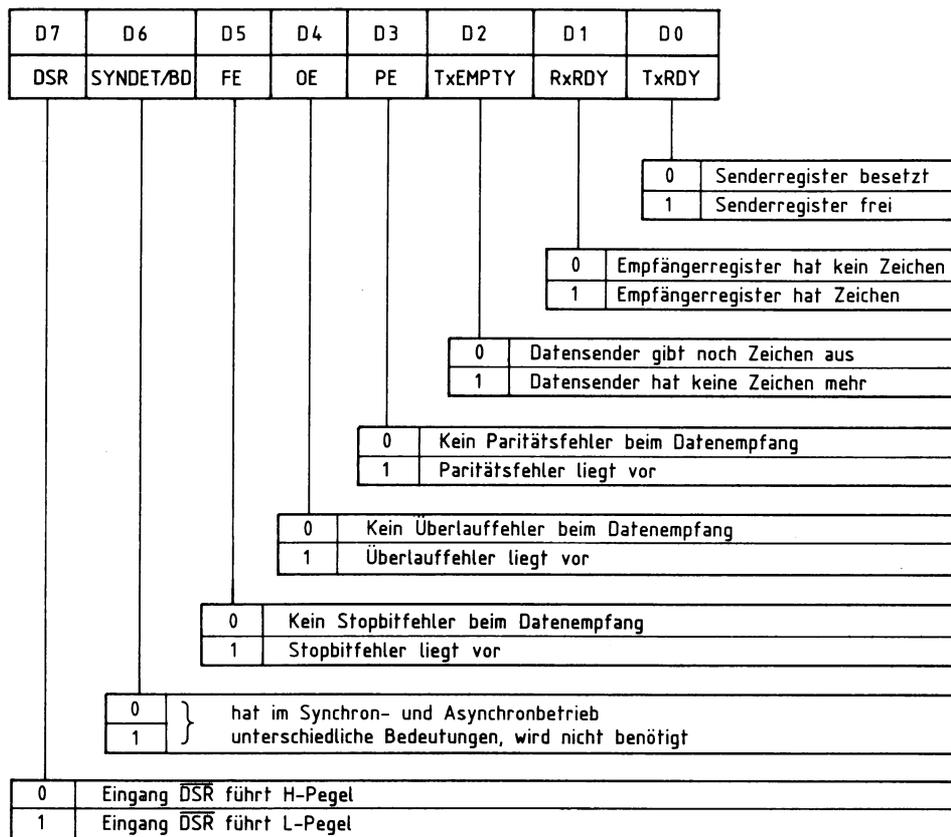


Bild 65: Das Statusregister des Bausteins 8251.

Die Bezeichnung und die Bedeutung der einzelnen Status-Bits:

EA ↓

TxRDY (D0), Transmitter Ready (Sender fertig. Besser: Sender bereit zum Empfang des nächsten auszugebenden Zeichens von der CPU):

Dieses Bit des Statusregisters zeigt dem Prozessor durch H-Pegel an, daß das Senderregister leer ist und vom Prozessor ein neues Zeichen aufnehmen kann. Hierdurch wird der Datenaustausch zwischen Mikroprozessor und Schnittstellenbaustein nach dem Abfrageverfahren (Polling-Verfahren) gesteuert, das hier benutzt wird.

read

Anstelle des Polling-Verfahrens kann die Schnittstelle auch durch Interrupts gesteuert werden. Hierzu ist aber nicht das Statusbit TxRDY, sondern der gleichbenannte Bausteinanschluß TxRDY vorgesehen (Bild 61). Die Funktion des Statusbits TxRDY ist jedoch im Gegensatz zum Bausteinanschluß TxRDY nicht mit dem Bausteinanschluß CTS und dem Kommandobit TxENABLE verknüpft. Weitere Hinweise entnehmen Sie bitte den Datenbüchern.

RxRDY (D1), Receiver Ready (Empfänger fertig. Besser: Im Empfängerregister liegt ein Zeichen vor, das vom Prozessor gelesen werden kann):

Dieses Bit des Statusregisters zeigt dem Prozessor durch H-Pegel an, daß im Empfängerregister ein Zeichen vorliegt, welches vom Prozessor gelesen werden kann. Hierdurch kann der Datenaustausch zwischen Mikroprozessor und Schnittstellenbaustein nach dem Polling-Verfahren gesteuert werden. Das Bit D1 des Statusregisters besitzt die gleiche logische Aussage wie der Bausteinanschluß RxRDY, der zur Interruptsteuerung vorgesehen ist (Bild 61).

TxEMPTY (D2), Transmitter Empty (Datensender leer):

Das Bit D2 des Statusregisters hat H-Pegel, wenn alle Zeichen des seriellen Datensenders (Senderregister und Schieberegister) vollständig ausgegeben wurden. Diese Information entspricht dem logischen Pegel des Bausteinanschlusses TxEMPTY (siehe Bild 61).

PE (D3), Parity Error (Paritätsbit-Fehler):

Bit D3 meldet durch H-Pegel das Auftreten eines Paritätsfehlers, wodurch aber die weitere Arbeitsweise des Empfängers nicht beeinflußt wird. Dieses und die beiden anderen Fehlerbits des Statusregisters können durch ein Kommandowort zurückgesetzt werden.

OE (D4), Overrun Error (Überlauferfehler):

Wenn der Prozessor während des seriellen Datenempfangs das Empfängerregister nicht rechtzeitig liest, kann es vorkommen, daß ein Zeichen, das noch im Empfängerregister steht, bereits durch ein neues Zeichen überschrieben wird. In diesem Falle meldet das Bit D4 des Statuswortes einen Überlauferfehler.

FE (D5), Frame Error (Rahmenfehler. Besser: Stopbit-Fehler):

Durch das auf der Sender- und der Empfängerseite stets gleiche Übertragungsformat kann der Schnittstellenbaustein das Auftreten des ersten Stopbits vorhersagen. Tritt anstelle des erwarteten H-Signals beim Stopbit ein L-Signal auf, meldet Bit D5 des Steuerregisters dies durch H-Signal.

SYNDET/BD (D6):

Dieses Bit des Statusregisters hat im Synchronbetrieb eine andere Bedeutung als im Asynchronbetrieb. Es wird hier nicht ausgewertet.

DSR (D7), Data Set Ready, (Betriebsbereitschaft eines [anderen] Gerätes):

Der Bausteineingang  $\overline{\text{DSR}}$  kann für beliebige Zwecke verwendet werden. Beispielsweise kann ein anderes Gerät dem Prozessor über diesen Anschluß mitteilen, daß es betriebsbereit ist (Bild 61). Führt der Anschluß  $\overline{\text{DSR}}$  einen H-Pegel, so besitzt das Statusbit DSR einen L-Pegel (und umgekehrt).

*EA-Ende*

### Beispiele zur Programmierung des seriellen Schnittstellenbausteins

Zwei Beispiele sollen die Bildung des Betriebsarten- und Kommandowortes sowie die Programmierung des Bausteins verdeutlichen.

Beispiel 1:

Bei 2400 Bd wird folgende Betriebsart gewünscht:

- Asynchrone Datenübertragung mit einem internen Teilverhältnis von 16:1
- Zeichenlänge 7 Bits
- Paritätsprüfung auf gerade Parität
- 1 1/2 Stopbits.

Das Kommandowort soll bewirken:

- Freigabe des Senders und des Empfängers
- Ausgang  $\overline{\text{DTR}}$  führt L-Pegel, Ausgang  $\overline{\text{RTS}}$  führt H-Pegel
- Normalbetrieb (kein Break-Betrieb)
- Fehlerbits des Statusregisters zurücksetzen
- keine Umschaltung auf das Betriebsartenregister.

Für die gewünschte Baudrate und das gewählte Teilerverhältnis ist gemäß Bild 60 die Brücke B erforderlich (siehe auch Stromlaufplan; Bild 62).

Das Betriebsartenwort kann mit Hilfe von Bild 63 gebildet werden. Es ist in Bild 66 dargestellt.

Das Kommandowort wird mit Hilfe von Bild 64 gebildet und ist in Bild 67 dargestellt.

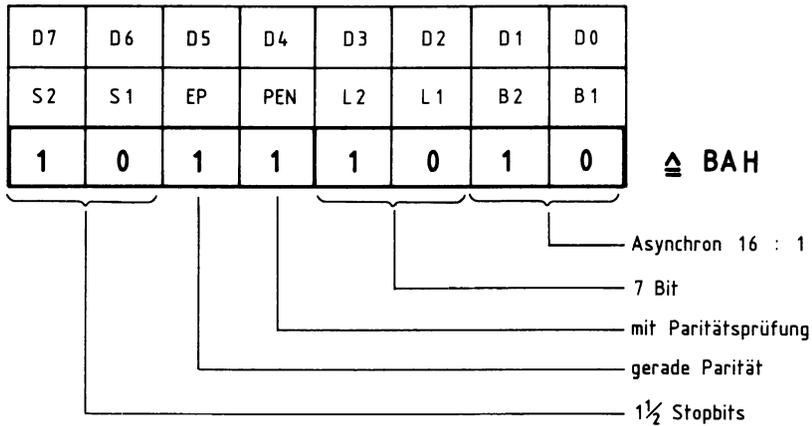


Bild 66: Das Betriebsarten-Wort für Beispiel 1.

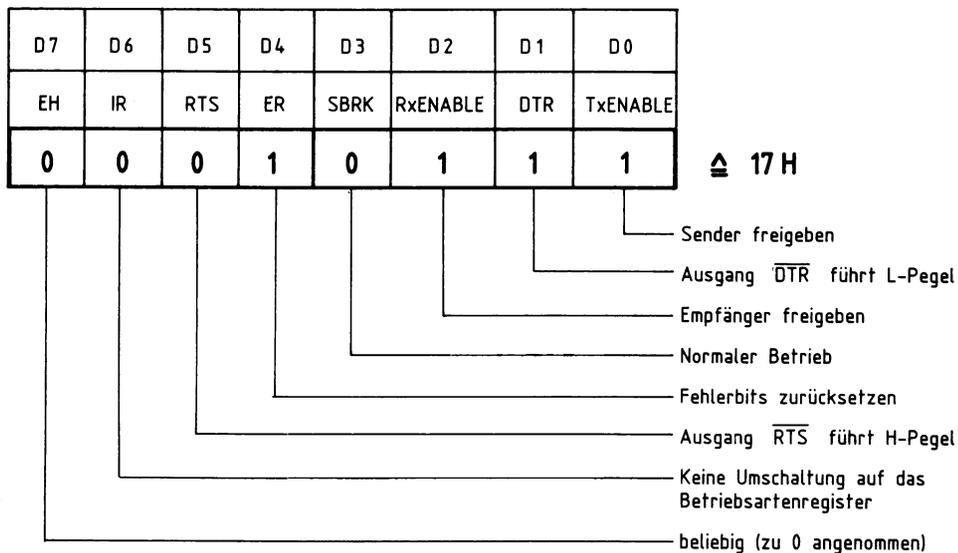


Bild 67: Das Kommando-Wort für Beispiel 1.

Zur Initialisierung des Schnittstellen-Bausteins müssen die beiden Hex-Werte BA und 17 in die Steuer-Register geschrieben werden. Dabei muß zuerst das Betriebsartenwort und danach das Kommandowort ausgegeben werden. Das folgende Programm, das z.B. Teil des Betriebsprogramms sein kann, initialisiert den Schnittstellenbaustein. Hierbei wird die Baugruppennummer FXH verwendet.

Initialisierungsprogramm zum Beispiel 1:

Befehle	Kommentar
...	...
MVI A,0BA	Zuerst das Betriebsartenwort in den Akkumulator des Prozessors laden.
OUT 0F1	Akkuinhalt an Steuer-Register ausgeben.
MVI A,17	Danach das Kommandowort in den Akku laden.
OUT 0F1	Akkuinhalt an Steuer-Register ausgeben.
...	...

Beispiel 2:

Die programmierbare Serienschnittstelle kann z.B. auch für ein Kassetten-Interface eingesetzt werden. Bei der Initialisierung lautet das Betriebsartenwort dann z.B. CFH, und das Kommandowort ist 25H. Wie ist die Schnittstelle programmiert, wenn die Brücke A des Baudratenteilers verwendet wird?

Zur Lösung dieser Aufgabe wird das Betriebsartenwort mit Hilfe von Bild 63 und das Kommandowort mit Hilfe von Bild 64 analysiert. Die Ergebnisse sind in den Bildern 68 und 69 wiedergegeben.

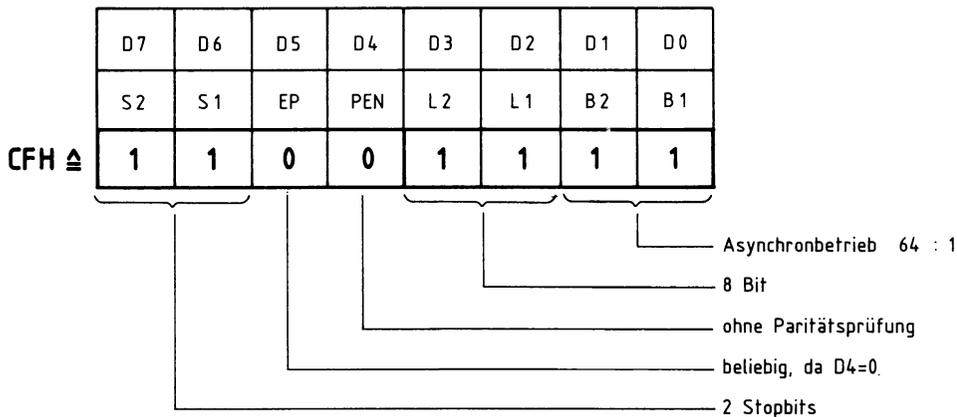


Bild 68: Das Betriebsarten-Wort für Beispiel 2.

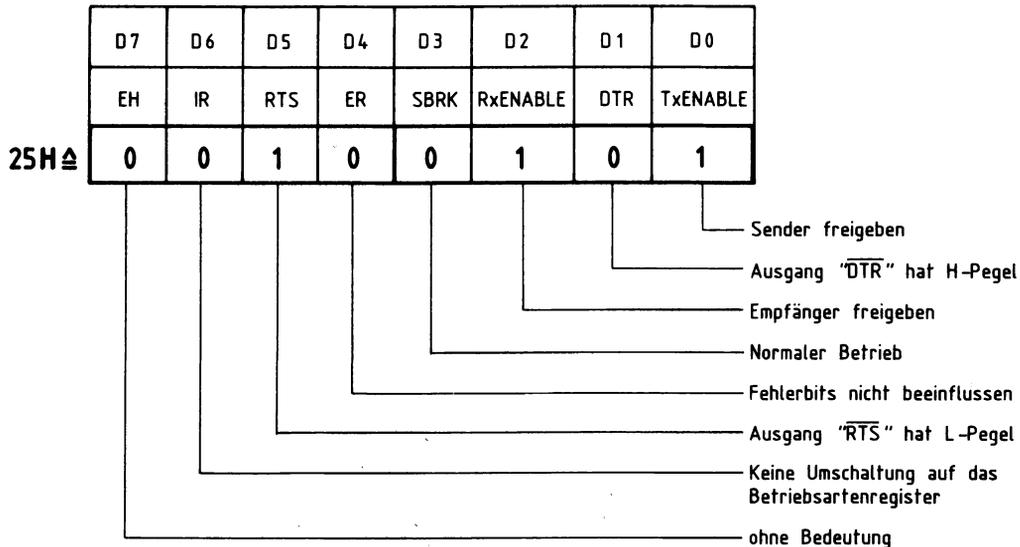


Bild 69: Das Kommando-Wort für Beispiel 2.

Das Kassetten-Interface kann durch das Betriebssystem folgendermaßen programmiert werden:

- asynchrone Betriebsart mit internem Teilerfaktor 64:1
- Wortlänge 8 Bits
- keine Paritätsprüfung
- zwei Stopbits
- Sender und Empfänger freigegeben
- Ausgänge:  $\overline{\text{DTR}}$  hat H-Pegel, und  $\overline{\text{RTS}}$  hat L-Pegel
- Fehlerbits bleiben unbeeinflusst.

Aus Bild 60 ist zu entnehmen, daß sich bei der vorhandenen Baudratenteilung (Brücke A) eine Baudrate von 1200 Bd ergibt.

### Steuerprogramme zum Senden und Empfangen von Zeichen

Nach der Initialisierung ist die Schnittstelle bereit, Zeichen auszusenden und zu empfangen. Da Sender und Empfänger getrennt voneinander arbeiten, kann dies auch gleichzeitig erfolgen. Das Aussenden und der Empfang von Zeichen erfolgt programmgesteuert, wozu der Arbeitsspeicher des Computers spezielle Programme besitzen muß. Beim Interrupt-Verfahren sind dies Interrupt-Service-Routinen, die beim Auftreten eines Interrupts bearbeitet werden.

Bei der Abfrage-Methode (Polling-Verfahren) werden Ein/Ausgabe-Unterprogramme benötigt, die das Statusregister abfragen (überwachen). Das Betriebsprogramm besitzt meist entsprechende Unterprogramme für den Betrieb einer seriellen Schnittstelle als Kassetten-Interface. Andere Anwendungen erfordern eventuell davon abweichende Steuerprogramme. In den folgenden Abschnitten werden daher die wichtigsten Gesichtspunkte einfacher Eingabe- und Ausgabe-Programme behandelt. Dabei wird von der Baugruppennummer FXH ausgegangen.

### Steuerprogramm für den Datensender

Zeichen, die der Datensender in serieller Form ausgeben soll, gelangen stets durch OUT-Befehle aus dem Akkumulator des Prozessors in das Senderregister der Schnittstelle. Vorher muß der Prozessor jedoch das Statusbit TxRDY prüfen, um festzustellen, ob das Senderregister frei oder besetzt ist. Nur bei freiem Senderregister darf er ein neues Zeichen ausgeben. Ist das Senderregister besetzt, so muß der Prozessor warten, bis es freigeworden ist. Da die Statusabfrage den Akkuwert verändert, wird das auszugebende Zeichen im Register C der CPU zwischengespeichert, um anschließend wieder für die Ausgabe bereitzustehen.

Bild 70 zeigt das Flußdiagramm und das Programm SEND für den Datensender. Der erste Befehl (MOV C,A) kopiert den Inhalt des Akkumulators in das Register C. Anschließend wird das Statusregister der seriellen Schnittstelle durch IN 0F1 in den Akkumulator des Prozessors eingelesen. Nun erfolgt die Überprüfung von Bit D0. Hat D0 einen L-Pegel, ist der Datensender besetzt und kann im Moment kein neues Zeichen aufnehmen. Führt D0 jedoch H-Pegel, ist der Datensender frei und kann ein neues Zeichen erhalten.

Zur Überprüfung, ob das Statusbit D0 H- oder L-Pegel besitzt, dient der Befehl ANI 01, der Bit für Bit eine UND-Verknüpfung des Akku-Inhalts mit dem Wert 01 bildet. Diesen Vorgang nennt man maskieren, da durch die Maske 01 die nicht benötigten Bits D1 bis D7 abgedeckt (unterdrückt) werden. Nur D0 bleibt noch wirksam.

Hat D0 L-Pegel, führt die UND-Verknüpfung zum Gesamtergebnis null. In diesem Fall wird der bedingte Sprungbefehl JZ STATUS ausgeführt und das Statusregister erneut überprüft. Dies geschieht so lange, bis D0 durch H-Pegel meldet, daß der Sender frei ist.

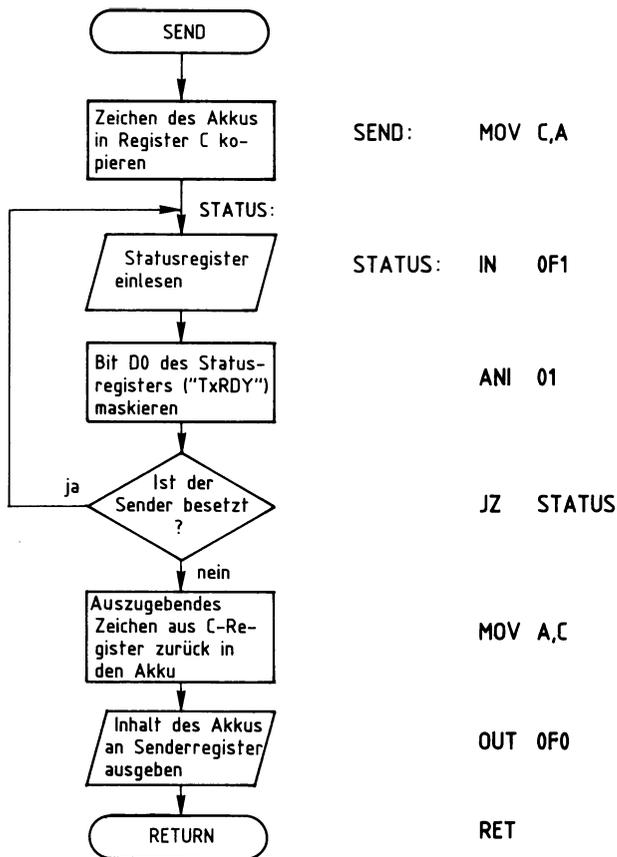


Bild 70: Flußdiagramm und Programm »SEND« für den Datensender.

Nun ist das Gesamtergebnis der UND-Verknüpfung nicht mehr null. Daher wird der Befehl JZ STATUS übergangen und an dessen Stelle der Befehl MOV A,C ausgeführt. Hierdurch gelangt das auszugebende Zeichen aus dem C-Register in den Akkumulator zurück. Der folgende Befehl OUT 0F0 bewirkt die Ausgabe des Zeichens an das Senderregister. Danach wird das Unterprogramm SEND durch den Rücksprungbefehl RET wieder verlassen.

### Steuerprogramm für den Datenempfänger

Seriell empfangene Zeichen werden vom Prozessor aus dem Empfängerregister in den Akkumulator gelesen. Hierzu wird das Statusbit RxRDY (D1) überwacht. Es meldet durch H-Signal, daß ein Zeichen vollständig empfangen wurde. Bild 71 zeigt das Flußdiagramm und das Programm »EMPF« für den Empfänger.

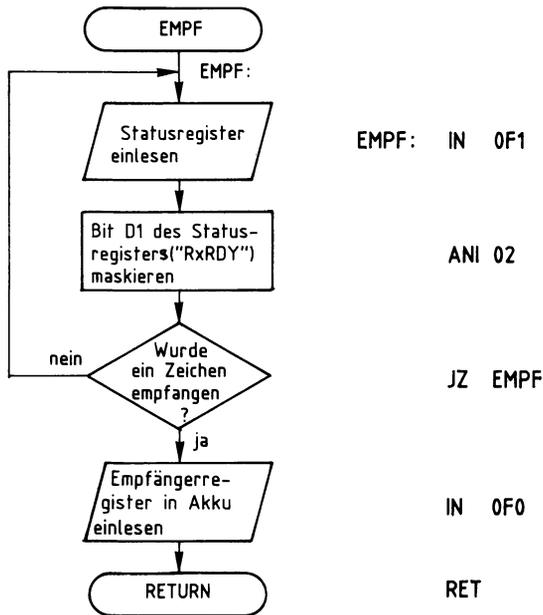


Bild 71: Flußdiagramm und Programm »EMPF« für den Datenempfänger.

Durch den ersten Befehl (IN 0F1) wird das Statusregister der seriellen Schnittstelle in den Akkumulator des Prozessors eingelesen. Anschließend wird Bit D1 überprüft: Solange D1 L-Pegel führt, wurde kein Zeichen empfangen, und das Empfängerregister ist leer. Führt D1 jedoch H-Pegel, liegt im Empfängerregister ein Zeichen vor, das der Prozessor einlesen kann.

Zur Überprüfung, ob das Statusbit D1 (RxRDY) H- oder L-Pegel besitzt, wird es durch den Befehl ANI 02 maskiert. Hat D1 L-Pegel, so führt die UND-Verknüpfung zum Gesamtergebnis null. In diesem Fall wird der bedingte Sprungbefehl JZ EMPF ausgeführt und das Statusregister erneut überprüft, bis D1 durch H-Pegel meldet, daß ein empfangenes Zeichen vorliegt.

Nun wird der Befehl JZ EMPF übergangen und an dessen Stelle der Befehl IN 0F0 ausgeführt. Er liest den Inhalt des Empfängerregisters in den Akkumulator. Danach wird das Unterprogramm EMPF durch den Rücksprungbefehl RET wieder verlassen.

### Ein Testprogramm für die serielle Schnittstelle

Verbindet man den Ausgang TxD des Datensenders mit dem Eingang RxD des Datenempfängers, so empfängt die Baugruppe alle ausgegebenen Daten selbst. Hierdurch läßt sich die Funktion von Sender und Empfänger auf einfache Weise überprüfen; hierzu kann der in Bild 72 gezeigte Aufbau dienen.

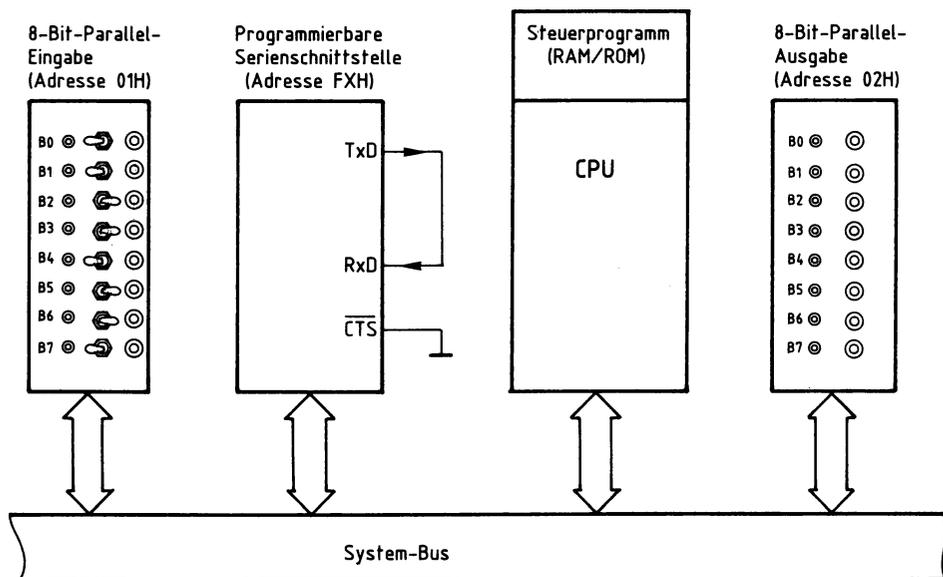


Bild 72: Der Aufbau zum Prüfen der seriellen Schnittstelle.

Die mit den Schaltern B0 bis B7 der 8-Bit-Parallel-Eingabe (siehe Kapitel 4.1) eingestellten Daten werden vom Prozessor gelesen und anschließend an den Sender der seriellen Schnittstelle übergeben. Er ist durch L-Signal an CTS freigegeben und gibt die Daten am Anschluß TxD in serieller Form aus.

Da der Empfängereingang RxD mit dem Senderausgang verbunden ist, werden die gesendeten Daten wieder vom Empfänger aufgenommen. Die CPU liest diese Daten wieder ein und gibt sie zur Kontrolle an die 8-Bit-Parallel-Ausgabe (siehe Kapitel 4.2) aus.

Das zur Steuerung dieser Vorgänge erforderliche Programm ist in Bild 73 wiedergegeben. Es verwendet die bereits beschriebenen Steuerprogramme für den Sender und den Empfänger nach Bild 70 und Bild 71. Eine Initialisierung des Schnittstellenbausteins ist nicht mehr erforderlich, wenn diese beim Einschalten des Gerätes durch das Betriebsprogramm erfolgt. Hinweise zu dieser Initialisierung finden Sie in den Bildern 68 und 69.

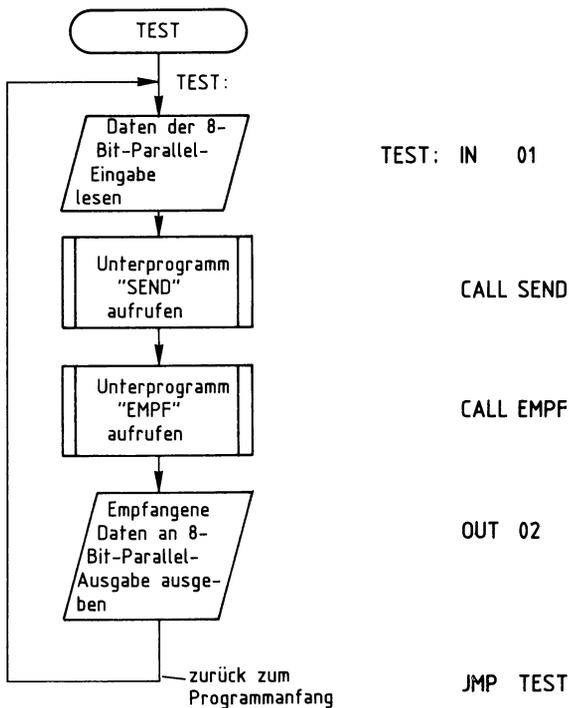


Bild 73: Das Testprogramm für die serielle Schnittstelle.

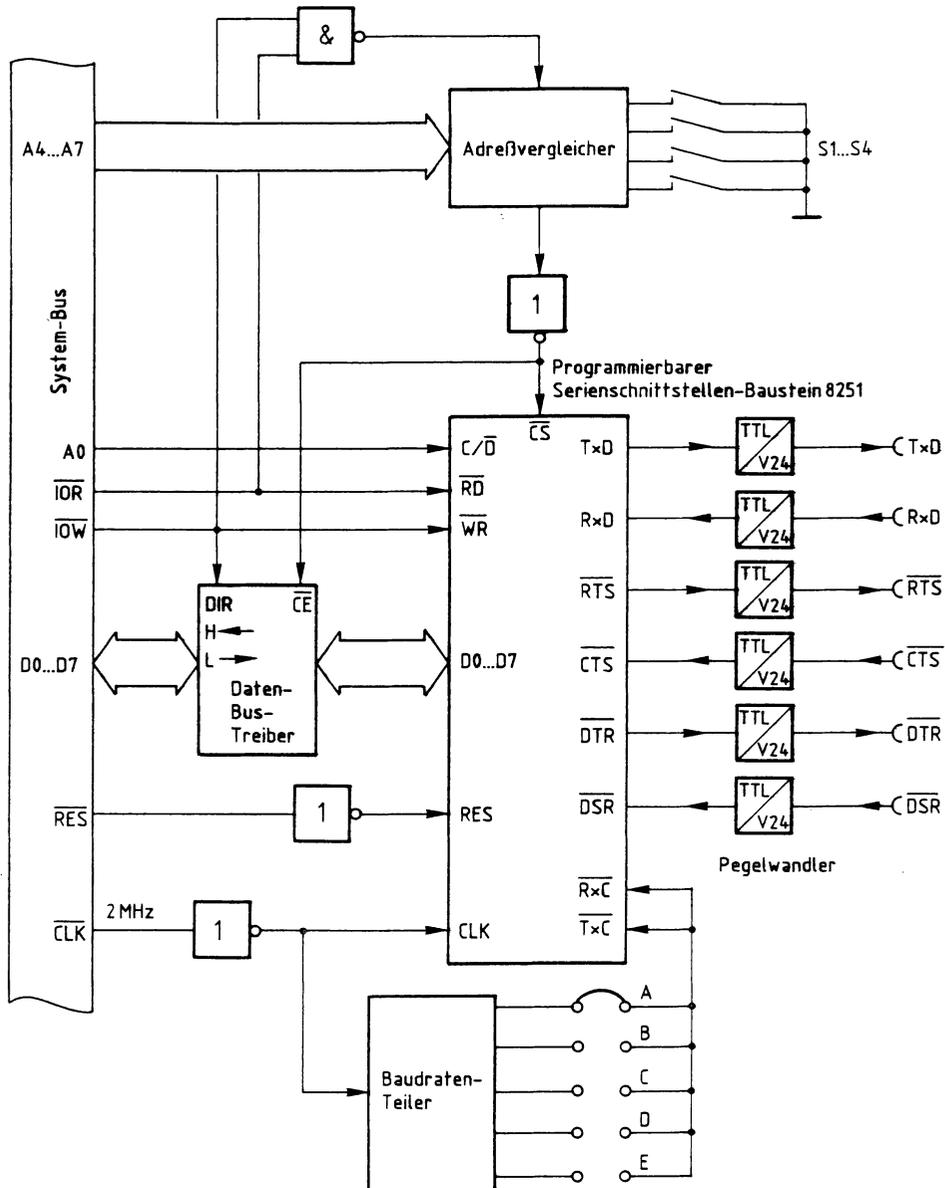


Bild 74: Der Ausbau der seriellen Schnittstelle zur V.24-Schnittstelle; es sind Baudraten bis zu 9600 möglich (Stromlaufplan: Seite 450).

Die nachfolgende Funktionsbeschreibung der V.24-Schnittstelle bezieht sich nur auf das Blockschaltbild (Bild 74). Einzelheiten des verwendeten Schnittstellenbausteins 8251 wurden zuvor behandelt (ab Seite 405).

Die serielle Daten-Sendeleitung TxD, die serielle Daten-Empfangsleitung RxD sowie einige Steuerleitungen des Schnittstellenbausteins 8251 sind über TTL/V24-Pegelwandler mit einer 25poligen Buchsenleiste verbunden.

Für interne Steuerzwecke erhält der Baustein am Anschluß CLK den von einem Inverter aufgefrischten 2-MHz-Systemtakt. Aus dem System-Takt wird außerdem im Baudraten-Teiler der zur parallel/seriell- und seriell/parallel-Wandlung erforderliche Baudraten-Takt gewonnen und dem Baustein an den Anschlüssen  $\overline{\text{RxC}}$  und  $\overline{\text{TxC}}$  zugeführt. Durch Brücken und geeignete Initialisierung können 7 verschiedene Baudraten eingestellt werden (siehe Tabelle im Stromlaufplan, Seite 450).

Der Daten-Bus ist über einen bidirektionalen Datenbus-Treiber in Tristate-Technik mit den Datenleitungen des Schnittstellen-Bausteins verbunden. Der Datenbus-Treiber trennt im Ruhezustand den Datenweg und gewährleistet im aktiven Zustand einen ausreichend hohen Treiberstrom in Richtung System-Bus. Zur Aktivierung des Datenbus-Treibers muß der Anschluß  $\overline{\text{CE}}$  (Chip Enable) L-Pegel erhalten.

Über den Anschluß DIR (direction, Richtung) wird mit dem Signal  $\overline{\text{IOW}}$  die Arbeitsrichtung des Datenbus-Treibers gesteuert. Hierdurch wird festgelegt, ob Daten vom Prozessor zum Baustein 8251 und von dort über den Anschluß TxD seriell ausgesendet werden, oder ob der Prozessor die über den Anschluß RxD seriell empfangenen Daten liest.

Die Freigabe des Datenbus-Treibers und des Schnittstellenbausteins erfolgt durch den Adreßvergleich. Dies geschieht nur, wenn die Adresse auf den Adreßleitungen A4 bis A7 mit derjenigen Baugruppen-Nummer übereinstimmt, die mit den Schaltern S1 bis S4 eingestellt ist. Zur Freigabe des Adreßvergleichers muß eine der Steuerleitungen  $\overline{\text{IOW}}$  oder  $\overline{\text{IOR}}$  aktiv sein.

Mit der Adreßleitung A0 werden bei H-Pegel die Steuer-Register und bei L-Pegel die Datenregister (Sender- und Empfänger-Register) des 8251 adressiert.

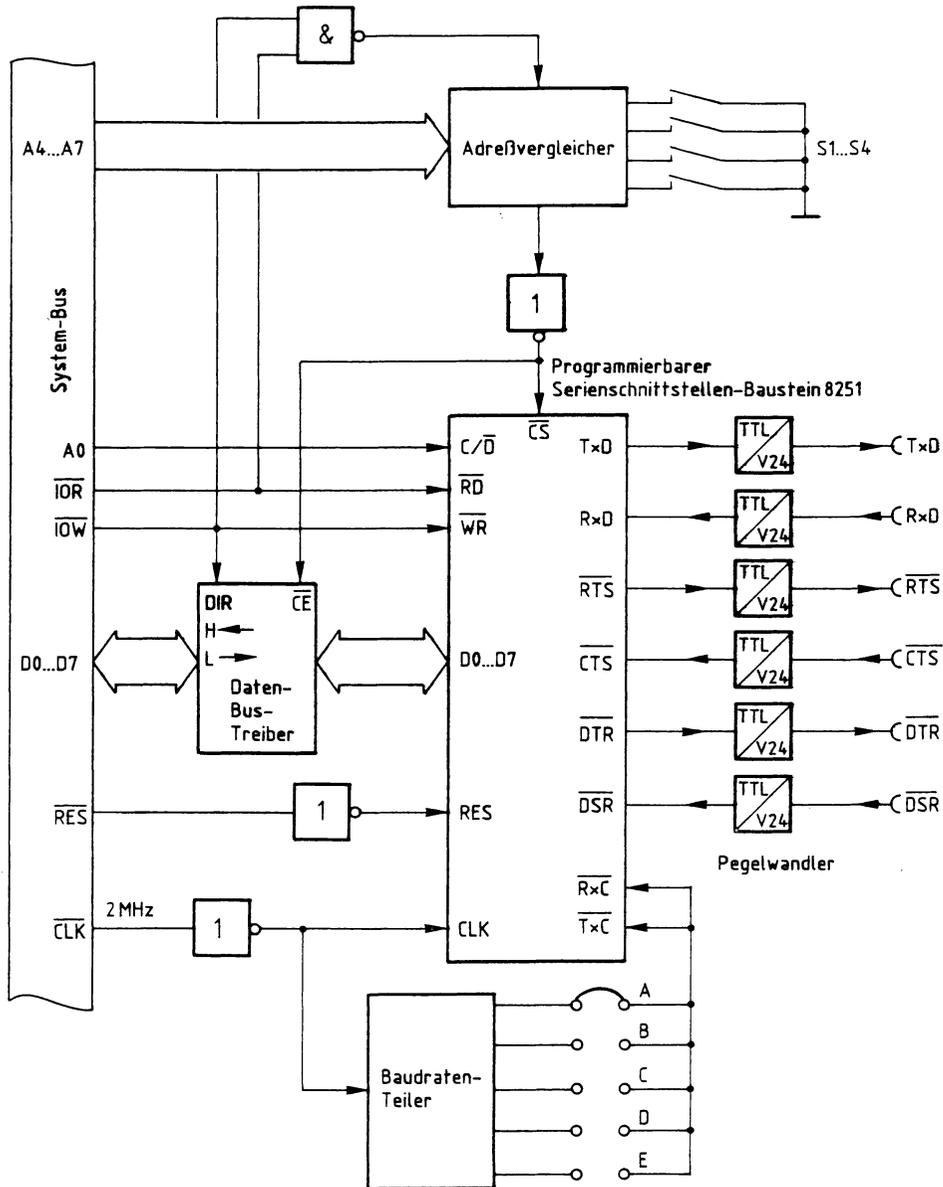


Bild 74: Der Ausbau der seriellen Schnittstelle zur V.24-Schnittstelle; es sind Baudraten bis zu 9600 möglich (Stromlaufplan: Seite 450).

Die nachfolgende Funktionsbeschreibung der V.24-Schnittstelle bezieht sich nur auf das Blockschaltbild (Bild 74). Einzelheiten des verwendeten Schnittstellenbausteins 8251 wurden zuvor behandelt (ab Seite 405).

Die serielle Daten-Sendeleitung TxD, die serielle Daten-Empfangsleitung RxD sowie einige Steuerleitungen des Schnittstellenbausteins 8251 sind über TTL/V24-Pegelwandler mit einer 25poligen Buchsenleiste verbunden.

Für interne Steuerzwecke erhält der Baustein am Anschluß CLK den von einem Inverter aufgefrischten 2-MHz-Systemtakt. Aus dem System-Takt wird außerdem im Baudraten-Teiler der zur parallel/seriell- und seriell/parallel-Wandlung erforderliche Baudraten-Takt gewonnen und dem Baustein an den Anschlüssen  $\overline{\text{RxC}}$  und  $\overline{\text{TxC}}$  zugeführt. Durch Brücken und geeignete Initialisierung können 7 verschiedene Baudraten eingestellt werden (siehe Tabelle im Stromlaufplan, Seite 450).

Der Daten-Bus ist über einen bidirektionalen Datenbus-Treiber in Tristate-Technik mit den Datenleitungen des Schnittstellen-Bausteins verbunden. Der Datenbus-Treiber trennt im Ruhezustand den Datenweg und gewährleistet im aktiven Zustand einen ausreichend hohen Treiberstrom in Richtung System-Bus. Zur Aktivierung des Datenbus-Treibers muß der Anschluß  $\overline{\text{CE}}$  (Chip Enable) L-Pegel erhalten.

Über den Anschluß DIR (direction, Richtung) wird mit dem Signal  $\overline{\text{IOW}}$  die Arbeitsrichtung des Datenbus-Treibers gesteuert. Hierdurch wird festgelegt, ob Daten vom Prozessor zum Baustein 8251 und von dort über den Anschluß TxD seriell ausgesendet werden, oder ob der Prozessor die über den Anschluß RxD seriell empfangenen Daten liest.

Die Freigabe des Datenbus-Treibers und des Schnittstellenbausteins erfolgt durch den Adreßvergleich. Dies geschieht nur, wenn die Adresse auf den Adreßleitungen A4 bis A7 mit derjenigen Baugruppen-Nummer übereinstimmt, die mit den Schaltern S1 bis S4 eingestellt ist. Zur Freigabe des Adreßvergleichers muß eine der Steuerleitungen  $\overline{\text{IOW}}$  oder  $\overline{\text{IOR}}$  aktiv sein.

Mit der Adreßleitung A0 werden bei H-Pegel die Steuer-Register und bei L-Pegel die Datenregister (Sender- und Empfänger-Register) des 8251 adressiert.

Die Verbindungen vom Schnittstellenbaustein zur 25poligen Buchsenleiste (siehe Bild 75):

**TxD (Transmitter Data, Senderdaten):**

Serieller Datenausgang des Bausteins, durch Brücke 2 mit Anschlußstift 2 der 25poligen Buchsenleiste verbunden.

**RxD (Receiver Data, Empfängerdaten):**

Serieller Dateneingang des Bausteins, durch Brücke 4 mit Anschlußstift 3 der 25poligen Buchsenleiste verbunden.

**$\overline{\text{RTS}}$  (Request To Send, Sendeteil der Peripherie einschalten):**

Der Signalzustand dieses beliebig verwendbaren Bausteinausgangs wird bei der Initialisierung durch das Kommandowort festgelegt und kann danach auch wieder per Programm verändert werden. In der Grundverdrahtung der V24/RS232-Schnittstelle wird jedoch nicht das Bausteinsignal  $\overline{\text{RTS}}$ , sondern Signal RxRDY auf den  $\overline{\text{RTS}}$ -Anschluß der 25poligen Buchsenleiste geführt (Anschlußstift 4).

**RxRDY (Receiver Ready, Empfänger hat ein Zeichen empfangen / ist besetzt):**

Ausgang, der beim Vorliegen eines Zeichens im Empfängerregister des Schnittstellenbausteins H-Pegel annimmt. In der Grundverdrahtung der Schnittstelle wird RxRDY über die Brücke 16, den Pegelwandler IC7.2 und die Brücke 6 zum Anschlußstift  $\overline{\text{RTS}}$  der 25poligen Buchsenleiste geführt. Hiermit kann der externe Datensender so lange gestoppt werden, bis das Empfängerregister des Bausteins von der CPU gelesen (»geleert«) wurde und damit wieder bereit ist, ein neues Zeichen aufzunehmen.

**$\overline{\text{CTS}}$  (Clear To Send, Senderfreigabe):**

L-Signal an diesem Bausteineingang gibt den eigenen Datensender frei, H-Signal sperrt ihn. Über den Pegelwandler IC8.2 und die Brücke 8 ist der  $\overline{\text{CTS}}$ -Anschluß des Bausteins mit Anschlußstift 5 der 25poligen Buchse verbunden. Hier ist zur Senderfreigabe 0-Signal (+3 V bis +12 V) erforderlich. Bei 1-Signal (-3 V bis -12 V) oder offenem Eingang ist der Datensender gesperrt. Wird anstelle der Brücke 8 die Brücke 9 geschlossen, so erfolgt die Senderfreigabe nicht über die Peripherie, sondern über R5 (Datensender ist ständig freigegeben).

$\overline{\text{DTR}}$  (Data Terminal Ready, Betriebsbereitschaft des eigenen Gerätes):

Der Signalzustand dieses beliebig verwendbaren Ausgangs wird bei der Initialisierung des Bausteins durch das Kommandowort festgelegt und kann danach auch wieder per Programm verändert werden. Mit L-Pegel am Bausteinanschluß  $\overline{\text{DTR}}$  kann einem Peripheriegerät mitgeteilt werden, daß die Serienschnittstelle betriebsbereit ist. Durch den Pegelwandler IC7.3 führt der  $\overline{\text{DTR}}$ -Anschluß der 25poligen Buchsenleiste dann 0-Signal (+3 V bis +12 V).

$\overline{\text{DSR}}$  (Data Set Ready, Betriebsbereitschaft eines anderen Gerätes):

Der Signalzustand dieses beliebig verwendbaren Eingangs kann über das Statuswort des Schnittstellenbausteins abgefragt werden. Mit Hilfe des Anschlusses  $\overline{\text{DSR}}$  ist der Prozessor hierdurch z. B. in der Lage, die Betriebsbereitschaft eines Peripheriegerätes festzustellen. Hierzu muß dieses Gerät 0-Signal (+3 V bis +12 V) auf die  $\overline{\text{DSR}}$ -Leitung legen.

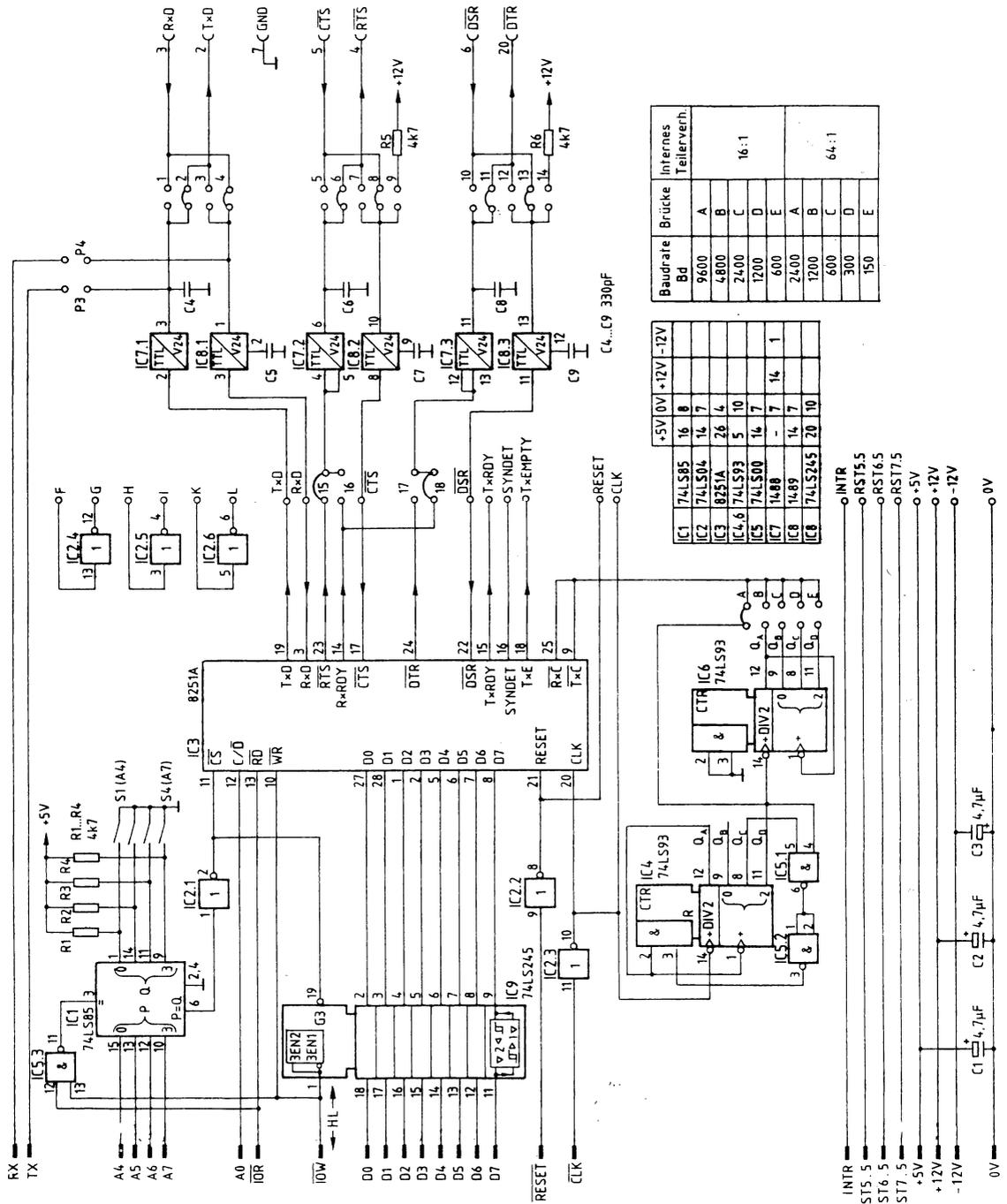


Bild 75: Der Stromlaufplan der V.24-Schnittstelle.

In diesem Kapitel wird eine Baugruppe beschrieben, mit der auf 2 Kanälen unabhängig voneinander je eine DA- oder AD-Wandlung durchgeführt werden kann.

Fast alle elektronisch gesteuerten Geräte und Anlagen sind mit äußeren Schaltungsteilen verbunden, welche Eingabesignale an die Steuerung abgeben, aber auch Ausgabesignale von der Steuerung erhalten. Durch dieses Zusammenwirken der Steuereinheit mit den Eingabe- und Ausgabe-Signalen wird der Zustand einer Anlage überwacht und beeinflusst. Eingabesignale stammen beispielsweise von Endschaltern, Niveaufühlern und Thermostaten. Hierbei liegen vielfach bereits digitale Informationen vor (»EIN«- und »AUS«-Zustände), deren Erfassung, Auswertung und Weiterverarbeitung in digitalen und computergestützten Einrichtungen direkt möglich ist. Auch digitale Werte z. B. zum Ansteuern von Lampen, Relais, Magnetventilen und Schützen können direkt ausgegeben werden.

Neben digitalen Signalen müssen häufig auch analoge Werte überwacht und verarbeitet werden. Beispiele hierfür sind Temperaturmessungen mit Thermoelementen, Drehzahlerfassungen mit Tachogeneratoren und Drehmomentmessungen mit Dehnungsmeßstreifen. Obwohl es zum Teil auch hierfür bereits digital arbeitende Meßfühler gibt, verwendet man aus Kostengründen und wegen der Kompatibilität zu bereits bestehenden Anlagenteilen dennoch oft analoge Meßfühler, die häufig einen Ausgangspegel von 0 V bis 10 V besitzen. Dieser Signalbereich muß daher auch von digital arbeitenden Steuereinrichtungen verarbeitet werden können. Hierzu wird der analoge Spannungsbereich von 0 V bis 10 V in Digitalwerte umgesetzt, wozu Analog/Digital-Wandler (AD-Wandler) dienen. Sollen Analogwerte von der Steuerung ausgegeben werden, so werden die Digitalwerte mit Hilfe von Digital/Analog-Wandlern (DA-Wandler) umgesetzt.

Dieses Kapitel beschreibt Aufbau, Funktion und Einsatz einer Baugruppe, mit der sich DA- und AD-Wandlungen durchführen lassen. Dabei wird eine in der Mikrocomputertechnik übliche Schaltungstechnik angewendet. Die Baugruppe besitzt zwei 8-Bit-DA-Wandler.

Ein AD-Wandler ist nicht vorhanden; daher wird auf die direkte AD-Wandlung nicht eingegangen. Ausführlich beschrieben wird jedoch die AD-Wandlung mit Hilfe eines DA-Wandlers sowie das hierzu erforderliche Steuerprogramm.