MUX-KE

MULTITERMINAL-CONTROLLER
Technische Beschreibung

Version 5/83
D 920.221

Dieses Handbuch beschreibt die Funktionen des MUX-KE und gibt Hinweise bei der Installation der Baugruppe in Systeme.

Best.-Nr.: D 920.221
Version : 1.0
Datum : Mai 1983

MUX-KE-Multiterminal-Controller-Einheit

1. MUX-KE-Ubersicht

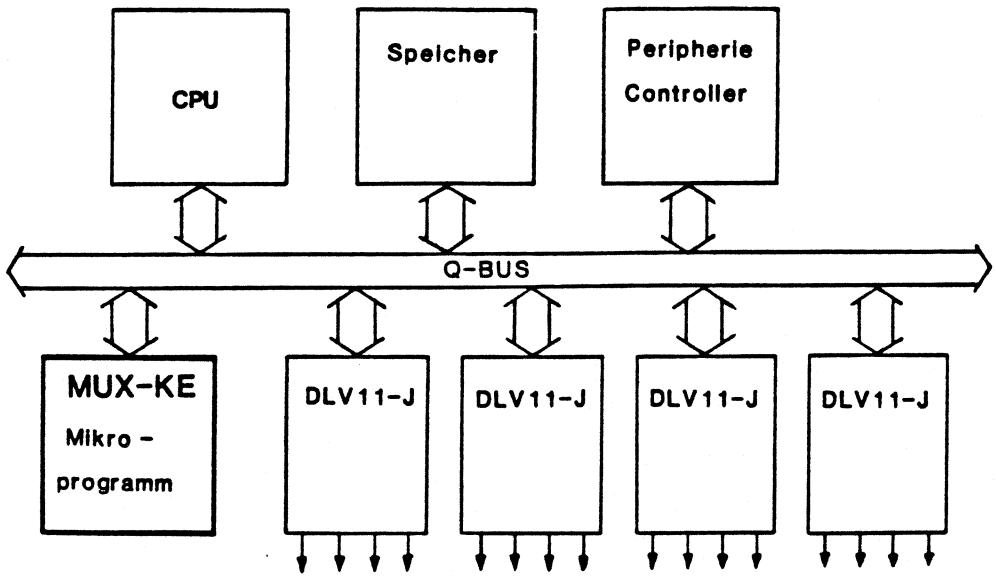
1.1 Emulation des DMA-fähigen Multiplexers DH11 für den DEC-Q-BUS

Das MUX-KE emuliert das UNIBUS-Device/DH11-Multiplexer auf dem DEC-Q-BUS. Damit ist der Anschluss von 16 asynchronen seriellen Kommunikationskanaelen an Q-BUS-Systeme moeglich. Das Zeichenformat und die Uebertragungsgeschwindigkeit sind programmierbar. Als wesentlicher Vorteil gegenueber Standard-E/A-Kanaelen bietet das MUX-KE die Bufferung der Eingabedaten und den DMA-gesteuerten Datentransfer bei der Ausgabe. Diese Moeglichkeit bietet eine beträchtliche Entlastung der CPU von den üblicherweise interruptgesteuerten Ein/Ausgabe- Operationen. Bei der Eingabe ist dadurch ein Zeichenverlust weitgehend ausgeschlossen und bei der Ausgabe ein wesentlicher Geschwindigkeitsgewinn zu verzeichnen.

1.2 Einsatz

Die MUX-KE-Einheit ist besonders dann geeignet, wenn ein bestehendes Q-BUS- System mit mehreren Ein/Ausgabe-Kanälen über Standard E/A-Module an die Grenzen der Übertragungsgeschwindigkeit gelangt. In diesem Fall ist es moeglich, durch die zusätzliche Bestückung des Systems mit dem MUX-KE- Modul (Dual-Slot) diesen Engpass zu beseitigen. Ausser dem zusaetzlichen Modul ist keinerlei HW-Aenderung am System notwendig (vorhandene Standard-E/A-Karten werden weiterhin verwendet; keine neuen Verbindungskabel notwendig; keine neue Steckerplatine notwendig). Softwaremässig wird der Standard-Treiber DH11 verwendet. Eine Modifikation ist nicht notwendig. Eine moegliche Konfiguration ist dem Blockschaltbild zu entnehmen.

Blockschaltbild für System mit MUX-KE



1.3 Spezifikation

- Die Emulation geht davon aus, keine Anderung an der bestehenden Hardware und Software vorzunehmen.
- In das bestehende System wird nur der CP-Modul mit Mikroprogramm eingesteckt. Die vorhandenen DLV11-Module koennen weiter verwendet werden.
- softwaremässig wird lediglich der DH11-Treiber in das Betriebssystem eingebunden.
- Zeichlänge: 5,6,7 oder 8 Bit.
- Zahl der Stop-Bits:
  1 oder 2 für 6,7,8-bit Zeichen
  1 oder 1,5 für 5-bit Zeichen.
- Parität: gerade, ungerade oder keine.
- Operationsmodus: 50 bis 19200 baud,
  externer Takt, 0 entspricht Abschaltung
- Break-Erkennung und Generierung.
- Hardware-Echo
- Getrennte Einstellung für jeden Kanal.
- 64 Zeichen HW-Buffer für empfangene Zeichen
- Direkter Speicherzugriff beim Datentransfer in Ausgaberichtung.
- 5x Dual-Slot-Boards für max. Ausbau
- Stromversorgung MUX-KE + 4xDLV11-J 5V/7A, 12V/1A

Zusätzliche Vorteile gegenüber orginal DH11-Modul:

-Abholung der Zeichen aus Buffer auch zeitgesteuert moeglich.

-Baudraten groesser 9600 bearbeitbar
(Abschaltung nicht verwendeter Kanäle).

1.4 Kurzbeschreibung Funktion

Bei der Ansprache des DH11 wickelt der CP zusammen mit den DLV11-Modulen den Datentransfer ab. Zu diesem Zweck fragt der CP die Statusregister der DLV11-Moduln ab.

Bei der Eingabe wird das empfangene Zeichen vom Datenregister des DLV11 in das SILO im CP übertragen. Von dort koennen die Zeichen dann vom Treiber abgeholt werden.

Bei Ausgabe übergibt der Treiber die Blockadresse und Anzahl der Ausgabedaten. Der CP holt daraufhin die Daten per DMA und transferiert sie zeichenweise in das Ausgaberegister der jeweiligen DLV11.

2. Terminaleingabe/-ausgabe bei Q-BUS-systemen

2.1 Terminaleingabe/-ausgabe über ungepufferte serielle Kanäle

Alle Terminaleingaben bzw. Ausgaben in Q-BUS-Systemen werden zeichenweise durchgeführt. Da die Zeichen in beiden Uebertragungsrichtungen (vom Terminal,zum Terminal) interruptgesteuert behandelt werden, ist für jedes Zeichen der Einsprung in ein Interruptprogramm und dessen Abarbeitung notwendig. Die für die Interruptbehandlung benoetigte CPU Zeit steht somit für andere Aufgaben nicht mehr zur Verfügung. Bei einer groesseren Anzahl von Terminals kann es sogar soweit kommen, dass die CPU allein durch die Eingabe-/Ausgabe-Operationen überlastet wird und im ungünstigsten Fall in der Eingaberichtung Zeichen verloren gehen. Waehrend in der Eingaberichtung sich durch diesen Engpass Fehlverhalten ergeben kann, wird in der Ausgaberichtung lediglich der Bedienkomfort herabgesetzt, indem die Zeichen nur verzoegert auf dem Terminal erscheinen. Beide Fälle führen zu einer erheblichen Leistungseinbusse des Systems.

2.2 Entlastung der CPU von Terminal-Eingabe-/Ausgabeoperationen

Der beschriebene Engpass, der bei allen Q-BUS-Systemen bei entsprechendem Ausbau zum Problem werden kann, wird z.B. bei UNIBUS-Systemen durch Verwendung eines Terminal-Multiplexers behoben.

Ohne das Konzept der unterbrechungsgesteuerten Ein- und Ausgabe aufzugeben, wird mit Hilfe des Multiplexers eine blockweise Ausgabe betrieben. Erst nach Beendigung einer Blockübertragung - die Länge des Blockes ist variabel - wird eine Programmunterbrechung erzeugt. Diese veranlasst im Rechner die Bereitstellung eines moeglicherweise weiteren auszugebenden Datenblockes. Die Zeichen des Blockes werden, nachdem die notwendigen Kommandos an den Multiplexer gegeben worden sind, durch direkte Speichzugriffe (DMA) abgeholt und ohne weiteres Zutun der CPU an die Schnittstelle gesendet. Da jetzt in Ausgaberichtung nur sehr wenig zeitraubende Unterbrechungsbehandlungen anfallen und zudem auf die Zeichen im "cycle-stealing"-Verfahren zugegriffen wird, bleibt die CPU-Belastung in diesem Fall minimal.

Auf der Eingabeseite bietet dieser Multiplexer zwar keine DMA-Unterstützung, so doch einen grossen "First-In-First-Out"-Puffer, in dem die von den 16 moeglichen Terminals ankommenden Zeichen zwischengepuffert werden. Damit wird das Risiko eines Zeichverlustes bei der Eingabe auf ein Minimum reduziert. Der Puffer entschärft die zeitlichen Anforderungen bei Eingabeoperationen und die SW hat in der Regel genügend Zeit die Zeichen aus dem Puffer abzuholen. Weiterhin besteht auch die Moeglichkeit nicht für jedes Zeichen das Interruptprogramm neu zu starten. Es ist moeglich mit einer Unterbrechung sofort mehrere Zeichen vom Multiplexer einzulesen. Diese Eigenschaft ist insbesondere dann nützlich, wenn die serielle Schnittstelle für asynchrone Datenübertragung verwendet wird.

Die Verwendung eines Terminal-Multiplexer entlastet also die CPU stark von Ein-/Ausgabe-Operationen und führt zu einer erheblichen Leistungssteigerung des Systems. Durch das MUX-KE zusammen mit den DLV11-J-Modulen wird ein solcher Multiplexer auch auf dem Q-BUS zur Verfügung gestellt. Das MUX-KE emuliert dabei das UNIBUS-Device DH11 voll, so dass die vorhandene Treiber-Software auch füer Q-BUS-Systeme übernommen werden kann.

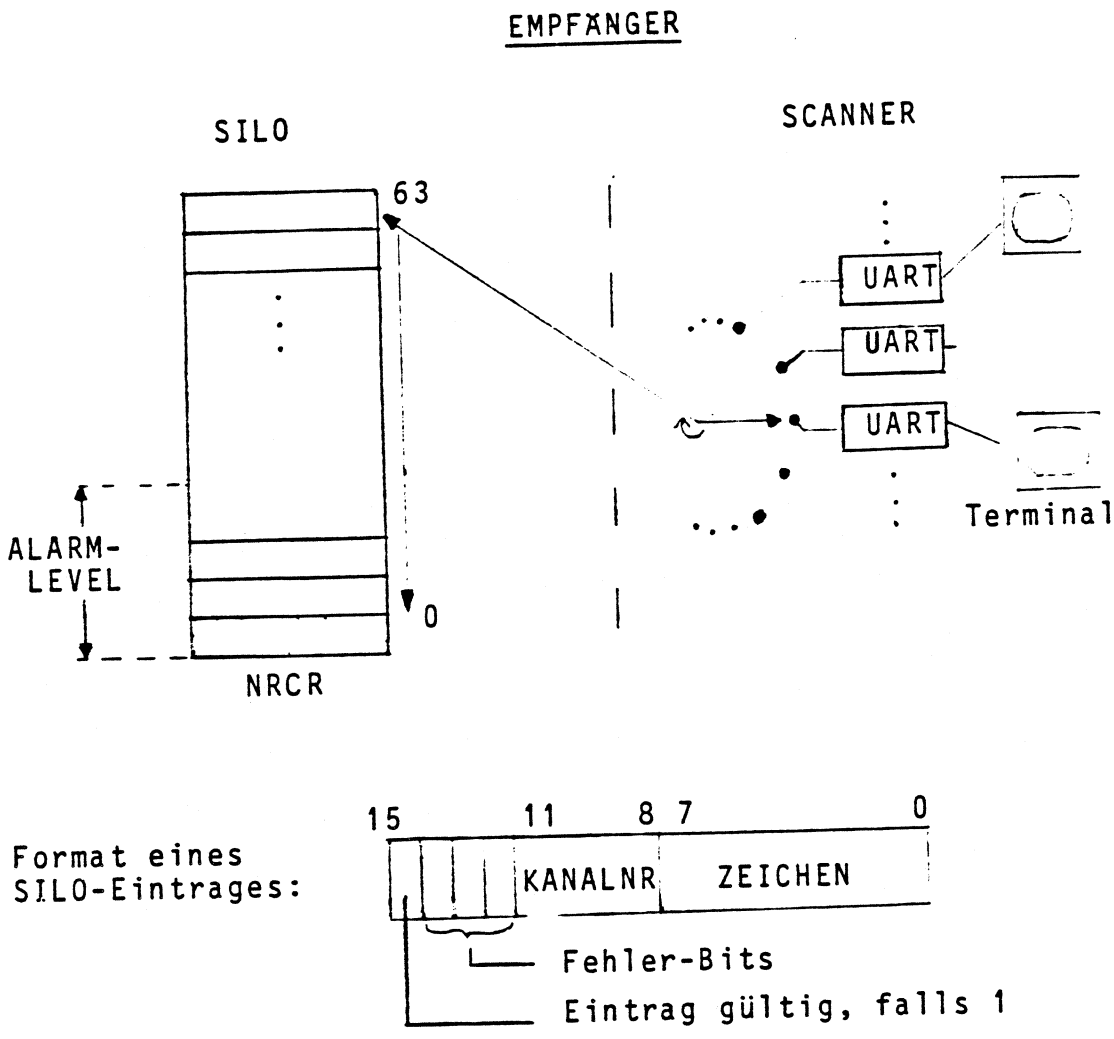
3. Funktionsbeschreibung des Terminal-Multiplexers

3.1 Empfänger Operation

Bild 2 stellt systematisch den Empfängerteil des Multiplexers dar. Zu sehen sind zwei Funktionseinheiten:
der Abtastmechanismus (Scanner) sowie das SILO als Puffer.

Der Abtastteil bedient 16 UART-Module, die die bekannten Funktionen wie Paritätsprüfung, Start-bzw. Stop-Bit Erkennung usw. aufweisen. Nachdem ein Zeichen über das Schieberegister eingelesen und parallel in das Halteregister übertragen worden ist, kann es durch den Abtastmechanismus ausgelesen werden. Bevor es in das SILO eingetragen wird, müssen Identifikation (Kanalnummer) und Gültigkeitsanzeige (Bit 15 eines Silo Eintrages) noch hinzukommen. Ist diese Operation beendet, wird zyklisch zum nächsten Kanal übergegangen.

Bild 2 Empfänger des DH11-Multiplexers



3.2 Silo-Operation

Das Silo ist ein "First-In-First-Out"-Puffer. Ein empfangenes Zeichen wird an der obersten Stelle des 16 Bit breiten Silos parallel eingetragen. Dann rückt es automatisch abwärts bis zu der ersten Stelle, die noch kein Zeichen enthält. Ist das Silo leer, "wandert" es direkt in das Next Received Character Register, welches der unterste Platz des Silos ist. Ein Interrupt wird dann abgesetzt, wenn der Silo Alarm Wert (Silo Alarm Level) überschritten und mindestens ein Zeichen bereits am "Boden" des Silos angekommen ist.

Bei der Emulation des Silos mit den Mitteln der Mikroprogrammierung wird die oben erwähnte Pufferorganisation durch einen Ringpuffer nachgebildet. Der Ringpuffer wird dabei über zwei Zeiger verwaltet. Der eine (GETPTR) verweist auf die Stelle im Speicher, in der sich das nächste auszulesende Zeichen befindet. Der andere (PUTPTR) auf diejenige, in die das nächste gelesene Zeichen einzutragen ist.

Das "Next-Received-Character-Register" (NRCR) ist ein read-only-register. Ein Auslesen des NRCR veranlasst, dass alle weiteren Zeichen im Silo eine Position nach unten geschoben werden, so dass beim nächsten Lesezyklus bereits ein anderes Zeichen an unterster Stelle des Silos ist. Ob ein Eintrag gültig ist, das heisst, ob sich tatsächlich ein von einem Kanal gelesenes Zeichen im NRCR befindet, muss vom Abwender durch Ueberprüfung des Bit 15 im NRCR festgestellt werden. Stoesst man auf ein ungültiges Zeichen, ist die Folge der Lesezyklen zu beenden und auf eine erneute Unterbrechung zu warten.

3.3 Sender Operation

Jedem Kanal sind zwei Register zugeordnet, das Byte-Count-Register und das Current Address Register, mit deren Hilfe eine Blockuebertragung kontrolliert wird. Die beiden Register sind über das System-Control- Register auswählbar (siehe Bild 3) und koennen dann durch programmierte Zugriffe gelesen oder beschrieben werden. Intern benutzt die DMA-Logik das Adressregister für die Adressierung der auszugebenden Zeichen eines Datenblockes im LSI-11 Hauptspeicher. Der Abtastmechanismus des Senders sorgt für die richtige Zuordnung der Zeichen zu den Kanälen. Gestartet wird eine DMA-Blockübertragung, nachdem das entsprechende Bit des Buffer-Active-Registers (BAR) gesetzt worden ist. Die Beendigung der Uebertragung hat schliesslich das Loeschen dieses Bits zur Folge. Damit ist es für ein Anwenderprogramm auch moeglich, unter Benutzung des BAR, eine statusgesteuerte Ubertragung zu realisieren.

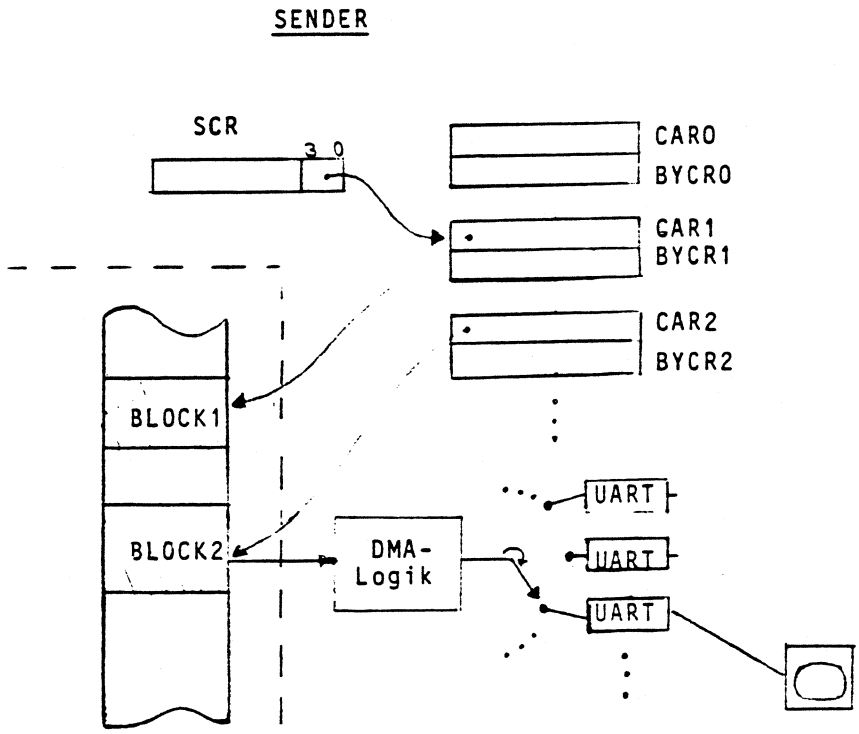


Bild 3 Sender des DH11-Multiplexers

3.4 Die Auto-Echo-Operation

Die Auto-Echo-Einrichtung bietet die Moeglichkeit das empfangene Zeichen ohne Treibereingriff zurückzuschicken. Dieser Betriebsmodus wird durch entsprechende Belegung der Systeme Control Register und Line Parameter Register eingestellt. Bei der Aufo-Echo-Operation sind folgende Fälle zu unterscheiden:

1. Wird ein Zeichen von einem Kanal gelesen, dessen Auto-Echo Modus nicht eingeschaltet ist, so wird es - wie oben beschrieben - in das Silo eingetragen.
2. Ist der Auto-Echo-Modus eingeschaltet und wird ein Zeichen mit einer Fehlerbedingung gelesen, ist folgendes zu prüfen:

   -Bei einem Framing-Error oder Overrun-Error wird das Zeichen ins Silo eingetragen, aber nicht zurückgesendet. Diese Fehler aktivieren die BREAK-Bedingung, die im Rechner zu verschiedenen Reaktionen fuehren kann.

   -Bei anderen Fehlern wird das Zeichen zurückgesendet und ins Silo eingetragen.
3. Der Auto-Echo-Modus ist eingeschaltet und es ist ein Zeichen fehlerlos gelesen worden.

   -Ist das Senderegister des UART frei, kann das Zeichen sofort zurückgeschickt werden; anderenfalls ist der Versuch, es zurückzusenden, bei der nächsten Abtastung des Kanals zu wiederholen.

   -Ist zwischenzeitlich ein weiteres Zeichen angekommen, das ein Echo erfahren soll, geschieht ein Zeichenüberlauf, wenn das vorherige noch nicht zurückgesendet werden konnte.

Die Eintragung ins Silo findet bei einem fehlerfrei gelesenen Zeichen immer statt.

3.5 Interrupts vom Terminal-Multiplexer

Der Multiplexer kennt vier Arten von Interrupts, wobei jeweils zwei der Empfänger-bzw. der Senderseite zugeordnet sind. Im folgenden sind die verschiedenen Unterbrechungsbedingungen aufgefuehrt:

Empfänger-Interrupts

1.) Die aktuelle Anzahl der Zeichen im Silo uebersteigt den im Silo Status Register angegebenen Silo-Alarm-Wert.
(Receiver Interrupt)

2.) Das Silo ist voll und ein weiteres Zeichen solllte in das Silo eingetragen werden. (Storage Overflow Interrupt)

Sender-Interrupts

1.) Die Blockuebertragung eines Kanals (oder auch mehrerer Kanäle) ist beendet. Das ist dann der Fall, wenn das letzte Zeichen eines Blockes in das Pufferregister des entsprechenden UART's gesendet worden ist.
(Transmitter Interrupt)

2.) Beim DMA-Zyklus wird eine nicht gueltige Adresse angesprochen. Dies ist dadurch festzustellen, dass die Antwort des adressierten Slave-Moduls nicht innerhalb von ca. 20 usek eintrifft.) (Non Existent Memory Interrupt)

Den Interruptbits im System Control Register (insgesamt also 4) sind entsprechende "Interrupt-Enable"-Bits zugeordnet. Dabei ist zu beachten, dass fuer die beiden Senderunterbrechungen nur ein gemeinsames "Enable"-Bit existiert.

4. Realisierung des Terminal-Multiusers mit MUX-KE und DLV11-J-Modulen

Wie oben erwähnt wird das UNIBUS-Device DH11 durch einen mikroprogrammierbaren Bit-Slice-Processor (MUX-KE) und Standard-DLV11-J Module nachgebildet. Im folgenden wird auf diese Module und das Mikroprogramm fuer die Emulation eingegangen.

4.1 Hardwareaufbau des Terminal-Multiusers

Bei der Realisierung des Terminal-Multiplexers wurde grosser Wert auf die Verwendung von Standard-Modulen gelegt. Ausserdem sollte keinerlei Aenderung an den bestehenden System-HW und System-SW notwendig sein. Erreicht wurde dies durch die Verwendung des mikroprogrammierbaren Bit-Slice-Prozessors (Kommunikations-Prozessors CP) und den seriellen Schnittstellenmodulen DLV11-J. Der DLV11-J Modul stellt vier Kanäle zur Verfügung, so dass bei der Verwendung von 4 Modulen ein Multiplexer mit 16 Kanälen entsteht. Die Kontrolle über die 4 Module hat der Bit-Slice Prozessor(MUX-KE).

Der Kommunikations-Prozessor CP ist ein Standardmodul, der durch ein entsprechendes Mikroprogramm den entsprechenden Anforderungen angepasst werden kann.

Man kann diesen Modul als Zusatzprozessor auf dem Q-BUS betrachten, der festgelegt durch das Mikroprogramm (ROMS) starr eine Aufgabe ausführt. Die Geschwindigkeit dieses Prozessors ist verglichen z.B. zur LSI11 hoch. In einem Befehlszyklus von 200 nsec werden eine arithmetische/ logische Operation, eine Schiebeopoeration und ein Sprungbefehl ausgeführt. Der CP hat einen Programmspeicher von 4 KByte wobei die einzelnen Befehle 64bit breit sind. Das lokale RAM hat die Groesse von 256 Byte. Die Schnittstellen nach aussen bilden zum einen der Q-BUS-Anschluss und zum anderen der sehr einfach gehaltene MIKI-BUS. Der MIKI-Bus wird in der Anwendung MUX-KE nicht verwendet. Der Kommunikations-Prozessor ermoeglicht die Abarbeitung aller Q-Bus-Protokolle wie programmierten Datentransfer (Wort/Byte), DMA und Interrupt. Die Kommunikation erfolgt über maximal 256 Bytes in der I/O-page des Q-Bus-Systems. (siehe auch Ke-Handbuch).

Die DLV11-J-Module bieten 4 asynchrone, serielle Schnittstellen mit V24-Pegel an. Die 4 Kanäle koennen unabhängig voneinander konfiguriert werden bezüglich Baudrate und Zeichenformat.

Der Modul führt die Seriell/Parallel-und Parallel/Seriell-Datenkonversionen mit einem universellen asynchronen Empfänger/Sender (UART) pro Kanal durch. Der UART enthaelt alle Sender-und Empfaengerfunktionen. Der Empfaenger führt die Seriell/Parallel-Konversion von 7-oder 8-Bit-Codes durch ,die Zeichen erschein mit Datenpuffer rechtsbündig ohne Start-, Stop-und Paritätsbits. Der Sender führt die Parallel/Seriell-Konversion der Daten,die vom Q-Bus kommen, aus und versieht diese mit Start-, Stop- und Paritätsbits.

Die Kommunikation CPU und DLV11-J-Modul erfolgt über 16 Register, wobei jeweils 4 einem Kanal zugeordnet sind. Bei den Registern handelt es sich um

- Receive Control/Status Register
- Receive-Buffer
- Transmit Control/Status Register
- Transmit-Buffer

(siehe auch DEC LSI-Interface-Handbook)
Mit diesen Komponenten wird nun der Terminal Multiplexer realisiert, so dass die Schnittstellen zu den Terminals durch die DLV11-J Module und die Schnittstelle zur CPU über den Q-Bus durch den Kommunikations- Prozessor CP (MUX-KE) gebildet wird. Die Q-Bus wird allerdings auch für Kommunikation zwischen MUX-KE und DLV11-J-Boards verwendet. Bei der Eingabe wird das empfangene Zeichen vom Datenregister der DLV11 in das SILO im CP übertragen. Von dort koennen dann die Zeichen abgeholt werden. Bei der Ausgabe übergibt der Treiber die Blockadresse und Anzahl der Ausgabedaten. Der CP holt daraufhin die Daten per DMA und transferiert sie zeichenweise in das Ausgaberegister des jeweiligen Kanals. Dies bedeutet eine zusätzliche Q-Bus-Belastung, die jedoch wie später beschrieben, relativ gering ist. Ein Problem bei der Verwendung dieser Standardmodulen bildet die Notwendigkeit den Empfang von Zeichen festzustellen. Die DLV11-Module melden das normalerweise über einen Interrupt zur CPU, wenn das READY-Bit im Receive Control/Status- Register gesetzt ist. Der Interrupt wird aber in diesem Fall nicht verwendet, da der CPU ja von diesen Aufgaben entlastet werden soll. Bei der realisierten Loesung wird nicht versucht, auf eine Aktivierung des "Ready"-Signals so schnell wie moeglich zu reagieren. Vielmehr werden jetzt die Status-Bits in den Status/Control-Registern der Kanäle zyklisch abgefragt. Durch die damit notwendig gewordenen DMA-Zyklen ergibt sich eine Grundbelastung des Q-Busses, die zur Abfragehäuffigkeit proportional ist. Stoesst man bei dieser "Polling"-Betriebsweise auf einen Kanal mit gesetzten Ready-Bit, schliesst sich unmittelbar ein DMA-Lesezugriff an, um das Datum aus dem Empfangspuffer zu entnehmen. Dieses Verfahren funktioniert, solange man garantieren kann, dass die Intervalle der Abfragezeitpunkte an dem selben Kanal kürzer sind, als die Zeit, die zwischen zwei aufeinander folgenden Zeichen vergeht. Massgebend ist dabei die Baudrate der Schnittstelle, sowie die Anzahl der Bits eines für die Uebertragung aufbereiteten Zeichens (Anzahl Datenbits, Stop-Bit usw.) Da ein Kanal mit einer eingestellten Baudrate von 9600 Baud etwa jede Millisekunde abgetastet werden muss, liegt es nah, -im Interesse einer Minimierung der Q-Bus Belastung- eine Zeitrasterung für die Abfragezeitpunkte einzuführen. Das heisst, jeder Kanal soll nur so oft abgefragt werden, wie es einer Baudrate entspricht. Dadurch wird verhindert, dass das Statusregister des Kanals "unnoetig" oft gelesen wird. Durchgeführte Messungen am laufenden System haben gezeigt, dass sich dadurch die Anzahl der MUX-KE internen DMA -Zyklen auf 1/6 bis 1/7 reduzieren lässt.

4.2 Strategie zur Minimierung der Q-Bus Belastung

Vorausgesetzt sind im folgenden 16 zu bedienende, asynchrone, serielle vollduplex betriebene Kanäle, die alle mit 9600 Baud arbeiten. Diese Schrittgeschwindigkeit erweist sich in der Praxis in den meisten Faellen als ausreichend und ist hier Richtwert für das Optimierungskonzept.

Bei einer Baudrate von 9600 Baud berechnet sich der kürzeste Abstand DMIN zweier aufeinanderfolgender Zeichen zu

DMIN = 1/1066 sek = 0.938 msek.

Dabei sei angenommen, dass 9 Bit je Zeichen übertragen werden.

Der oben berechnete Wert bestimmt die Rahmenzeit, die Zeit, die zwischen zwei Bearbeitungspunkten desselben Kanals hoechstens vergehen darf. Die Bedienzeit für einen einzelnen Kanal ist dann

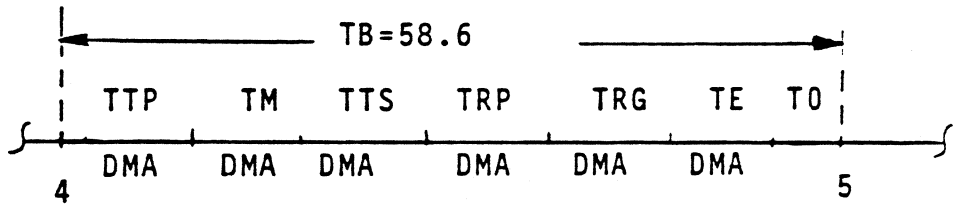
TB = DMIN / 16 = 58.625 usek.

In dieser Zeit müssen alle für die Behandlung eines Kanals notwendigen Taetigkeiten verrichtet werden koennen.

Eine andere Ueberlegung geht von den Bedingungen im realen Benutzerbetrieb aus. Wohl sehr selten werden an ein Q-Bus System 16 Terminals angeschlossen, die mit 9600 Baud betrieben werden und zudem alle zur gleichen Zeit sowohl in Eingabe- als auch in Ausgaberichtung mit Vollast arbeiten. Im Normalfall wird es ein Benutzerbetrieb sein mit 4-6 angeschlossenen Bildschirmen; die ein oder andere Schnittstelle wird fuer Uebertragungszwecke reserviert sein. Daher ist der Zeitaufwand zur Bedienung der Kanaele von Kanal zu Kanal unterschiedlich. Zudem wird zu unterschiedlichen Zeitpunkten an ein und demselben Kanal mal rege Aktivitaet, mal ueberhaupt keine sein. Die gemachten Ueberlegungen berechtigen zu einer Strategie, die es erlaubt, mehr Zeit fuer die Bearbeitung eines Kanals zu verwenden, als die theoretische Bedienzeit angibt. Insgesamt sollte aber die Rahmenzeit nicht ueberschritten werden, was bedeutet, dass die Summe aller Bedienzeiten kleiner oder gleich der Rahmenzeit sein muss.

Das Verfahren setzt eine Moeglichkeit voraus, die individuellen Bedienzeiten der Kanaele messen zu koennen. Da der KE-Prozessor ueber keine Uhr ("Timer") verfuegt, die das Verfahren wesentlich vereinfachen wuerde, laesst sich die Zeit nur indirekt ueber das Zaehlen der Mikrobefehlszyklen ermitteln.

Das Bild 4 zeigt die Aufteilung der zur Verfuegung stehenden Bedienzeit eines Kanals in die Teilzeiten fuer die einzelnen Bearbeitungsschritte. Dargestellt sind alle, zu einem "unguenstigen" Zeitpunkt denkbaren Aktionen in ihrer tatsaechlichen zeitlichen Folge. Die reine Programmlaufzeit, die bei der Aufbereitung der Parameter fuer die DMA-Transfers anfaellt, sowie die zwischen den DMA-Zyklen benoetigten Verarbeitungszeiten, sind nicht gesondert eingezeichnet. Man sieht, dass gleich sechs DMA-Zyklen hintereinander auftreten koennen.



TB = Bedienzeit des Kanals
TTP = Abfragezeit des Senders
TM = Zeit für DMA,um Zeichen aus HSP zu holen
TTS = Senden des Zeichens an DLV-Modul
TRP = Abfragezeit des Empfängers
TRG = Zeichen aus Empfangspuffer holen
TE = Zeichen zurücksenden (Hardware-Echo)
TO = Organisationszeit für Kanalumschaltung

Bild 4 Aufteilung der Bedienzeit TB

Rechnet man mit einer Latenzzeit von 5 usec pro DMA-Anforderung und jeweils 2usec fuer den Transfer an sich so kommt man bereits auf 42 usec, um nur die Kommunakationsaufgaben zu loesen. Es bleiben folglich noch etwa 16 usec, um reine Verarbeitungsaufgaben durchfuehren zu koennen. Dies reicht bei der angegebenen Situation nicht aus! Sollte dennoch eine solche Situation eintreten, muessen Massnahmen zur Vermeidung von Fehlern getroffen werden. Die dargestellte "worst-case"-Bedingung nach Bild 5 schliesst einen logischen Fehler ein: Die Kombination, das Auto-Echo einzuschalten und zugleich eine davon unabhaengige Ausgabe anzustossen ist sinnlos. Der Ausgabestrom wuerde mit den zurueckzusendenden Zeichen des Eingabestroms vermischt, wodurch undefinierte und wohl auch unerwuenschte Zeichenketten entstehen wuerden. Zum anderen wuerde der Echo-Mechanismus nicht mehr zuverlaessig arbeiten, da es sich nicht vermeiden liesse,dass Zeichen ueberschrieben wuerden.

Der praktische Betrieb zeigt, dass 58.6 usec im Regelfall ausreichen,dennoch darf man Extremfaelle nicht ausschliessen. Im Bild 5 ist ein Fall dargestellt, bei dem nur die Eingabe aktiv ist und das Hardware-Echo abgeschaltet ist. Die dafuer notwendige reale Bedienzeit liegt weit unter der zulaessigen maximalen Zeit TB. Trotzdem wird nicht sofort auf den naechsten Kanal umgeschaltet und mit dessen Bearbei- tung begonnen, sondern es wird gewartet, bis die theoretische Bedienzeit abgelaufen ist. Die Wartezeit tW berechnet sich aus der tatsaechlichen Bearbeitungszeit teff und der maximalen (vorgegebenen) Bedienzeit zu

TW = TB -Teff -TO

Die Umschaltzeit TO kann als konstant angesehen werden und tritt immer vor der Bearbeitung eines neuen Kanals auf.

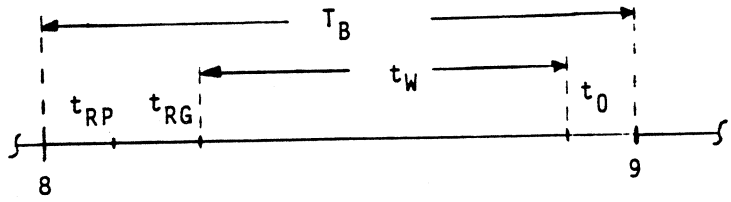


Bild 5 Kanal mit ausschliesslicher aktiver Eingabe

Selbst bei einem abgeschalteten Kanal (theoretischer Bedienzeit=0) werden die zulaessigen 58.6 usek allein durch Warten ueberbrueckt. Dabei wird das schon erwaehnte Ziel eines festen Rasters für die Abtastzeitpunkte erreicht und garantiert, dass der gleiche Kanal erst nach der Zeit 16 x TB wieder bearbeitet wird (Q-Bus-Belastung!)

Wie verhaelt sich nun das Verfahren, wenn mehr als die maximale Bedienzeit benoetigt wird? Dazu ist im Bild 6 die Situation dargestellt. Die Bearbeitung von Kanal 1 erfordert mehr Zeit als theoretisch zulaessig ist. Die über TB hinaus gebrauchte Zeit plus die Umschaltzeit werde mit TVL bezeichnet. Der zweite Kanal ist wenig aktiv und zur Bearbeitung ist nur TEff2 notwendig, so dass sogar noch Wartezeit zur Verfügung steht.

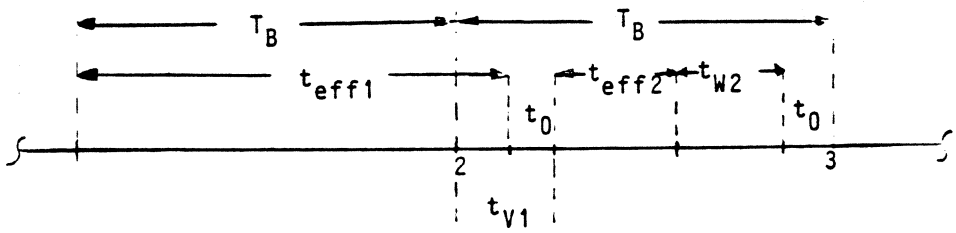


Bild 6 Effektive Bedienzeit des Kanals 1 ist größer als TB

Die Laenge der Wartezeit muss nun so berechnet werden, dass der Rasterpunkt 3 rechtzeitig erreicht wird. Dazu muss der Zeitverlust TVL vom vorherigen Kapitel berücksichtigt werden. Die Zeit TW2

berechnet sich also folgendermassen:
TW2 = TB - Teff2 - TO-TVL

Im allgemeinen kann sich der Zeitverlust, wenn er einmal entstanden ist, über mehrere der folgenden Kanaele hinwegschleppen. Dann naemlich, wenn beim naechsten und mö glicherweise noch beim übernaechsten Kanal die effektiven Bedienzeiten auch relativ gross ausfallen. Wird dann ein Kanal mit potentieller Wartezeit erreicht, so muss diese mit der Summe der vorher entstandenen Verlustzeiten abgerechnet werden.
Die vorgestellte Strategie hat sich im Betrieb bisher bewaehrt. Sie passt sich dynamisch den wechselnden Belastungsverhaeltnissen an und ist, was in diesem Fall wichtig ist, mit vertretbaren Aufwand im Mikroprogramm zu realisieren.

4.3 Software-Schnittstelle

Die Kontrolle und der Datentransfer von und zum Terminal-Multiplexer erfolgt über 8 Register, die das MUX-KE in der I/O-Page des Q-Bus-Systems belegt. Neben dem programmierten Datentransfer werden Daten auch per DMA übertragen. Verschiedene Ereignisse werden per Interrupt der CPU mitgeteilt.

4.3.1 Register des MUX-KE

Das MUX-KE verfügt über folgende 8 Register:

- XXXX00 System Controll Register
- XXXX02 Next Received Character Regfister
- XXXX04 Line Parameter Register
- XXXX06 Current Adress Register
- XXXX10 Byte Count Register
- XXXX12 Buffer Active Register
- XXXX14 Break Controll Register
- XXXX16 Silo Status Register

Die 8 Register sind lückenlos hintereinander angeordnet. Das System muß dabei immer auf einer durch 16 teilbaren Adresse liegen.

4.3.1.1 System Controll Register SCR (XXXX00)

Bit Beschreibung

0-3 Line Selection Bits (R/W)
Jeder der 16 Kanäle hat im MUX-KE seinen eigenen Speicherbereich für die Kanal-Parameter-Informationen, die Current-Address und den Byte-Count. Diese Speicherbereiche werden über das Line-Parameter-Register, das Current-Address-Register und das Byte-Count-Register geladen. Zu diesem Zweck muß jedoch zunächst festgelegt werden, welcher Kanal spezifiziert werden soll. Dies geschiet über die Line-Selection-Bits.

4-5 Memory-Extention-Bits (R/W)
Diese Bits stellen die Adresserweiterung(Bit 16/17) für das Current-Address-Register dar.

6 Receiver Interrupt Enable
Dieses Bit enabled den Receiver-Interrupt (Bit 7)

7 Receiver Interrupt
Dieses Bit zeigt an, daß die Anzahl der im SILO befindlichen Zeichen den Alarm-Level erreicht hat (Alarm-Level siehe SiLo Status Register). Wenn dieses Bit gesetzt wird und Bit 6 gesetzt ist, wird ein Interrupt erzeugt.

8 Clear Non-Existent Memory Interrupt (R/W)
Dieses Bit löscht die Non-Existent-Memory Anzeige (Bit 10) und sich selbst

9 ohne Bedeutung

10 Non-Existent Memory (R) Dieses Bit zeigt an, daß per DMA auf einen nicht existierenden Speicherbereich zugegriffen werden sollte. Ein Interrupt wird erzeugt, wenn Bit 13 gesetzt ist und Non-Existent-Memory aktiv wird.

11 ohne Bedeutung

12 Storage Interrupt Enable (R/W)
Dieses Bit enabled den Storage Interrupt (Bit 14)

13 Transmitter und Non-Existent-Memory Interrupt Enable (R/W)
Dieses Bit enabled den Non-Existent-Memory-Interrupt (Bit 14) und den Transmitter Interrupt (Bit 15)

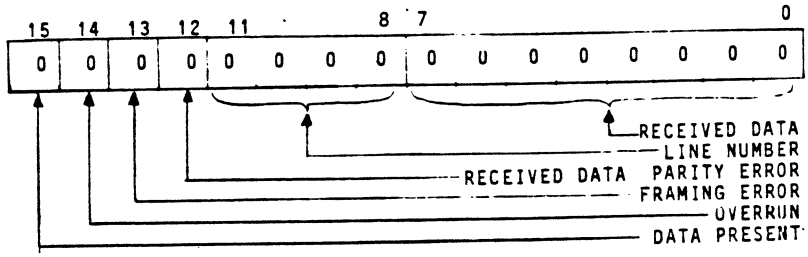
14 Storage Interrupt (R)
Dieses Bit zeigt an, daß ein empfangenes Zeichen aus Platzmangel nicht im Silo abgelegt werden kann. Ein Interrupt wird erzeugt, wenn gleichzeitig Bit 12 gesetzt ist.

15 Transmitter Interrupt (R/W)
Dieses Bit wird gesetzt, wenn bei der Blockübertragung in Senderichtung der Byte-Count Null wird, d.h. alle Zeichen übertragen werden. Es wird gleichzeitig ein Interrupt abgesetzt, wenn Bit 13 gesetzt ist.

Wichtig bei diesem Register ist die Implementierung des Lese/Schreib-Zyklus ("read-modify-write cycle"), da auf einzelne Bit oder Bitgruppen (häufig) zugegriffen wird. Bei allen schreibenden Operationen ist ferner auf die Einhaltung des Schreibschutzes bei einigen Bits des Registers zu achten. Zum Beispiel führt der Versuch, das Receiver-Interrupt-Bit (Bit 7) zu verändern, nicht zum Erfolg. Es kann nur durch das Mikroprogramm selbst gesetzt bzw. gelöscht werden. Dabei bleibt das Interrupt-Bit des Empfängers solange gesetzt, bis die Bedingung, die zum Setzen

diese Bit geführt hat, nicht mehr erfüllt ist. Das heißt, es wird erst gelöscht, wenn die Anzahl der in Silo befindlichen Zeichen den Alarm-Wert erreicht oder unterschritten hat. Weiterhin müssen bei einem schreibenden Zyklus das Interrupt-Bit und das Interrupt-Enable-Bit gemeinsam betrachtet werden. Sind beide Bits gesetzt, ist eine Unterbrechung zu generieren. Der Grund dafür ist folgender: Noch bevor der DH11-Treiber beginnt das Silo auszulesen, setzt er das "Enable-Bit" der Empfängerunterbrechung zurück. Nachdem das Silo geleert worden ist, vergeht eine kurze Zeit, bis die Unterbrechung wieder freigegeben wird. Zwischenzeitlich können aber Zeichen in das Silo eingelesen worden sien, die keine Unterbrechung auslösen durften (Interrupt disable!). Als Folge der Unterbrechungsfreigabe muß deshalb die Unterbrechung "nachgeholt" werden. Würde man nicht so verfahren, käme eine Unterbrechung erst beim nächsten Eintrag eines Zeichens ins Silo zustande.

4.3.1.2 Next Received Charakter Register NRCR (XXXX02)



Bit Beschreibung

0-7 Next Received Charakter (R)
Diese Bits enthalten den Next Received Charakter; rechtsbündig; LSB ist Bit 00.

8-11 Line Number (R)
Kanalindikation für den Next Receive Charakter; LSB ist Bit 08

12 Parity Error (R)
Zeigt an,ob die Paritätsprüfung einen Fehler erkannt hat

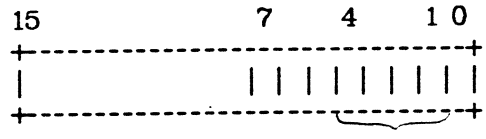
13 Framing Error (R)
Dieses Bit wird gesetzt, wenn der Empfänger ein Stop-Bit erwartet und ein 'Space'-Bedingung findet. Diese Bedingung zeigt gewöhnlich ein Break an

14 Data Overrun (R)
Die Scanner-Einrichtung hat nicht schnell genug ein Zeichen vom Empfangspuffer abgeholt.

15 Valid Data Present (R)
Dieses Bit zeigt an, daß die Bit 00 - 14 gültig sind und von der CPU gelesen werden können.

Das MUX-KE enthält bezüglich dieses Registers eine Erweiterung. Dieses Register hat beim Lesen die Bedeutung wie oben beschrieben. Das MUX-KE besitzt jedoch auch die Möglichkeit dieses Register zu beschreiben.

Beim Beschreiben des NRCR sind den unteren fünf Bits bestimmte Funktionscodes zugeordnet. Einige Funktionscodes haben lediglich einen Effekt auf das Register R15 in CP-RAM, wodurch angezeigt wird, welch Variable bei einem Lesen des Line Parameter Registers (LPR) angesprochen werden soll. Uber den "Umweg" des LPR, lassen sich somit einige MUX-KE-spezifische Informationen , wie Status der Kanäle, Basisadresse u.a., lesen.



Bedeutung der Bits:

Bit 0 = 0 :

Basisadresse BASADR wird durch den gesamten Wert des Wortes (MSB,LSB) ersetzt. Da eine solche Adresse auf jeden Fall gerade sein muß, läßt sich das unterste Bit heranziehen, um die übrigen Funktionscodes vom Verändern der Basisadresse zu unterscheiden.

= 1 :

Bit 1 bis 4 sind als Funktionscodes zu interpretieren.

Funktionscodes:

Bit 1 - 4

0000 = die Zeit bis zur Generierung einer Unterbrechung ist zu verändern : TIMCON wird durch den Wert des MSB ersetzt.

0001 = die minimale Bedienzeit pro Kanal ist zu ändern : DELCON wird durch den Wert des MSB ersetzt.

0010 = Bit 0 in R15 wird gesetzt, um anzuzeigen, daß beim nächsten Lesen des LPR die Basisadresse gelesen werden soll.

0011 = Bit 1 in R15 wird gesetzt, um anzuzeigen, daß beim nächsten Lesen der Baud-Code gelesen werden soll.

0100 = Bit 2 in R15 wird gesetzt, um anzuzeigen, daß beim nächsten Lesen des LPR die minimale Bedienzeit (DELCON) gelesen werden soll.

0101 = dieser Code führt zu einer Neuinitialisierung des MUX-KE Insbesondere werden die Baudraten der Peripheriemodule neu gemessen.

0110 = Bit 3 in R15 wird neu gesetzt, um anzuzeigen, daß beim nächsten Lesen des LPR die Zeit bis zur Unterbrechung (TIMCON)gelesen werden soll.

4.3.1.3. Line Parameter Register LPR (XXXX04)

Beim Beschreiben dieses Registers ist darauf zu achten, daß die Line-Selections-Bits im SCR entsprechend gesetzt sind.

Bit Beschreibung

0-1 Charakter Length (W)

Diese Bits müssen entsprechend der verwendeten Zeichenlänge (Senden/ Empfangen) gesetzt sein

Bit 01 00	Länge (exclusiv Parity)
0 0	5 Bit
0 1	6 Bit
1 0	7 Bit
1 1	8 Bit

(ohne Bedeutung muß auf DLV11-g eingestellt werden)

2 Two Stop Bits (W)

Wenn dieses Bit gesetzt ist und die eingestellte Zeichenlänge 6,7 oder 8 Bit ist, werden 2 Stop-Bits generiert. Bei einer Zeichenlänge von 5 bit werden 1½ Stop-Bits generiert. Ist dieses Bit nicht gesetzt, wird generell 1 Stop-Bit gesendet. (Ohne Bedeutung muß auf DLV11-J eingestellt werden).

3 nicht verwendet

4 Parity Enable (W)

Mit diesem Bit wird die Party-Generierung und Party-Prüfung eingeschaltet (Ohne Bedeutung muß auf DLV11-J eingestellt werden)

5 Even Parity (W)

Entscheidung, ob Even- oder Odd-Parity Generierung bzw. Prüfung durchgeführt wird (nur wenn auch Bit4 gesetzt ist) (ohne Bedeutung muß auf DLV11-J umgestellt werden)

6-9 Receiver Speed (W)

Wird HW-mäßig auf den DLV11-J-Moduln eingestellt.

10-13Transmitter Speed (W)

Wird HW-mäßig auf den DLV11-J-Moduln eingestellt.

14 Half Duplex / Full Duplex (W)

Wenn Bit gesetzt ist, Halb-Duplex Betrieb (ohne Bedeutung)

15 Auto-Echo-Enable

Wenn dieses Bit gesetzt ist, wird jedes auf dem Kanal empfangene Zeichen automatisch zurtückgeschickt.

Dieses Register wurde beim MUX-KE auch als lesbares Register ausgebildet. Damit ist es möglich verschiedene Variable des MUX-KE zu lesen. Folgende Variablen sind lesbar :

-Kanal-Statusregister (LINSTA) (0000)

-Basisadresse (BASADR) (0001)
-Baudregister (BAUDRG) (0010)
-min. Bedienzeit (DELCON) (0100)
-Zeit bis Interrupt (TIMCON) (1000)

Die Auswahl der Variablen erfolgt über entsprchendes Setzen der Bit 0-3 in R15. Die bits in R15 können uber das NRCR gestzt werden. Nach dem Lesezyclus werden die 4 bits in R15 gelöscht.

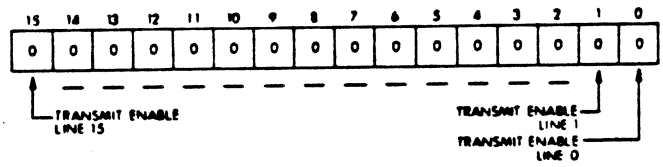
4.3.1.4.Current Address Register CAR (XXXX00) (W/R)

Dieses Register darf nur geladen werden,wenn im SCR die Line Selection Bits entsprechend gesetzt sind. Die 16 Bits aus diesem Register sowie die Extended-Address-Bits werden beim Beschreiben dieser Register in den Speicherbereich des MUX-KE übernommen. Beim Lesen wird die Current-Address aus dem Speicherbereich des MUX-KE in diese Register übertragen. Die Extended Adress-Bits erscheinen im Silo-Status- Register.

4.3.1.5.Byte-Count-Register BCR (XXXX10) (W/R)

Das Register wird mit dem 2-er Komplement der Anzahl der zu sendenden Zeichen geladen. Die Kanalauswahl für den Byte Count erfolgt wie beim CAR.

4.3.1.6.Buffer Active Register BAR (XXXX12) (R/W)



Das Register enthält 1 Bit für jeden Kanal. Das Setzen eines Bits initialisiert die Sende-Operation auf der entsprechenden Leitung.Das Bit wird gelöscht, wenn das letzte Zeichen in das Data Holding Register des UART übertragen wurde. Das Bit zeigt damit jedoch noch nicht an, da alle Zeichen übertragen wurden, dies resultiert aus der Doppelbufferung.

4.3.1.7.Break Control Register BCR (XXXX14)

Das Register enthält wie das BAR für jeden Kanal ein Bit. Das setzen eines Bits generiert unmittelbar die Break-Condition auf dem entsprechenden Kanal.

4.3.1.8.Silo Status Register SSR (XXXX16)

Bits Beschreibung

0-5 Silo Alarm Level (R/W)
Diese sechs Bits dürfen folgendermaßen geladen werden 0,1,2,4,8,16,32 und geben die Anzahl der im Silo befindlichen Zeichen wieder, bei der ein Interrupt an die CPU abgesetzt werden soll. Der Interrupt wird nur abgesetzt, wenn das entsprechende Bit SCR gesetzt ist.

6-7 Read *Extended* Memory (R)
Diese Bits geben die Bits 16, 17 der Current Address wieder. Der entsprechende Kanal muß in SCR definiert sein.

8-13 Silo-Fill-Level (R)
Diese Bits geben den aktuellen Füllstand des Silos wieder.

14-15 ohne Bedeutung

4.3.2. Interrupt vom MUX-KE

Wie bei der Registerbeschreibung bereits erwähnt, kann das MUX-KE bei vier verschiedenen Ereignissen einen Interrupt an die CPU absetzen.

4.3.2.1 Receiver Interrupt (SCR,Bit7)

Dieser Interrupt wird abgesetzt, wenn die Anzahl der Zeichen im Silo den Silo-Alarm-Level erreicht.

4.3.2.2 Storage Overflow Interrupt (SCR,Bit 14)

Dieser Interrupt wird abgesetzt,wenn das Silo gefüllt ist und ein weiteres Zeichen abgespeichert werden soll.

4.3.2.3 Transmitter Interrupt (SCR, Bit 15)

Dieser Interrupt wird abgesetzt, wenn die Ubertragung von einem oder mehreren Datenblocks abgeschlossen wurde, d.h. wenn der jeweilige Byte-Count Null geworden ist.

4.3.2.4 Non-Existent-Memory Interrupt (SCR,Bit 10)

Dieser Interrupt wird abgesetzt, wenn das MUX-KE beim DMA-Transfer auf eine nicht existierende Adresse zugreifen will.

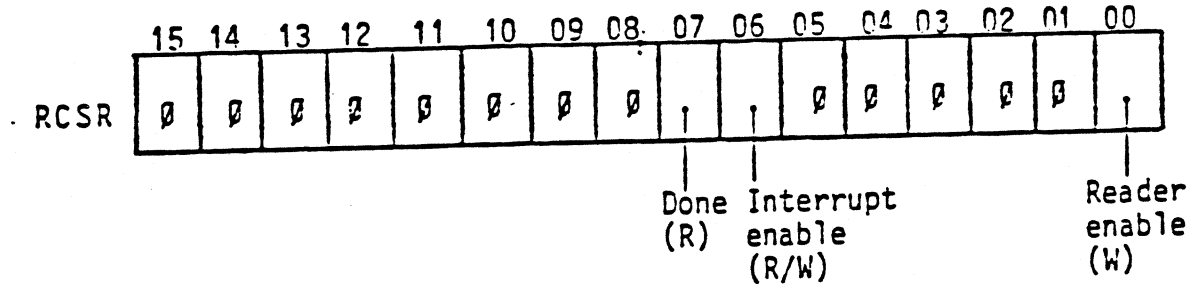
4.3.3 Register der DLV11-J-Module

Die MUX-KE Einheit belegt neben den 8 Registern, die vom DH11-Treiber direkt angesprochen werden, für jeden Kanal weitere 4 Register in der I/O-Page des Q-Bus_Systems. Es handelt sich dabei um folgende Register :

- Receive Control / Status Register RCSR
- Receive-Buffer RBUF
- Transmit Control / Status Register XCSR
- Transmit Buffer XBUF

Diese Register werden nicht vom DH11-Treiber angesprochen, sondern nur per DMA vom Kommunikationsprozessor.

Receive Control / Status-Rgister (XXXXX0)



Bit Beschreibung

8/15 Nicht benötigt.Beim Schreiben = 0 .

7 Receiver Done.
Wird gesetzt, wenn ein vollständiges Zeichen empfangen wurde und für die Eingabe in den Prozessor bereit ist. Das Bit wird automatisch gelöscht, wenn RBUF gelesen wird, BINITL angelegt wird oder das Bit Reader Enable gesetzt ist.
Read Only Bit. Wenn das Bit 6 des RCSR gesetzt ist, bewirkt das Setzen des Bits 7 den Einsprung in eine Interrupt-Sequenz.

6 Receiver Interrupt Enable.Wird unter zprogrammkon- trolle im Falle einer Receiver Interrupt Anforderung gesetzt (wenn ein Zeichen zur Ubertragung in den Prozessor bereit ist). Wird gelöscht durch das Programm oder durch BINIT. Read/Write Bit.

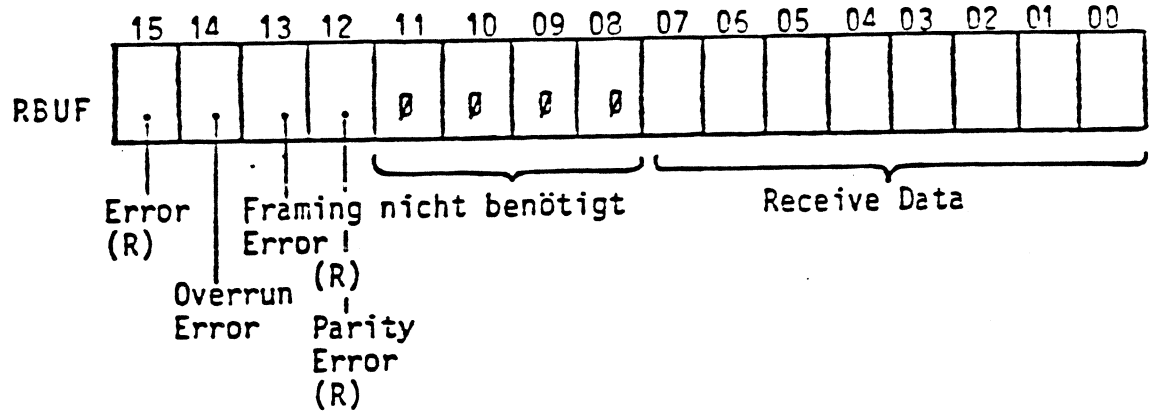
1-5 Nicht benötigt. Beim Lesen = 0

0 Reader Enable. Indem man dieses Bit setzt, geht der Lochstreifenleser amLT33-Terminal um ein Zeichen weiter. Durch das Setzen wird das Done-Bit (Bit 7) gelöscht. Write Only Bit.

Die Benutzung diese Bits setzt das Vorhandensein der DLV11-KA-Option

voraus.

Receive Buffer (XXXXX2)



Bit Beschreibung

15 Channel Error Status. Logisches 'OR' aus Bit 14,13 und 12. Read Only Bit.

14 Overrun Error

Wenn dieses Bit gesetzt ist, wird angezeigt, daß das Lesen eines Datums vor dem Senden eines neuen nicht beendet wurde.Wird durch BINIT gelöscht. Read Only Bit.

13 Framing Error

Das gesetzte Bit zeigt beim gelesenen Zeichen ein ungültiges Stop-Bit an. Wird durch BINIT gelöscht.Read Only Bit.

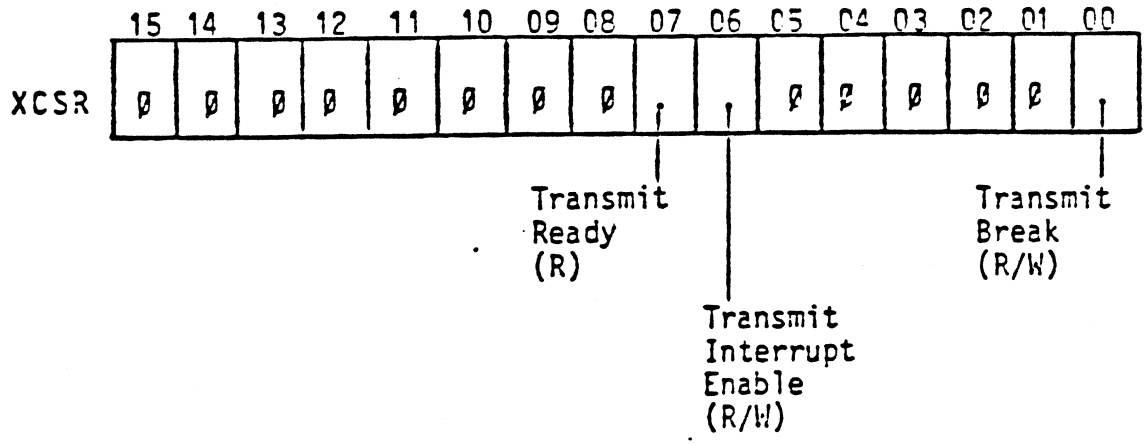
12 Parity Error

Die empfangene Parität stimmt nicht mit der erwarteten überein, wenn dieses Bit gesetzt ist. Falls das Device ohne Parität betrieben wird, ist dieses Read Only Bit immer 0 .

8-11 Nicht benötigt. Beim Lesen = 0

0-7 Daten-Bits. Enthält 7 oder 8 Bits rechtsbündig. Read Only Bits.

Transmit Control/Status-Register (XXXXX4)



Bit Beschreibung

8-15 Nicht benötigt. Beim Lesen = 0

7 Transmit Ready

Wird gesetzt, wenn XBUF leer ist und ein neues Zeichen übertragen werden kann. Es wird ebenfalls durch INIT während einer Power-Up- Sequenz oder während einer Reset-Instruction ge- setzt. Read Only Bit. Wenn das Transmitter Interrupt Enable Bit (Bit6) gesetzt ist, bewirkt das Setzen des Transmit Ready Bits eine Interrupt Sequenz.

6 Transmit Interrupt Enable.

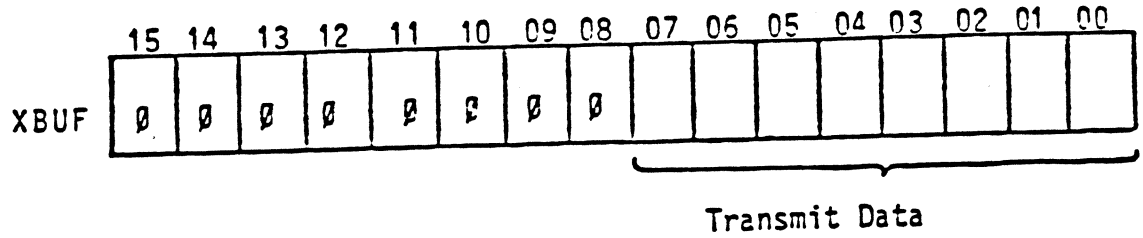
Wenn es erforderlich wird, eine Transmit Interrupt Anforderung zu genererieren, wird dieses Bit unter Programmkontrolle gesetzt. Während Power-Up oder Init-Function wird dieses Bit gelöscht. Read/Write Bit.

1-5 Nicht beötigt. Beim Lesen = 0

0 Transmit Break.

Wird unter Programmkontrolle gesetzt oder gelöscht. Wenn es gesetzt ist, wird eine zusammenhängende Leerzeichenkette übertragen. Transmit Done und Transmit Interrupt können jedoch weiter benutzt werden. Im gelöschten Zustand kann eine normale Zeichenübertragung stattfinden. Wird durch BINIT gelöscht. Read/Write Bit.

Transmit Buffer (XXXXX6)



Bit Beschreibung

8-15 Nicht benötigt. Beim Lesen = 0

0-7 Daten-Bits.
Enthält sieben oder acht rechtsbündige Datenbits. Werden unter Programmkontrolle für die serielle Übertragung geladen.

4.4 MUX-KE-Mikroprogramm

Die Emulation des DH11 wird weitgehend durch das Mikroprogramm des Kommunikationsprozessors realisiert. Das grobe Flußdiagramm des Programms ist im Bild zu sehen.

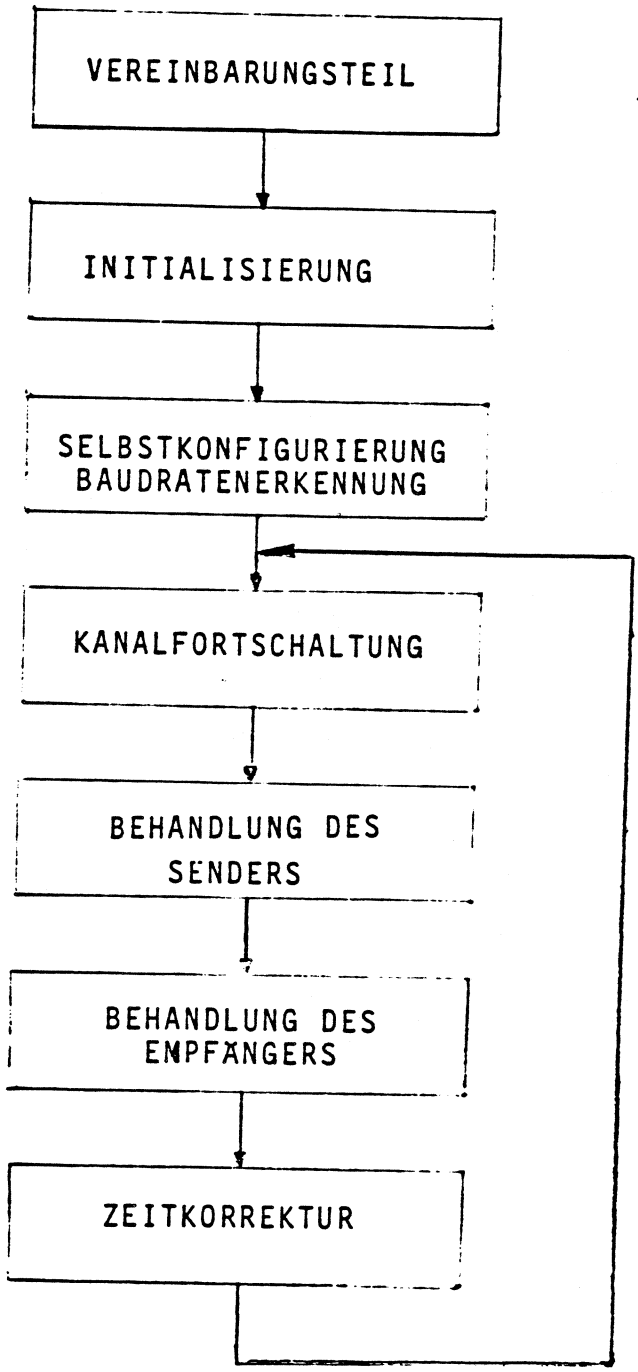
Nach der Initialisierung des Gesamtsystems (Bus-Init) durchläuft das Mikroprogramm zunächst den Vereinbarungsteil, die Initialisierung, die Selbstkonfiguration und die Baudratenerkennung. Diese Teile werden nur einmal durchlaufen.

Eine Besonderheit stellt dabei die Baudratenerkennung dar. Wie bereits erwähnt ist es sinnvoll eine Abfrage der einzelnen Kanäle nur entsprechend der Baudrate durchzuführen. Die Erkennung der hardwaremäßig eingestellten Baudraten auf dem DLV11-J-Modulen wird durch das Mikroprogramm selbst festgestellt, so daß der Benutzer keine gesonderte softwaremäßige Einstellung vornehmen muß.

Die folgenden Programmteile werden in einer Endlosschleife durchlaufen. Dabei erfolgt jeweils die Kanalfortschaltung, die Behandlung des Senders, die Behandlung des Empfängers und die Zeitkorrektur.

Die detaillierte Beschreibung des Mikroprogramms ist der Beschreibung KE-DH11-Mikroprogramm zu entnehmen.

Bild 6 : Programmaufbau



4.4.1 Selbstkonfigurierung

Durch die Verwendung von Standard DLV11-J-Modulen als Schnittstellen einheiten,ist es nicht möglich, softwaremäßig die dort eingestellten Baudraten zu ändern. Die Leitungsparameter werden durch Drahtbrücken festgelegt und können für jeden der vier Kanäle unterschiedlich sein. Der DH11-Multiplexer aber besitzt die Fähigkeit der programmgesteuerten Einstellung aller Leitungsparameter. In der Systeminitialisierungsphase werden sämtliche Kanäle mit der Baudrate 110/sek programmiert, was folglich für den emulierten Multiplexer keine Wirkung haben kann. Die Baudratenwerte kpnnen nicht an die DLV-Module weitergegeben werden. Trotzdem ist das Line-Parameter-Register dahingehend zu emulieren, daß zumindest eine "Null-Baudrateneinstellung" berücksichtigt wird.

Während beim originalen DH11-Multiplexer immer davon ausgegangen werden kann, daß sämtliche 16 Kanäle physikalisch vorhanden sind - die 16 UART sind Teil der Multiplexer-Hardware - ist beim MUX-KE dies nicht zwingend notwendig. Es können sich auch weniger als vier DLV11-J-Module im System befinden. Dann aber ist es sinnvoll, die nicht vorhandenen Kanäle gar nicht erst in den Abfragezyclus mit aufzunehmen, um nicht Adressen der I/O-Page anzusprechen, die keinem (DLV-)- Register zugeordnet sind. Eine der zwei Aufgaben des Programmoduls "Selbstkonfigurierung und Baudratenerkennung" ist es, in einem durch die Basisadresse festgelegten Adreßbereich solche nicht gültigen Adressen festzustellen. Die Selbstkonfigurierung ordnet jeder nicht vorhandenen Geräteadresse einen nicht existierenden Kanal zu; in den entsprechenden Kanal-Status-Registern (LINSTA) wird das durch Löschen des Bits 7 vermerkt.

Das Verfahren zur Selbstkonfigurierung bringt zwei Vorteile mit sich :

1.)Es brauchen nicht alle 16 Kanäle (alle 4 DLV11-J -Module) physikalisch vorhanden zu sein. Nicht vorhandene Module führen zu keinem Fehlverhalten des MUX-KE.

2.)Nicht existente DLV-Module werden später vom zyclischen Abfragen ausgeschlossen, so daß dies zusätzlich zur Verminderung der Q-Bus Belastung beiträgt.

Ausschließliches Ziel der nun zu erklärenden automatischen Baudratenerkennung ist die Reduzierung der Anzahl der DMA- Transferoperationen in Abhängigkeit der eingestellten Baudraten.

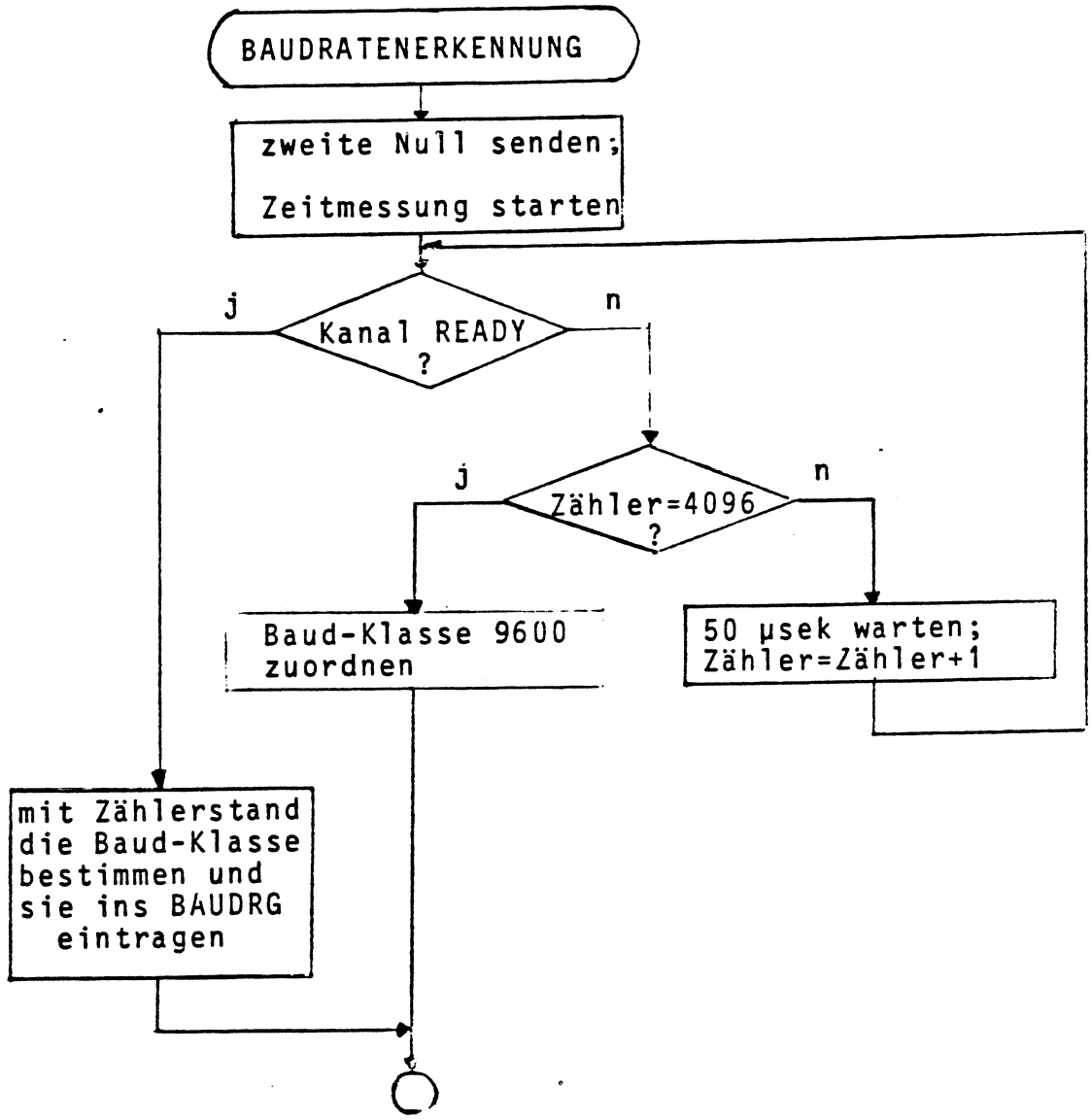
4.4.2 Automatische Baudratenerkennung

Im Vorangegangenen ist die Baudrate von 9600 Baud als Berechnungsgrundlage für die Rahmenzycluszeit von ungefähr einer Milli- sekunde genommen worden. Jeder Kanal muß spätestens nach einer Millisekunde wieder bedient werden können. Befinden sich nun außerdem Kanäle mit niedrigeren Baudraten im System, so werden diese Schnittstellen häufiger (und damit unnötigerweise!) abgefragt, als es ihrer Baudrateneinstellung entsprechen würde. Ein Kanal mit 2400 Baud braucht nur alle 4 Millisekunden, einer mit 600 Baud sogar nur alle 16 Millisekunden abgefragt werden.
Der Zweite Teil des Unterprogramms "Selbstkonfigurierung und Baudratenerkennung" versucht, die eingestellten Baudraten der Kanäle zu messen, daraus einen Wert zu bilden, der dem Kanal zugeordnet wird. Beim zyclischen Durchgehen der Kanäle in der Senderschleife (Transmitter-Loop) kann mit Hilfe dieses Wertes und dem Wert eines Zählers geprüft werden, ob ein Kanal jetzt oder zu einem späteren Zeitpunkt abgetastet werden muß. Es ist bei der gewählten Organisation des Abtastmechanismus nur sehr schwer möglich, jede einstellbare Baudrate individuell in die Optimierungsstrategie aufzunehmen: Zum einen würde ein solches Verfahren zusätzliche Zähler zur Abrechnung der einzelnen Zeiten benötigen (mindestens 16 x 2 Byte), zum anderen ist der relative Leistungsgewinn hinsichtlich der Busbelastung bei z.B. 135.5 Bd gegenüber 150 Bd sehr gering!
Diese Uberlegungen gaben Anlaß, eine Kompromißlösung zu entwickeln,die alle zugelassenen Baudraten in vier Baudratenklassen (Baud-Klassen) eingeteilt. Es sind die Klassen 9600 Bd, 4800 Bd, 2400Bd und 1200 BD.

Die Realisierung der Baudratenmessung wird wie folgt durchgeführt :
Nachdem ein Kanal als verfügbar erkannt worden ist, wird einige Mikrosekunden später eine zweite binäre Null ("ASCII- Null") andas gleiche Register gesendet. Dort verbleibt das Zeichen, bis die zuerst gesendete Null aus dem Schieberegister des UART verschwunden ist.Solange bleibt auch das READY-Bit im Control/Statusregister (XCSR) des Kanals gelöscht. Durch Abfragen dieses Bits im 50-Mikrosekunden Takt wird mit Hilfe eines 2-Byte Zählers die Zeit bis zum erneuten READY-Status gemessen. Bei der Messung ist folgendes zu beachten:

Die zweite auszugebende Null kann erst einige Mikrosekunden (10-20usek) nach der ersten an das Register des DLV-Moduls geschickt werden, da die Reaktionszeit der Karte zu beachten ist. Sendet man das zweite Zeichen in zu kurzem Abstand nach dem ersten, wird das erste überschrieben und die Messung liefert falsche Werte.

Bild 7 Automatische Baudratenerkennung



Es wird davon ausgegangen, daß sich bei Beginn der Messung keine Zeichen im UART des entsprechenden Kanals befinden. Es kann sein, daß die Existenz des Kanals zwar erkannt wird, die eingestellte Baudrate sich auf diese Weise aber nicht feststellen läßt. Dieser Fall kann eintreten, wenn der angesprochene UART mit externem Takt versorgt wird, der Taktgeber aber ausgeschaltet ist. Um nicht in einer Endlos- schleife zu verbleiben, muß ein Abbruchkriterium geschaffen werden. Dazu wird der Wert des 50-Mikrosekundengebers überwacht. Erreicht er den Wert 4096, wird daraus geschlossen, daß die Baudrate in diesem Fall nicht ermittelt werden kann. Der Wert 4096 entspräche einer Schrittgeschwindigkeit von etwa 44 Bd, die auf der DLV-Karte nicht einstellbar ist. Damit auch dieser Kanal während des Betriebs (das Gerät kann ja erst später hinzugeschaltet werden) sicher bedient werden kann, ist für ihn 9600 Baud vorzusehen.

4.5. Besonderheiten des MUX-KE

Die Emulation des DH11-Multiplexers hat die Möglichkeit geboten, zusätzliche Funktionen zu realisieren. Zum einen sind es Funktionen, die den internen Ablauf de MUX-KE manipulieren, zum anderen lediglich Kontroll- und Testfunktionen.

4.5.1 Zeitabhängiges Auslösen des Empfängerinterrupts

Standardmäßig ist der Alarm-Wert des Silos auf Null gesetzt. Das bedeutet, dab jedes in ein leeres Silo eingetragene Zeichen auch eine Unterbrechung in der CPU auslöst. Setzte man den Alarm-Wert ungleich Null, entstünde eine Unterbrechung erst dann, wenn die Anzahl der Zeichen im Silo den Alarm-Wert übersteigt. Der Zeitpunkt der Unterbrechung hängt also von der Zeichenanzahl ab. Die in der Emulation zusätzlich implementierte Funktion erlaubt, den Zeitpunkt des Unterbrechung anzugeben. Mit Hilfe eines Kommandos zB in einer RSX Task, läßt sich die Zeitspanne zwischen dem Zeitpunkt des Eintrags eines Zeichens in ein leeres Silo und der Ezeugung der dazugehörenden Unterbrechung festlegen. Der wählbare Zeitbereich liegt dabei zwischen 0 und 255 Millisekunden. Sinnvoll ist eine Einstellung dieser Zeit nur, wenn auch der Alarm-Wert des Silos von Null verschieden gewählt wird.Nur dann ist es möglich, während der eingestellten Zeitspanne einige Zeichen zu "sammeln" und sie durch eine Unterbrechung gewissermaßen als Datenblock an die CPU zu schicken.

4.5.2 Veränderung der normalen Bedienzeit

Die minimale Bedienzeit eines Kanals ist in der Systemvariablen DELCON abgelegt. Sie wird bei der Initialisierung mit einem Wert vorbelegt, der einer Bedienzeit von ca. 54 Mikrosekunden entspricht. Diese Variable ist durch ein Kommando in einer RSX Task veränderbar. Die kleinste einstellbare Bedienzeit ist ca. 4 usek, die größte mögliche etwa 102 usek. Für die Praxis bedeutet

dies, daß man in einem gewissen Rahmen die Abtasthäufigkeit variieren kann.
Dadurch ist es möglich, auch Kanäle mit Baudraten zu betreiben die größer sind als 9600 baud. Soll z.B. ein Kanal mit 19200 Baud betrieben werden, so ist es notwendig die Bedienzeit zu halbieren (27 usek) . Damit wird dieser häufig abgefragt. Die Busbelastung steigt folglich an. Sie läßt sich aber reduzieren, wenn nicht benutzte Kanäle aus dem Abfragezyclus herausgenommen werden. Die Abschaltung der Kanäle ist unter RSX z.B. durch das Kommando SET /SPEED=TTX:0:0 zu erreichen, wobei TTX die abzuschaltende Schnittstelle bezeichnet. Eine Reduzierung der Kanalanzahl auf acht, würde es erlauben, 8 Schnittstellen mit je 19200 Baud Schrittgeschwindigkeit zu betreiben. Dabei wäre die Busbelastung gleich der des Standardfalles (16 x 9600 Baud) und die für die Kanäle zur Verfügung stehende Bedienzeit läge im Mittel auch bei ca. 54 usek (hierin sind die "Reservezeiten" der abgeschalteten Kanäle enthalten!). Bei einem entsprechenden Verhältnis von Bedienzeit zu Kanalzahl sind grundsätzlich auch noch höhere Baudraten als 19200 Baud zugelassen.

4.5.3 Veränderung der Basisadresse

Die Basisadresse des MUX-KE ist definiert als die Adresse des Receiver Control/Status Registers (RCSR) des ersten DLV11-J-Moduls. Alle weiteren Registeradressen der noch folgenden Module werden aufsteigender, lückenloser Reihenfolge erwartet. Die Systemvariable BASADR beinhaltet die Basisadresse, die mit dem Standardwert 176500 (oktal) vorbelegt ist. Eine Veränderung dieser Adresse, z.B. RSX Task, verschiebt den Adreßbereich der DLV-Register. Dadurch wird es möglich, die Schnittstellenmodule auch auf andere, als auf Die Standardadressen einzustellen. Die Befehle, die zur Anderung der Adresse und zu der dann notwendugen softwaremäßigen Neuinitialisierung des MUX-KE notwendig sind, können mit Hilfe der RSX Task DHMON durchgeführt werden. Eine andere Möglichkeit ist, die entsprechende Befehlssequenz in einem Programm niederzuschreiben, und dieses Programm dann beim "Startup" des Systems aufzurufen.

4.5.4 Die RSX-Task DHMON

DHMON ist ein unter RSX11M entwickeltes Assemblerprogramm, welches zusammen mit dem MUX-KE einige Kontroll- und Steuerfunktionen erlaubt. Das Programm arbeitet mit den Geräteregistern des MUX-KE, wodurch besondere Vorsicht beim Umgang mit diesem Programm geboten ist. Manche Kommandos lassen sich nur von der Konsole des Systems sinnvoll ausführen, da sie unabhängig vom KE-Prozessor arbeitet.

Nach dem Starten der Task DHMON meldet sich das Programm und erwartet eine Eingabe, die immer mit "carriage return, line feed" abgeschlossen werden muß. Die Kommandos, deren Syntax und Wirkung, sind im folgenden zusammengestellt :

BA ändern der Basisaddresse;
nach Anzeige der eingestellten Basisadresse, kann die neue in oktaler Form eingegeben werden.

BR,E(A),NR Break-Modus der angegebenen Kanäle
ein-(E) oder ausschalten(A)

DI ändern der Bedienzeit (DELCON);
Eingabe dezimal; Wert zwischen 4 und 102

DM,E(A),NR schaltet den DMA ein bzw. aus,
indem die entspr. Bit im BAR verändert werden

EC,E(A),NR schaltet den Hardware-Echo-Modus der angegebenen Kanäle
ein bzw. aus

EN beide Unterbrechungen freigeben (ENABLE)

IN Neukonfigurierung u. Initialisierung;
- führt Baudratenmessung durch
- schaltet alle Kanäle ab
- sperrt alle Unterbrechungen
- Basisadresse bleibt unverändert

KA,E(A),NR schaltet die angegeben Kanäle ein bzw. aus.
(Bem.: SET /SPEED führtbezgl. des KE die gleiche Funktion aus, macht diesen Zustand aber auch noch dem Betriebssys.bekannt)

Q Quit = Ende des Programms

RE,L DH11-Register lesen und anzeigen

SI verändern des Silo-Alarm-Wertes;
Eingabe dezimal; Wert zwischen 0 und 63

ST,NR Status der angegebenen Kanäle lesen und anzeigen

UE,S(F) Unterbrechung des Empfängers sperren (S) bzw. freigeben (F)

US,S(f) Unterbrechung des Senders sperren (S) bzw. freigeben (F)

VZ Zeit bis zur Generierung des Unterrupts einstellen;
Eingabe dezimal; Wert zwischen 0 und 255

Bemerkungen zur Syntax:

E(A) = entweder E oder A
S(F) = entweder S oder F
NR = Zahlen von 0 bis 15, durch Komma getrennt
oder/und Bereichsangaben der Form Zahl1-Zahl2

Beispiele:

ST,0-8 gibt den Status der Kanäle
0 bis 8 aus

ST,4,6-9,14,15
gibt den Status derKanäle
4,6 bis9, 14 und 15 aus

5. Leistungsvergleiche bei System mit Multiterminal-Betrieb

5.1 Leistungsvergleich unter Betriebssystem RSX11-M

Für den Leistungsvergleich wurden die Verhältnisse für Eingabe und Ausgabe getrennt untersucht. Der Vergleich wurde zwischen MUX-KE, DLV11-Modul und DLV11-J-Modul (allein ohne MUX-KE) durchgeführt.

Bei der Ausgabe sollte festgestellt werden, welcher Zeitanteil der für bestimmte Aufgaben benötigten Zeit zur Ausgabe eines Bits verbraucht wird. Damit ist ein Vergleich möglich, wie die Ausgabe eines Bits das System belastet. Zur praktischen Durchführung wurde eine Task (Task1) gestartet, die intern bestimmte Operationen durchführte. Parallel dazu wurden Ausgabe-Tasks gestartet, die abgebrochen wurden, sobald Task1 beendet war. Es konnte danach festgestellt werden, wieviel Bit durch die Ausgabe Tasks gesendet wurden. Dadurch konnte die prozentuale Belastung durch die Ausgabe eines Bit berechnet werden. Die Ergebnisse wurden jeweils auf den Wert der KE-MUX-Konfiguration normiert.

Prozentuale Belastung des Systems pro Bit (normiert auf KE-MUX)

Anzahl der Ausgabekanäle	KE-MUX	DZV11	DLV11-J
1	1	1.7	1.6
2	1	1.7	1.6
3	1	1.6	1.6
4	1	1.6	1.6

Damit kann durch das Mux-KE die Ausgabe ungefähr doppelt so schnell, wie bei DZV11-J erfolgen. Bei der Eingabe wurde der Test nur jeweils mit einem Kanal durchgeführt. Es wurde wie beim Ausgabetest wieder die prozentuale Belastung pro Bit festgestellt. Die Task1 konnte vom Ausgabetest übernommen werden, die Ausgabe Task wurde durch eine Eingabe Task ersetzt. Da bei der Eingabe die Zeichen nach einer maximalen Zeit vom Treiber abgeholt werden müssen um nicht durch das nächste Zeichen überschrieben zu werden, konnte der Test nur mit gepufferten Modulen durchgeführt werden. Der Test war mit dem DLV11-J -Modul nicht möglich.

Prozentuale Belastung des Systems pro Bit (normiert auf MUX-KE)

Anzahl der Eingabekanäle	MUX-KE	DZV11
1	1	0.95

Durch das MUX-KE wird das System bei der Eingabe geringfügig mehr belastet, da jedes Zeichen zweimal über den Q-Bus transferiert werden muβ (DLV11-J --> MUX-KE --> Speicher). Dieser Nachteil fällt allerdings bei der Terminaleingabe nicht ins Gewicht.

5.2 Leistungsvergleich unter Betriebssystem MUNIX

Das Betriebssystem Munix erfordert im Multi-User-Betrieb Eingabekanäle mit Bufferung, da im anderen Fall Zeichen verloren gehen. Ein Vergleich konnte deshalb nur zwischen dem Modul DZV11 von DEC und dem MUX-KE durchgeführt werden.

Hardware: - QU68000 Modell 230 mit 512 kB Dual Port Speicher
- 3 Terminals angeschlossen über
a) SLX-4 (DZV11-J)
b) MUX-KE mit SLU-4(DLV11-J)

Software: MUNIX 1.3

Gemessen wurde die Zeit, die zur Ausgabe der Datei "termcap.src" (38933 Bytes lang) auf einem Terminal (DSG101) benötigt wird, während auf den beiden anderen Terminals dieselbe Datei ausgegeben wird (tt2)

Shell-Prozedur für tt2:

```
cat/etc/termcap.src >/dev/tty10&
cat/etc/termcap.src >/dev/tty11&
time cat/etc/termcap.src >/dev/tty12&
```

Gemessen wurde ebenfalls die Zeit bei insgesamt 2, bzw. 1 Terminal (tt1,tt0)

Mux-KE

	real	user	sys	total	CPF	
tt0	1.08	1.1	7.9	9.0	0.083	
tt1	1.08	1.1	8.4	9.5	0.087	
tt2	1.08	1.1	8.6	9.7	0.089	

DZV11

	real	user	sys	total	CPF	
tt0	1.09	1.0	11.3	12.3	0.112	
tt1	1.08	1.0	13.9	14.9	0.137	
tt2	1.09	1.2	17.2	18.4	0.168	

Ergebnis:Bei 3 Terminals ist nach außen hin die Reaktionszeit in beiden Fällen gleich (real). Hingegen ist die System-Belastung (total) beim MUX-KE nur halb so groß wie bei der DZV11.

6. Installationshinweise

6.1 Hardware

Die Module für die MUX-KE-Einheit müssen vor der Installation im System durch verschiedene Brücken konfiguriert werden.

6.1.1 Kommunikationsprozessor CP (B900.475)

Auf dem Kommunikations-Prozessor-Board besteht nur die Möglichkeit die Basisadresse zu ändern. Des weiteren müssen die entsprechenden Mikroprogramm-PROMs und die Steuer-PROMs installiert sein. Der MIKI-BUS Stecker wird in dieser Anwendung nicht benötigt und bleibt frei.

- Einstellung der Basisadresse
Die Basisadresse wird durch fünf Brücken (Header) in Position C8 eingestellt. Bei der Einstellung der 16 Bit breiten Adresse ist es nur notwendig die Bitpositionen 8-12 zu konfigurieren. Die Adressbits 13-15 sind redundant, da das Device sich in der I/O-Page befindet und dieser obere Adreßraum durch das Q-Bus-Signal BBS7 ausgewählt wird. Die unteren 8 Adressbits (0-7) werden softwaremäßig decodiert. Bei der Adresseinstellung ist darauf zu achten, daß die Adresse invertiert einzustellen ist.Wird die Brücke auf dem Header durchtrennt, hat das Bit den Wert 1.
Für die Standardeinstellung des MUX-KE ist die Adresse 160000 (oktal)

```
+-----------------------------+
| 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 |  MUX-KE-Adresse
+-----)---------)---------------+
| BBS7  Brücken   softwaremässig
|                      decodiert
```

```
+-----------------------------
| 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 |  invertierte
+-------)-------)---------------+    Mux-KE-Adresse
|         Brücken
```

Die fünf Brücken müssen also durchtrennt sein, oder der Header wird vollkommen aus dem Sockel entfernt.

- Steuer-PROMs
Auf dem Board befinden sich 3 Steuer-PROMs. Die PROMs müssen folgende Bezeichnug haben :

Position CC8 : R0900.059 CC8 EC1
Position CC55 : R0900.059 CC55 EC2
Position Z51 : R0900.059 Z51 SMS

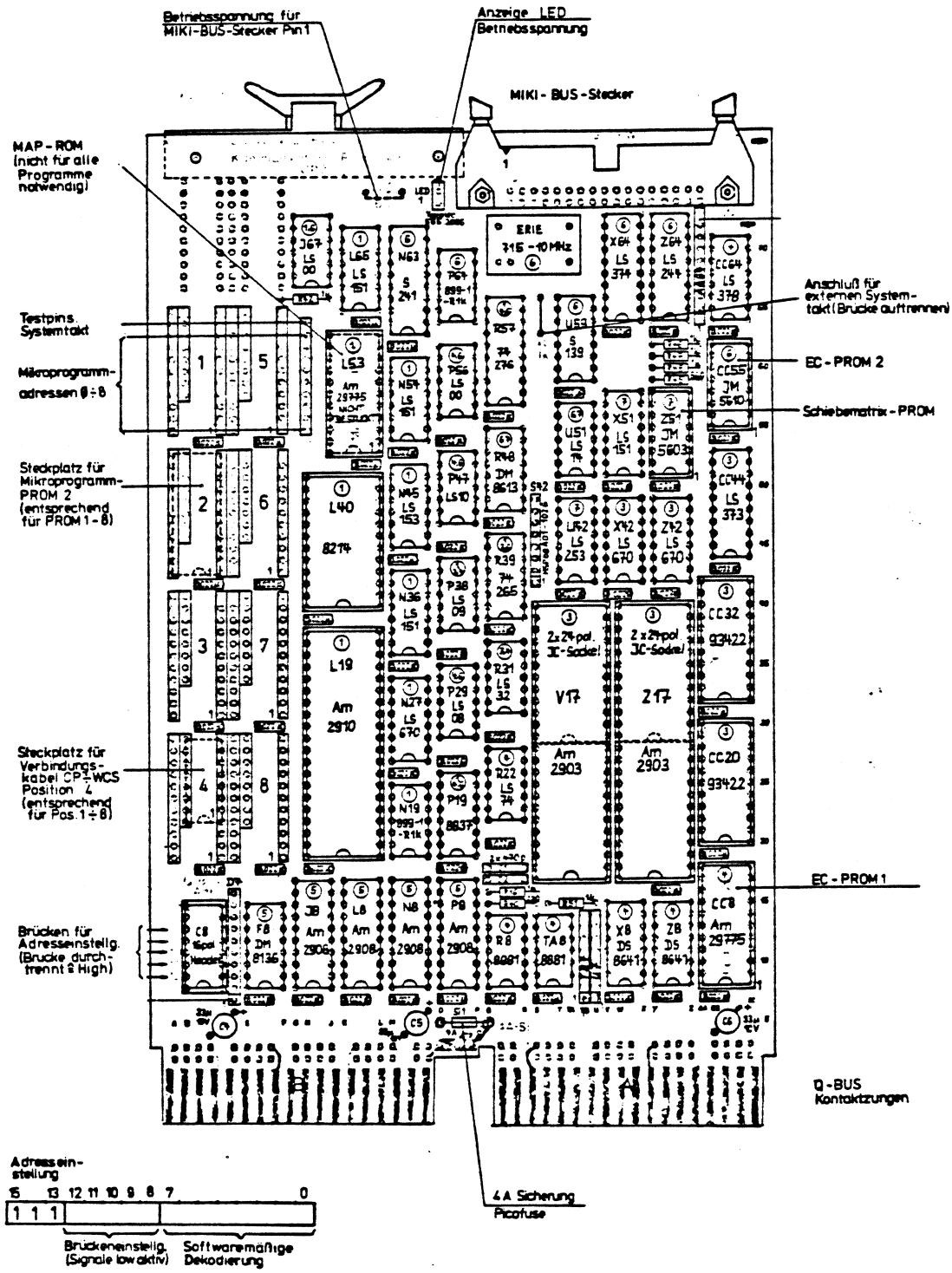
- Mikroprogramm-PROMs

Das Mikroprogramm ist in 9 PROMs untergebracht

Position C55 : R900.060 C55 KE-MUX1
Position C43 : R900.060 C43 KE-MUX2
Position C31 : R900.060 C31 KE-MUX3
Position C19 : R900.060 C19 KE-MUX4
Position F55 : R900.060 F55 KE-MUX5
Position F43 : R900.060 F43 KE-MUX6
Position F31 : R900.060 F31 KE-MUX7
Position F19 : R900.060 F19 KE-MUX8
Position L53 : R900.060 MAP KE-MUX

Die Header- und PROM-Positionen sind dem Plan zu entnehmen

Layout Kommunikationsprozessor CP 2



6.1.2 Serielle Schnittstellenkarten DLV11-J (B900.610)

Das MUX-KE kann maximal 16 Kanäle bedienen, diese Kanäle werden physikalisch durch 4 serielle Schnittstellenkarten DLV11-J mit je 4 Kanälen realisiert. Jeder Kanal kommuniziert über 4 Register (8 Byte) mit dem Prozessor, so daß bei Maximalausbau 128 Bytes in der I/O Page belegt sind. Die Registeradressen müssen für je 4 Kanäle durch die hardwaremäßige Einstellung einer Basisadresse definiert werden. Weitere hardwaremäßige Einstellungen sind für Zeichenformat, Baudrate und Schnittstellenanpassung notwendig. Die Einstellung eines Interruptvektors ist nicht notwendig, da die DLV11-J -Module selbst keinen Interrupt in dieser Anwendung absetzen.

- Adresseneinstellung: A5-A12(für jede Karte)

Mit den Jumpern A5-A12 läßt sich die Basisadresse (Adresse des RCSR von Kanal 0) für jede Karte einstellen. Alle weiteren Adressen der Karte folgen aufeinander. Standardmäßig sind für die 4 Module folgende Adressen einzustellen.

1.Modul:176500
2.Modul:176540
3.Modul:176600
4.Modul:176640

-Vektorumstellung : V5-V7 (für 4 Kanäle)

Ohne Bedeutung bei MUX-KE-Anwendung

-Definition System Konsole: C1-C2 (1x für 4 Kanäle)

Die Jumper ermöglichen die Festlegung,daß Kanal 3 einer der 4 DLV11-J-Module als Systemkonsole verwendet wird. Dieser Kanal steht dann für das KE-MUX nicht mehr zur Verfügung. Der Modul auf dem diese Definition durchgeführt wird, muß ein der folgenden Basisadressen haben: 176500, 176540, 177500

-Break response : (nur für Kanal 3)

Mit dem Jumper läßt sich festlegen, wie das System auf die Break-Bedingung auf Kanal 3 reagiert

- Einstellung Flankensteilheit der Signale R10,R23
(1x für 2 Kanäle)

Mit den beiden Widerständen läßt sich die Flankensteilheit bei den verschiedenen Baudraten einstellen.

- Baudrateneinstellung : U,T,V,W,Y,L,N,K,Z (für jeden Kanal)

Auswahl der Baudrate von150 bd-38,4bd

-Zeichenlänge : D (für jeden Kanal)

Zeichenlänge 7 oder 8 Bit

- Stopbit-Anzahl : S (für jeden Kanal)

Anzahl der Stopbits 1 oder 2

- Parity : P (für jeden Kanal)
  Einschalten der Parityprüfung bzw. Generierung
- Parity-Definition : E (Für jeden Kanal)
  Einstellung Even/Odd-Parity
- Definition der Physikalischen Schnittstelle : N,M (für jeden Kanal)
- Definition der physikalischen Schnittstell V24, RS422,20mA

Die Standardeinstellung für den 1.Modul ist der nachfolgenden Tabelle und Skizze zu entnehmen.

Kanalunabhängige Brücken

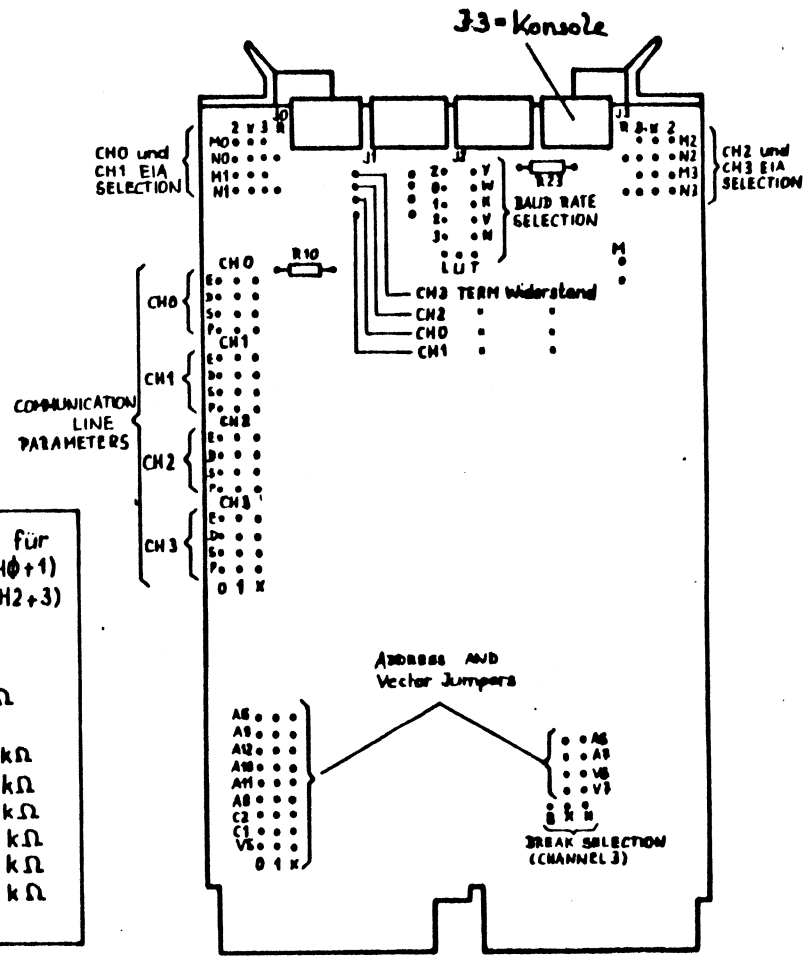
Brücke	Bed	Stellung	Std	Anw	Brücke	Bed.	Stellung	Std.	Anw	Brücke	Bed	Stellung	Std.	Anw
A12	Adresse	X-0 X-1	X		A6	Adr.	R J	X		C1	Konsole auf	X-0 X-1	X	
A11		X-0 X-1	X		A5		X-0 X-1	X		C2	CH3	X-0 X-1		
A10		X-0 X-1	X		V7	Basis - Vec	R J			Break Reakt. CH3	Inh. Halt Boot	R X-H X-B		
A9		X-0 X-1	X		V6		R J							
A8		X-0 X-1	X		V5	Konsole	X-0 X-1 R			R10	Impulsformung bei V24 je nach Baud-Rate (Tabelle)		120k	
A7		R J	X		M		R J	X		R23			120k	
Std.-Adr. 176500														

Einstellung je Kanal

Brücke	Bedeutung	Ø Stellung	Ø Std	Ø Anw	1 Stellung	1 Std	1 Anw	2 Stellung	2 Std	2 Anw	3 Stellung	3 Std	3 Anw
Baud-rate	150 bd	0-U			1-U			2-U			3-U		
	300 bd	0-T			1-T			2-T			3-T		
	600 bd	0-V			1-V			2-V			3-V		
	1,2 kbd	0-W			1-W			2-W			3-W		
	2,4 kbd	0-Y			1-Y			2-Y			3-Y		
	4,8 kbd	0-L			1-L			2-L			3-L		
	9,6 kbd	0-N	X		1-N	X		2-N	X		3-N	X	
	19,2 kbd	0-K			1-K			2-K			3-K		
	38,4 kbd	0-Z			1-Z			2-Z			3-Z		
D	7 Datenbits	X-0			X-0			X-0			X-0		
	8 Datenbits	X-1	X		X-1	X		X-1	X		X-1	X	
S	1 Stop Bit	X-0			X-0			X-0			X-0		
	2 Stop Bits	X-1	X		X-1	X		X-1	X		X-1	X	
P	Parity enable	X-0			X-0			X-0			X-0		
	Parity inhibit	X-1	X		X-1	X		X-1	X		X-1	X	
E	Odd Parity	X-0			X-0			X-0			X-0		
	Even Parity	X-1	X		X-1	X		X-1	X		X-1	X	
M+N TERM	EIA RS422	X-2 100Ω			X-2 100Ω			X-2 100Ω			X-2 100Ω		
M+N	V24	X-3	X		X-3	X		X-3	X		X-3	X	
M	20mA mit 900.611	X-3			X-3			X-3			X-3		
N		X-R			X-R			X-R			X-R		

Tabelle für
R10 (CHØ+1)
R23 (CH2+3)

150 bd	1MΩ
300 bd	1MΩ
600 bd	1MΩ
1,2 kbd	820 kΩ
2,4 kbd	430 kΩ
4,8 kbd	200 kΩ
9,6 kbd	120 kΩ
19,2 kbd	51 kΩ
38,4 kbd	22 kΩ



R = entfernt

J = eingesetzt

6.1.3.Installation der Module im System

Alle Module (CP + 4xDLV11-J) sind in der Q-Bus-Backplane zu installieren. Die Reihenfolge und die Platzierung unterliegt keiner Einschränkung, jedoch ist dafür Sorge zu tragen, daß die "Daisy-chain" für Interrupt und DMA nicht unterbrochen ist.
Das MUX-KE unterstützt die 18-BIT-Adressierung in vollem Umfang.
Bei der Installation in 22 Bit-Systemen müssen die höchst- wertigen 4 Bit extern zur Verfügung gestellt werden. Beim QU68000-System wird dies durch den Prozessor selbst unterstützt. Auf der Backplane muß eine entsprechende Verdrahtung zusätzlich installiert werden.

6.2 Betriebssystem RSX11-M

Bevor ein System mit MUX-KE in Betrieb genommen werden kann, muß ein Betriebssystem vorliegen, in welchem der DH11-Treiber integriert ist. Bei der Generierung des Betriebssystems sind dabei folgende Angaben zu machen(oktal):

Adresse DH11:
160020

Vektoradresse DH11:
340

Anzahl der Schnittstellen:

bis max. 16 angebbar(möglichst nur so viele, wie auch physikalisch, d.h. DLV-Module, vorhanden sind)

6.3Betriebssystem MUNIX

Um das MUX-KE unter Unix einsetzen zu können, müssen drei Voraussetzungen geschaffen werden:

a) Generierung eines neuen Unix Kerns
b) Einrichtung passender special files für die Terminals
c) Ausgabe der angeschlossenen Terminals zur Generierung eines Logins im Multiuser-Mode.

a) Das neue Unix System muß den DH-Treiber enthalten und die eventuell vorhandene DMA-Verdrahtung berücksichtigen (siehe HW-Installation). Als Hilfsmittel zur Systemgenerierung steht das /etc/newconf - Kommando zur Verfügung. Es erfragt interaktiv welche Treiber einzubinden sind:
Die Zeile
"DH11 Terminal Multiplexer (MUXKE)"
ist mit ja zu beantworten. Besitzt der Rechner eine DMA-Verdrahtung, wenn nicht nur dmafähige 22-Bit-Controller verwendet werden, ist die Zeile:
"additional wiring on the backplane for DMA"
zu bejahen und die richtige Zuordnung zwischen Steckplatz der Controller im Rahmen und DMA-Extension-Register zu treffen.

Die letzte Frage
" Only 22 bit controller on DMA extension reg.0"
ist auf jeden Fall zu beantworten.

Newconf generiert nun einen neuen Kern und legt ihn als nunix ab.

b) Bevor mit nunix in den Multi-User-Mode hochgefahren wird, müssen die special files, die zum DH-Treiber gehören, eingerichtet sein.

Im Directory /dev ist ein makefile vorhanden, das einen Absatz zur Generierung der special files für Terminals, die vom DH-Treiber bedient werden, enthält. Mit dem Aufruf
make dh
werden diese eingerichtet.

c) In der Datei /etc/ttys existiert für jedes Terminal, das angeschlossen werden könnte, eine Zeile. Bei den tatsächlich angeschlossenen Terminals muß in der zugehörigen Zeile in der 1. Spalte eine 1 stehen, damit beim Hochfahren in den Multi-User-Mode ein login generiert wird.