

Data-Routing Funktionsbeschreibung

Multiplexer für 16-Bit Dateneingabe

März 1995 K. Huber, Strahlencentrum Univ. Gießen
Version 11.Oct.2024

Inhaltsverzeichnis

1	Einleitung	1
2	Aufbau und Funktion	2
2.1	Aufbau des Routing	2
2.2	Überrahmen	4
3	Standardkomponenten	5
3.1	Routing-Steuerung (RST)	5
3.1.1	Aufgabe	5
3.1.2	Bedienungselemente, Anzeigen	5
3.2	Rechneranpassung (RAP)	7
3.2.1	Aufgabe	7
3.2.2	Bedienungselemente, Anzeigen	7
3.3	Interface-Steuerung (IFS)	8
3.3.1	Aufgabe	8
3.3.2	Bedienungselemente, Anzeigen	8
3.4	Kopplungskarte (KPL)	11
3.4.1	Aufgabe	11
3.4.2	Funktion	11
3.4.3	Bedienungselemente	12
3.5	Test-Datenquelle (TST)	14
3.5.1	Aufgabe	14
3.5.2	Bedienungselemente	14
3.6	Bus-Signal Anzeige (BUS)	15
3.7	Routing-Funktionsüberwachung (Alarm)	15
3.8	Totzeitanzeige	15
4	Standard-Interface-Karten	16
4.1	16-Bit-Digitaleingabe (Laben-ADC-Interface)	16
4.1.1	Aufgabe, Funktion	16
4.1.2	Signalbeschreibung	16
4.2	Silena-ADC-Interface	18
4.2.1	Aufgabe, Funktion	18
4.2.2	Signalbeschreibung	18
4.3	Uhr/Zähler (UHR)	20
4.3.1	Aufgabe	20
4.3.2	Bedienungselemente	20
4.3.3	Beispiel: Lebensdauerermessung mit Totzeitkorrektur	21
4.4	Up/Down-Zähler (U/D-ZLR)	22
4.4.1	Aufgabe	22
4.4.2	Funktion	22
4.4.3	Eingänge (BNC-Buchsen)	22

4.5	12-Bit ADC (AD574)	24
4.5.1	Aufgabe	24
4.5.2	Bedienungselemente	24
4.6	Zählerbatterie (ZLRS, ZLRx)	26
4.6.1	Aufgabe	26
4.6.2	Funktion	26
4.6.3	Zählerkarten	27
4.6.3.1	ZLR1 - Vierfachzähler mit Überlaufregistrierung (TTL-Eingänge)	27
4.7	Messung der Breiten- und Abstandsverteilung von Signalen (BAV) ..	28
4.7.1	Aufgabe, Funktion	28
4.7.2	Bedienungselemente	28
4.8	Messung der Abstandsverteilung von Signalen (SAV)	29
4.8.1	Aufgabe	29
4.8.2	Funktion	30
4.8.3	Bedienungselemente	30
4.8.4	Beispiele für Abstandsverteilungen	31
4.8.5	Tabellen	34
5	Spezielle Interface-Karten	35
5.1	Elektron-Positron-Spaltfragment-Winkelverteilungs-Experiment (K5) ..	35
5.2	Ion-Ion-Stoß-Experiment (Labor 016)	35
5.3	Elektron-Ion-Stoß-Experiment (Labors 017, SI)	35
6	Zählratenstatistik	36
7	Technische Details	37
7.1	Routing-Bus	37
7.1.1	Routing-Bus Signale	37
7.1.2	Routing-Bus Abschluss	40
7.2	Komponenten und Schnittstellen	42
7.2.1	Datenquellen <-> Quellen-Interfaces	42
7.2.2	Quellen-Interface <-> Interface-Steuerung	42
7.2.3	Interface-Steuerung	45
7.2.4	Interface-Steuerung <-> Routing-Steuerung	46
7.2.5	Routing-Steuerung	46
7.2.6	Routing-Steuerung <-> Rechneranpassung	53
7.2.7	USB Anpassung	55
7.2.7.1	FT2232H Mini Modul (USB)	55
7.2.7.2	Rechneranpassung (USB)	56
7.2.8	VME Anpassung	57
7.2.8.1	Rechneranpassung (VME)	57
7.2.8.2	Rechneranpassung <-> Rechner-Interface (VME)	59
7.2.8.3	Rechner-Interface (VME)	64
7.3	Schaltungsunterlagen	67
7.3.1	Board Interface-Steuerung	67

7.3.2	Board Routing-Steuerung	67
7.3.3	Board Rechneranpassung	69
7.3.4	Board Rechner-Interface	69
8	Oldies	70
8.1	Alte Routing-Steuerung	70
8.1.1	Funktion	70
8.1.2	Bedienungselemente, Anzeigen	71
8.1.3	Schnittstellen	72
8.1.4	Routing-Bus	73
8.2	Alte Rechneranpassungen	83
8.2.1	PDP11 - DRQ11-CA - Anpassung	83
8.2.2	TR86 - CALAS - Anpassung	84

1 Einleitung

Zur Datenerfassung und Experimentsteuerung existieren zwei verschiedene Rechner-Interfaces, das **Data-Routing** und das **Control-Routing**, so genannt nach den Aufgaben, zu denen sie im Wesentlichen eingesetzt werden:

- Zur schnellen Erfassung von großen Datenmengen wird vorzugsweise das **Data-Routing** verwendet. Es ermöglicht eine schnelle Dateneingabe von bis zu 8 Eingabekanälen. Eine Datenausgabe ist damit nicht möglich.
- Das **Control-Routing** hingegen wird hauptsächlich zur Steuerung des Experimentes eingesetzt. Es erlaubt die Ein- und Ausgabe von einzelnen Datenworten, wobei bis zu 8 Module mit je 8 Registern, also maximal 64 Register, adressierbar sind. Ferner ist eine Interrupt-Eingabe vorgesehen.

Die vorliegende Funktionsbeschreibung befasst sich ausschließlich mit dem **Data-Routing**. Es wird im folgenden oft abgekürzt als **Routing** bezeichnet. Für das **Control-Routing** existiert eine eigene Beschreibung.

Das Kapitel **Aufbau und Funktion** gibt einen Überblick über den Aufbau des Data-Routing. Das Kapitel **Standardkomponenten** enthält die Aufgabenbeschreibungen und Bedienungsanleitungen der Standardkarten des Routing.

Das Kapitel **Standard-Interface-Karten** enthält die Aufgabenbeschreibungen und Bedienungsanleitungen der Routing-Karten, die allgemein bei Experimenten zum Einsatz kommen.

Das Kapitel **Spezielle Interface-Karten** enthält eine Aufstellung der Routing-Karten, die für einzelne Experimente entwickelt wurden. Ihre Beschreibung findet man i.a. in den Unterlagen der Experimente, bei denen sie eingesetzt werden.

Das Kapitel **Technische Details** gibt technische Detail-Informationen über die Komponenten und deren Schnittstellen.

Im Kapitel **Oldies** sind überholte Beschreibungen gesammelt.

2 Aufbau und Funktion

2.1 Aufbau des Routing

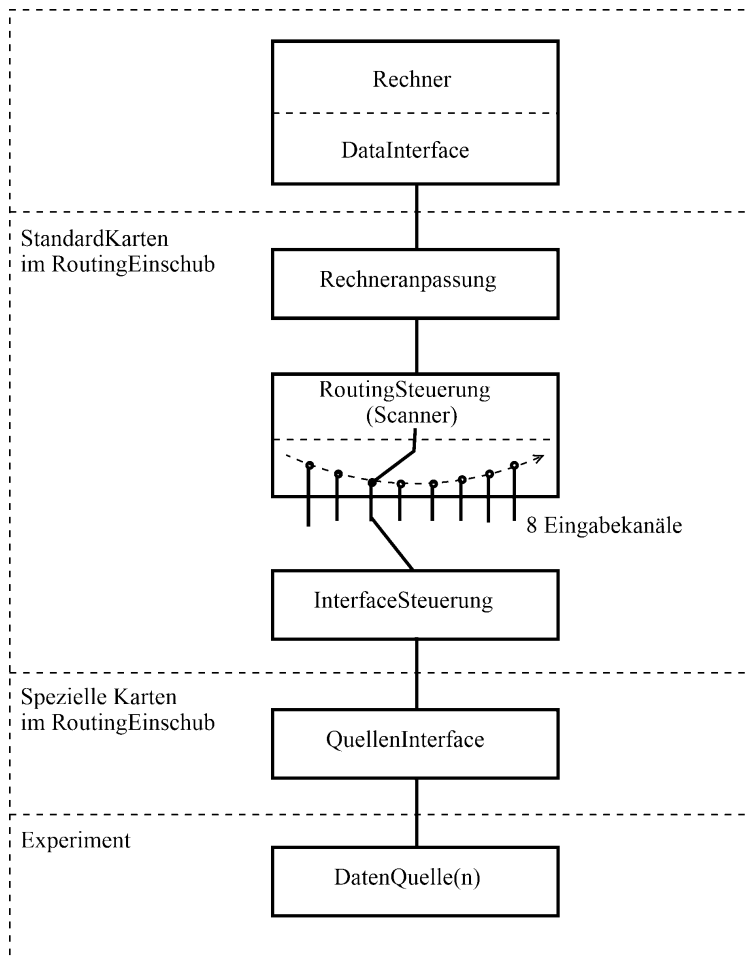


Abb. 2.1.0.1 Dateneingabe mit dem Data-Routing-System

- Rechner**
 Das Data-Routing kann an jeden Rechner mit geeignetem I/O-Port angeschlossen werden. Realisiert wurden bisher Anschlüsse an TR86-Rechner, PDP11-Rechner und VME-Systeme. Zur Zeit ist das Data-Interface auf einem IP-Board (Industrie Pack) implementiert, so dass es an jeden Rechner angeschlossen werden kann, für den ein IP-Board-Carrier existiert (VME, PC usw.).
- Data-Routing**
 Das Data-Routing ist in einem 19"-Überrahmen mit 20 Steckplätzen für Europakarten untergebracht.
- Rechneranpassung**
 Diese Karte fügt den 16 Bit Eingabedaten Quellenkennungs- und Status-Bits hinzu, damit die Software Herkunft und Status der Eingaben unterscheiden kann und speichert

sie in einem 2048 Worte tiefen und 3 Byte breiten Fifo zwischen. Ferner wickelt sie das Protokoll für die Übertragung der Daten über ein 50-poliges Kabel zum Rechner ab.

- **Routing-Steuerung**

Diese Karte koordiniert die Dateneingabe der 8 Eingabekanäle des Routing mittels zweier 16-fach Scanner mit zwei verschiedenen Prioritäten. Jedem Eingabekanal sind zwei Scanner-Schritte zugeordnet, wovon bei Einsatz der Standard-Interface-Steuerung einer zur Eingabe der Lifetime und der andere zur Eingabe der Daten verwendet wird. Die Wahl der Priorität bleibt der Datenquelle überlassen, zur Anpassung der Dateneingabe an die experimentellen Anforderungen. Die Beschränkung auf max. 8 Eingabekanäle in einem Routing wird relativiert durch die Möglichkeit über einen Kanal mehrere Datenquellen zu bedienen.

- **Interface-Steuerungen**

Zur Erleichterung des Anschlusses von Datenquellen wurde ein standardisiertes Interface entwickelt, das wenn möglich eingesetzt werden sollte. Zum Anschluss der Datenquellen ist dann zwar noch ein spezielles Quellen-Interface notwendig, dessen Aufbau im allgemeinen jedoch trivial ist.

- **Quellen-Interfaces**

Die Quellen-Interfaces setzen die vom Experiment kommenden Signale in digitale Daten um, die vom Routing übertragen werden können. Es gibt hier eine Reihe standardisierter Karten wie Zähler, Uhr, ADCs, ADC-Interfaces usw. und viele spezielle Entwicklungen, angepasst an die Anforderungen von Experimenten. So gibt es z.B. eine Karte, die unter zeitkritischen Bedingungen einen ganzen Satz von Messdaten aufnimmt und in einem FIFO zwischenspeichert, bis sie vom Routing übertragen werden.

- **Datenquellen am Experiment**

Die Datenquellen sind, abgesehen von Lifetime-Messung und ähnlichen 'Hilfsquellen', beim Experiment angesiedelt. Bei der Planung eines Experimentes sollte man sich möglichst an vorhandenen Quellen-Interfaces orientieren. Dies reduziert den Entwicklungsaufwand und ermöglicht im Falle eines Defektes den Austausch von Karten.

2.2 Überrahmen

Das Data-Routing ist in einem 19"-Überrahmen untergebracht mit 20 Steckplätzen für Europakarten (Abb. 2.2.0.1). Es wird der gleiche Überrahmen wie für das Control-Routing verwendet. In Ausnahmefällen können Data-Routing und Control-Routing im gleichen Überrahmen untergebracht werden, wenn die Bus-Verdrahtung in der Mitte durchtrennt wird und für die zweite Hälfte ein Bus-Abschluss nachgerüstet wird.

Die Routing Back-Plane enthält 42 allgemeine Bus-Leitungen für die Routing-Steuerung und 11 Privat-Bus-Leitungen zur Kommunikation benachbarter Karten.

Außer der Spannungsversorgung sind alle Schaltungskomponenten auf steckbaren Karten untergebracht.

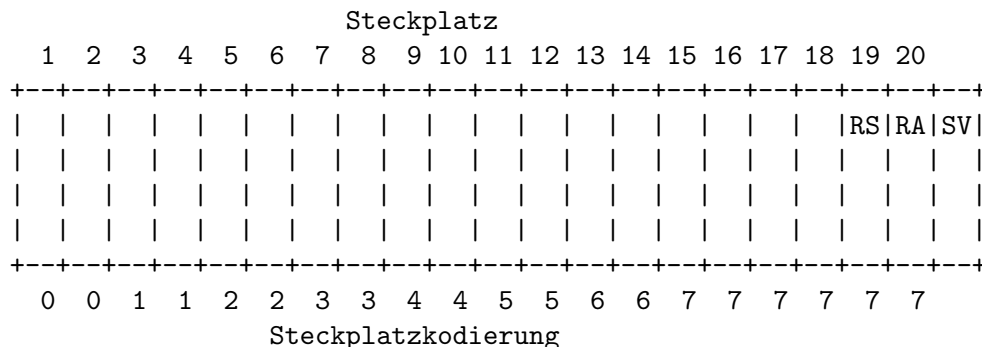
Als Steckverbindung zum Routing-Bus werden 64-polige VG-Stecker verwendet. Die Anschlüsse der VG-Leisten sind im Überrahmen zum Teil als durchgehender Bus verdrahtet (allg. Bus: 1a,1c,...,21c) und zu einem anderen Teil bestehen Verbindungen zu benachbarten Steckerleisten (Privat-Bus: 22a/c,...,32a/c) (Siehe Abschnitt 7.1 [Routing-Bus], Seite 37.). Ferner ist für jede Leiste von links beginnend eine Steckplatzkodierung von 0-7 verdrahtet, wobei jedoch jeweils zwei benachbarte Leisten die gleiche Kodierung haben. Die vier überbleibenden Steckplätze am rechten Ende erhalten alle die Kodierung 7. Sie sind vorzugsweise für die Rechner-Anpassung und Routing-Steuerung zu benutzen, da der Bus-Abschluss sich am linken Ende des Überrahmens befindet. Am äußersten rechten Ende ist die Netzkarte mit Netzschalter und Betriebsspannungsanzeigen fest installiert.

Die Frontplattenbreite für eine Steckkarte ist üblicherweise 20 mm, es stehen jedoch auch 40 und 50 mm Frontplatten zur Verfügung.

Achtung:

Es wird dringend empfohlen, die Frontplatten der Steckkarten mit dem Überrahmen zu verschrauben zur Vermeidung von Betriebsstörungen. Bitte die Schrauben nicht gewaltsam anziehen, da dies zur Zerstörung der Gewinde im Überrahmen führt. Schrauben von min. 10mm Länge verwenden, sonst besteht ebenfalls die Gefahr der Zerstörung der Gewinde.

Abb. 2.2.0.1 Routing-Überrahmen



- RS = Routingsteuerung
- RA = Rechneranpassung
- SV = Spannungsversorgung

3 Standardkomponenten

3.1 Routing-Steuerung (RST)

3.1.1 Aufgabe

Die Routing-Steuerung koordiniert die Eingabe der Daten von den bis zu acht anschließbaren Eingabekanälen. Ein Eingabekanal besteht aus den drei funktionellen Gruppen:

- Interface-Steuerung
- Datenquellen-Interface
- Datenquellen

Siehe Abschnitt 2.1 [Aufbau des Routing], Seite 2.

Siehe Abschnitt 7.2 [Komponenten und Schnittstellen], Seite 42.

Die Routing-Steuerung ist mit zwei alternativ laufenden Scannern unterschiedlicher Priorität ausgestattet, mit denen die Eingabekanäle abgefragt werden. Dadurch besteht die Möglichkeit, einen oder mehrere Eingabekanäle mit Vorrang vor den anderen abfragen zu lassen. Dazu müssen diese Eingabekanäle einen Vorrangwunsch bei der Routing-Steuerung anmelden, sobald sie bereit sind ein Datenwort zu übertragen. Der jeweils ruhende Scanner behält seine Position bei, so dass er an der unterbrochenen Stelle weiterlaufen kann.

Um verkürzte Scanner-Umläufe zu ermöglichen, sind die Scanner-Adressen in vier Gruppen zusammengefasst, die in Einer-, Zweier- oder Vierer-Kombinationen durch einen Jumper aktiviert werden können. Die Scanner-Adressen werden stets in einer Reihenfolge durchlaufen in der sich höchstens 2 Bits gleichzeitig ändern, um Störungen auf dem Routing-Bus zu minimieren (Siehe Abschnitt 7.2.5 [Routing-Steuerung], Seite 46.).

Es ist zu beachten, dass die verschiedenen Eingabekanäle von der Routing-Steuerung nicht nach einem 'first in - first out' Verfahren bearbeitet werden, sondern durch einen Scanner reihum abgefragt werden. Dies kann zur Folge haben, dass die einzelnen Eingabekanäle nicht in der Reihenfolge zum Rechner übertragen werden, in der sie zur Übertragung bereit waren. Die damit zusammenhängenden Probleme der Verkoppelung von Datenquellen z.B. bei Koinzidenzexperimenten sind jedoch berücksichtigt (Siehe Abschnitt 3.4 [Kopplungskarte (KPL)], Seite 11.).

Ein Stop der Routing-Steuerung kann vom Rechner her erfolgen oder über einen Schalter auf der Frontplatte. Beide bewirken ein Anhalten der Daten-Scanner. Ferner wird der Stop den Eingabekanälen mitgeteilt und über die Interface-Steuerung den Datenquellen.

Die Routing-Steuerung arbeitet eng mit der Rechneranpassungs-Karte zusammen und steckt deshalb direkt links neben ihr.

3.1.2 Bedienungselemente, Anzeigen

Schalter: Run/Stop

In Stellung *Stop* wird die Abfrage der Eingabekanäle angehalten durch Stoppen des Daten-Scanners. An die Eingabekanäle geht ein Stopsignal. Der Schalter ist disjunktiv (ODER) verknüpft mit dem Stop vom Rechner.

Anzeige: Run

Wenn weder vom Rechner noch durch den Schalter *Run/Stop* ein Stop erfolgt, d.h. die Routing-Steuerung läuft, so leuchtet die Anzeige *Run*.

Anzeige: Ready

Solange die Routing-Steuerung das Data Ready (*ReadyB*) eines Eingabekanals bearbeitet, leuchtet die Anzeige *Ready*.

Anzeige: Vorrang

Solange die Routing-Steuerung im Vorrangmodus arbeitet, leuchtet die Anzeige *Vorrang*.

Jumper: Group 0 1 2 3

Selektion von Scanner-Gruppen 0 1 2 3. Erlaubte Kombinationen von Gruppen sind 0, 1, 2, 3, 0-1, 0-2, 1-3, 2-3, 0-1-3-2. Bei unzulässigen Kombinationen, und falls kein Jumper gesetzt ist, wird 0-1-3-2 ausgeführt.

3.2 Rechneranpassung (RAP)

3.2.1 Aufgabe

Die Rechneranpassungs-Karte hat die Aufgabe aus den Daten- und Kennungsbits auf dem Bus ein Datenwort zusammenzustellen, es in Fifos zwischenzuspeichern und dann an den Rechner zu übertragen. Darüberhinaus enthält die Rechneranpassungs-Karte Sender, Empfänger und eventuell Pegelwandler für die Dialog-Signale zwischen dem Rechner und dem Routing.

Die Verbindung zum Rechner erfolgt über ein max. 40m langes 50-poliges Flachbandkabel. Kurze Kabellängen sind zu bevorzugen da die Länge in die Übertragungsgeschwindigkeit eingeht und kürzere Kabel störungssicherer sind.

Für Testzwecke am Routing kann das 50-polige Verbindungskabel zum Rechner abgezogen werden. Die Rechneranpassung nimmt dann dem Routing die Daten mit höchstmöglicher Geschwindigkeit ab (und wirft sie weg).

Bisher wurden drei Anpassungen an verschiedene Host-Rechner realisiert:

- VME - Industry Pack - Anpassung
- PDP11 - DRQ11-CA - Anpassung
- TR86 - CALAS - Anpassung

Aktuell ist die VME - Anpassung, die zusammen mit dem rechnerseitigen Interface eine Schnittstelle zu dem Industry Pack Standard darstellt. Für diesen IP-Standard werden Carrier-Boards für eine Anzahl weiterer Plattformen (z.B. PC) angeboten, so dass ein Anschluss des Data-Routing an einen anderen Rechnertyp ohne großen Aufwand möglich ist. Die TR86- und PDP11-Anpassungen wurden inzwischen ausgemustert. Ihre Beschreibung finden Sie bei den 'Oldies'.

Die sogn. Silena-Änderung wurde in die aktuelle VME-Rechneranpassung nicht mit übernommen, sondern es wird vorausgesetzt, dass eine entsprechende Änderung auf **allen** Silena-Interface-Karten vorgenommen ist.

Die Rechneranpassungs-Karte arbeitet sehr eng mit der Routing-Steuerung zusammen und steckt deshalb direkt rechts neben ihr.

3.2.2 Bedienungselemente, Anzeigen

Die Rechneranpassungskarte besitzt keine Bedienungselemente. Auf dem Board (von außen nicht sichtbar) befindet sich eine grüne LED zum Anzeigen einer aktiven Verbindung mit dem Rechner und eine rote LED zum Anzeigen eines Fifo-Errors.

3.3 Interface-Steuerung (IFS)

3.3.1 Aufgabe

Die Interface-Steuerung stellt eine standardisierte Karte zwischen der Routing-Steuerung und den Datenquellen dar. Im Zusammenspiel mit der Routing-Steuerung wickelt sie die Routing-Logik ab und bietet den Datenquellen eine bequeme Schnittstelle, die den Anschluss von Datenquellen wesentlich erleichtert. Ferner ist sie zur Lifetime-Messung, zur Eingabe von Daten im Vorrangmodus und zur Eingabe von gekoppelten Daten (Koinzidenz-Experimente) vorbereitet. Zum Anschluss von Datenquellen sind, je nach Komplexität der Aufgabe, noch ein oder mehrere Quellen-Interface-Karten notwendig, die im einfachsten Falle nur Pegelwandler und Bus-Leitungstreiber enthalten.

Zur Verständigung mit den Quellen-Interface-Karten baut die IFS-Karte nach rechts einen Privat-Bus (IFS-Bus) auf. Die Quellen-Interface-Karten müssen deshalb stets rechts von der IFS-Karte stecken. Die Frontplatten der so miteinander kommunizierenden Karten müssen fest mit dem Überrahmen verschraubt sein, um eine gute Masseverbindung von den Karten zur Frontplatte herzustellen. Fehlt die Verschraubung so kann dies zu schwerwiegenden Betriebsstörungen führen, weil die Karten wie die Kondensatorplatten eines gedämpften Oszillators gegeneinander schwingen.

Es existieren zwei Varianten (IFS und IFS1) der Interface-Steuerung, die sich in ihren Funktionen nicht wesentlich unterscheiden. Bei der älteren (IFS) sind die Konfigurationsbrücken als Lötbrücken (statt Steckbrücken) ausgeführt und unbeschriftet und auf der Frontplatte fehlt die Anzeige für den Vorrangbetriebsmodus.

3.3.2 Bedienungselemente, Anzeigen

Stop/Run/LFT (Schalter)

Schalter für den Betriebsmodus.

- **Stop:** Stop Datenübertragung
Die angeschlossene Quelle wird nicht abgefragt. Sie erhält außerdem ein Stop-Signal für Normierungszwecke.
- **Run:** Normalbetrieb der Datenquellen
Die Lifetime-Messung ist abgeschaltet.
- **LFT:** Betrieb mit Lifetime-Messung
Intern wird ein Lifetime-Signal ($LFTQ$) generiert, das sich konjunktiv (UND) zusammensetzt aus der negierten Busy-Meldung ($BusyQ\$) der Datenquelle, je nach Verdrahtung auf der IFS-Karte dem Eingang $GateQ$ der IFS-Karte und im Kopplungsmodus $K2$ der Lifetime-Meldung $LFTK$ der Kopplung (Siehe Abschnitt 3.4 [Kopplungskarte (KPL)], Seite 11.):

$$LFTQ = BusyQ\ * GateQ(Optional) * (K2 * LFTK\)$$

Die Option wird eingeschaltet durch Stecken der Brücke $LFT\ = By+G$ statt By . Immer, wenn dieses Signal $LFTQ$ ansteht wird der interne Lifetime-Zähler mit 1MHz Takt hochgezählt. Sobald auf diese Weise 1s bzw. 0.1s (Brücke: 1s/0.1s) Lifetime zusammengekommen sind, wird ein Datenwort übertragen, das zur Unterscheidung neben der normalen Quellenkennung noch das Lifetime-Bit (Siehe $\langle undefined \rangle$ [Rechneran-

passung], Seite (undefined).) gesetzt hat. Alle Daten-Bits sind null. Diese Funktion der IFS-Karte erlaubt totzeitkorrigierte Messungen.

Da die Totzeitsignale intern mit einem 1MHz-Takt abgefragt werden, kann es zu Fehlmessungen der Lifetime kommen, wenn die Totzeitsignale phasensynchron mit dem internen 1MHz-Takt und in der Größenordnung der Taktperiode sind. Für völlig vom internen Takt unabhängige Totzeitsignale sollten hingegen vom Schaltungsentwurf her auch kürzere als 1µs richtig verarbeitet werden, auf einer statistischen Basis, in dem sie mal einen Takt durchlassen und mal nicht (hab' dies aber bis jetzt noch nicht überprüft!).

Achtung: Aus Kostengründen sind nicht alle IFS-Karten mit den für die LFT-Messung notwendigen Bausteinen (SAJ141) bestückt.

Frei/K1/K2 (Schalter)

Sollen mehrere Datenquellen gekoppelt übertragen werden (z.B. bei Koinzidenz-Experimenten), so wird über diesen Schalter entschieden in welcher Weise die zugehörige Datenquelle an der Kopplung teilnimmt. Detaillierte Information ist der Beschreibung der Kopplungskarte zu entnehmen (Siehe Abschnitt 3.4 [Kopplungskarte (KPL)], Seite 11.).

- **Frei:** keine Kopplung
Die Datenquelle läuft frei, sie nimmt nicht an der Kopplung teil.
- **K1:** lose Kopplung
Die Datenquelle läuft frei, sie wird durch das von der Kopplungskarte angelieferte Kopplungssignal (*KoppK*) nicht auf- und zugegattet. Meldet die Datenquelle 'busy' während das Kopplungssignal ansteht, so wird in dem zugehörigen Datenwort bei der Übertragung das Kopplungs-Bit (Siehe (undefined) [Rechneranpassung], Seite (undefined).) gesetzt. Die Software kann dadurch unterscheiden, ob die Quelle koinzident mit anderen Quellen ein Ereignis registriert hat.
- **K2:** strenge Kopplung
Die Datenquelle wird nur aufgegattet, wenn das Kopplungssignal ansteht. Im übertragenen Wort ist immer das Kopplungs-Bit gesetzt.

GateQ: (TTL-Eingang)

Über diesen Eingang kann die Datenquelle auf- und zugegattet werden. Bei Betrieb im Modus *K2* erfolgt eine Verknüpfung mit dem Signal *GateK* und dem Kopplungssignal (siehe *GateK*).

High oder offener Eingang: Gate offen

GateK: (TTL-Eingang)

Über diesen Eingang kann, um spezielle Zeitbedingungen zu realisieren, das Kopplungssignal (siehe Modus *K1*, *K2*) extern zugeführt werden. Es wird mit dem internen Kopplungssignal (*KoppK*), das von der Kopplungskarte geliefert wird, konjunktiv (UND) verknüpft. Für den Kopplungsmodus *K2* wird das Ergebnis noch mit dem Eingang *GateQ* konjunktiv verknüpft und als Gate-Signal (*GateSQ*) an die Datenquelle übergeben.

High oder offener Eingang: Gate offen

$$GateSQ = GateQ * (K2 \setminus + K2 * KoppK * GateK)$$

Out: (TTL-Ausgang)

Je nach Konfiguration der IFS-Karte (Steckbrücke: Gate/By) kann über diesen Ausgang das Busy-Signal der Datenquelle oder das Gate-Signal für die Datenquelle beobachtet werden.

High: Quelle Busy/Gate offen

Run bzw. R/S: (LED-Anzeige)

Run/Stop-Anzeige

LED leuchtet: IFS-Karte läuft

B bzw. By: (LED-Anzeige)

Busy Anzeige

LED leuchtet: Datenquelle meldet Busy

VR: (LED-Anzeige; nur IFS1)

Vorranganzeige

LED leuchtet: Die Datenquelle arbeitet im Vorrangmodus

4 2 1: (LED-Anzeige)

Anzeige der Steckplatzadresse.

Zur Unterscheidung der verschiedenen Datenquellen wird dem Datenwort die Steckplatzadresse als Quellenkennung mitgegeben. Da im allgemeinen beim Start eines Messprogrammes die Quellenkennung angegeben werden muss, wird die Steckplatzadresse von der IFS-Karte angezeigt.

LED leuchtet: Bit der angegebenen Wertigkeit ist gesetzt.

1s / 0.1s: (Jumper)

Zeitbasis für die Übertragung des Lifetime-Datenwortes (s.o.).

Out = Gate / Busy: (Jumper)

Beschaltung der 'Out' BNC-Buchse mit Gate- oder Busy-Signal (s.o.).

LFT = Busy / Busy & Gate: (Jumper)

Option für die Lifetime-Messung (s.o.).

3.4 Kopplungskarte (KPL)

3.4.1 Aufgabe

Bei manchen Experimenten (z.B. Koinzidenzexperimente) werden zu einem Ereignis mehrere Datenworte erzeugt, die nach ihrer Übertragung für die Software als zusammengehörig erkenntlich sein müssen. Diesem Zweck dient die Kopplungskarte.

3.4.2 Funktion

Die Kopplungskarte arbeitet wie eine normale Datenquelle, d.h. sie benötigt eine Standard-Interface-Steuerung (IFS-Karte). Der Schalter *Frei/K1/K2* auf der IFS-Karte wird dabei außer Betrieb gesetzt, so dass die Kopplungskarte im Modus "*Frei*" läuft.

Die Kopplungskarte ist Ausgangspunkt für einen Privat-Bus, den Kopplungs-Bus, der sich nach links fortsetzt bis entweder ein Steckplatz frei ist, eine weitere Kopplungskarte auftritt, oder aber eine Karte steckt, die den Kopplungs-Bus unterbricht.

Die Kopplung wird gestartet über den BNC-Eingang Start auf der Kopplungskarte. Nach erfolgtem Start geht die Kopplung sofort in den Zustand Busy (*BusyK*). Das Busy steht solange an wie irgendeine der gekoppelten Quellen Busy (*BusyQi*) an die Kopplung meldet. Die Übertragung des Kopplungswortes trägt ihr Busy (*BusyQK*) wie eine normale Quelle bei. Zur Überbrückung der Reaktionszeiten der verschiedenen gekoppelten Quellen geht die Kopplung mit dem Start in eine interne Wartephase (*Wait*) einstellbarer Dauer. Diese Wartephase trägt ebenfalls zum Busy der Kopplung bei:

$$BusyK = Wait + BusyQK + Sum(i, BusyQi)$$

Eine eindeutige Zusammenfassung der gekoppelten Daten ist nur gewährleistet, wenn *BusyK* lückenlos ansteht bis alle Quellen ihre Daten registriert und abgeliefert haben. Durch geeignete Einstellung der Wartephase sollte dies immer möglich sein.

Das Signal *Wait* steht den gekoppelten Quellen über den Kopplungs-Bus unter dem Namen Kopplungssignal (*KoppK*) zu Synchronisationszwecken (Aufgaten der Quellen) zur Verfügung (Siehe Abschnitt 3.3 [Interface-Steuerung (IFS)], Seite 8.).

Je nach eingestelltem Start-Modus unterliegt der Start der Kopplung unterschiedlichen Bedingungen:

- Im Normalmodus erfolgt der Start nur, wenn keine der am Kopplungs-Bus hängenden Quellen oder die Kopplungskarte selbst Busy meldet, d.h. wenn $BusyK=0$.
- Im weniger strengen Spezialmodus kann ein Start immer erfolgen, wenn der vorhergehende Kopplungszyklus beendet ist. Ein Kopplungszyklus beginnt mit dem Start und endet sobald keine Busy-Meldungen mehr anstehen. Spätere Busy-Meldungen verhindern im Gegensatz zum Normalmodus einen erneuten Start nicht. Der Spezialmodus ist für Fälle gedacht, in denen die Datenquellen bereits Busy melden bevor der Kopplungsstart erfolgt.

Nach erfolgtem Start meldet die Kopplungskarte einen vorrangigen Bearbeitungswunsch bei der Routing-Steuerung an zur Übertragung eines Datenwortes, dem Kopplungswort. Durch entsprechendes Timing auf den IFS-Karten ist sichergestellt, dass das Kopplungswort auf jeden Fall vor den Datenwörtern der gekoppelten Quellen übertragen wird. Diese folgen

in beliebiger Reihenfolge je nach Bearbeitungszeit in den Quellen und Stand des Routing-Scanners. Nicht alle Datenquellen müssen notwendigerweise Daten übertragen, sondern nur die, die ein Ereignis erkannt haben. Die Synchronisierung der Software erfolgt durch das Kopplungswort am Anfang eines Blockes von gekoppelten Daten.

Solange sich die Kopplung im Zustand 'Busy' befindet, werden alle abgewiesenen Startversuche im 'Pile-up-Zähler' registriert. Mit jeder Datenübertragung wird im Kopplungswort der Zählerstand übertragen und gleichzeitig gelöscht. Falls mehr als 255 Pile-ups auftreten, bleibt der Zähler bei 255 stehen.

Das Kopplungswort ist wie ein normales Datenwort aufgebaut. Das Kopplungs-Bit ist null. In den Daten-Bits 0-7 steht der Inhalt des Pile-up-Zählers.

Für jede am Kopplungs-Bus angeschlossene Datenquelle kann der Kopplungsmodus individuell gewählt werden über den Schalter *Frei/K1/K2* auf der IFS-Karte.

Im Modus *Frei* nimmt die Quelle an der Kopplung nicht teil, sie läuft frei. Das von der Kopplung gelieferte Kopplungssignal (*KoppK*) wird nicht berücksichtigt und das Busy der Quelle wird nicht an die Kopplung gemeldet.

Im Modus *K1* läuft die Quelle im wesentlichen frei. Falls jedoch der Beginn der Busy-Meldung der Quelle in das von der Kopplungskarte vorgegebene Kopplungssignal fällt, wird in der Kennung des Datenwortes das Kopplungs-Bit gesetzt. Auf diese Weise können z.B. koinzidente und nicht koinzidente Ereignisse mit dem gleichen ADC aufgenommen werden. Für Ereignisse, die das Setzen des Kopplungs-Bits zur Folge haben, wird das Busy der Quelle an die Kopplung gemeldet. Dies verhindert einen weiteren Start der Kopplung bevor die Quelle mit ihrer Bearbeitung völlig fertig ist und garantiert, dass Datenworte verschiedener Koinzidenzereignisse sauber getrennt werden. Für höhere Zählraten ist der Modus *K1* nicht gut geeignet, da mit wachsender Zählrate die Wahrscheinlichkeit zunimmt, dass eine der gekoppelten Quellen noch durch ein nicht koinzidentes Ereignis busy ist. Der Start der Kopplung würde in diesem Falle zwar erfolgen, aber das Datenwort der betreffenden Quelle würde bei der Koinzidenz fehlen.

Im Modus *K2* können die Quellen nur während der Dauer des Kopplungssignals Ereignisse registrieren. Das Kopplungs-Bit ist immer gesetzt, Busy wird stets an die Kopplung gemeldet. Es werden also nur koinzidente Ereignisse registriert. Ein nicht-koinzidenter Untergrund belastet das System nicht durch Totzeit.

Es ist zu beachten, dass durch die unterschiedliche Wirkungsweise des Kopplungssignals in den Modi *K1* und *K2* sich für das Kopplungssignal ein unterschiedliches Timing ergibt. Der Unterschied besteht darin, dass im Modus *K1* das Busy der Quelle während des Kopplungssignals zum Setzen des Kopplungs-Bits führt und im Modus *K2* die Quelle vom Kopplungssignal aufge gated wird, was offensichtlich dem Busy zeitlich vorausgehen muss.

3.4.3 Bedienungselemente

Start: (TTL-Eingang)

Durch die positive Flanke eines Signals an diesem Eingang wird die Kopplung gestartet, falls sie nicht gerade busy ist.

Int/Ext: (Schalter)

In Stellung **Int** wird die Dauer der Wartephase intern erzeugt. Sie ist durch ein Potentiometer

auf der Frontplatte im Bereich von etwa 1-100 us einstellbar.
 In Stellung **Ext** wird die Dauer der Wartephase durch das Startsignal vorgegeben (high: Wartephase).

Out: (TTL-Ausgang)

Über diesen Ausgang kann das Signal *Wait* der Wartephase abgenommen werden.

norm./spez.: (Steckbrücke auf der Karte)

Über diese Steckbrücke auf der Kopplungskarte kann der Startmodus (normal bzw. spezial) eingestellt werden.

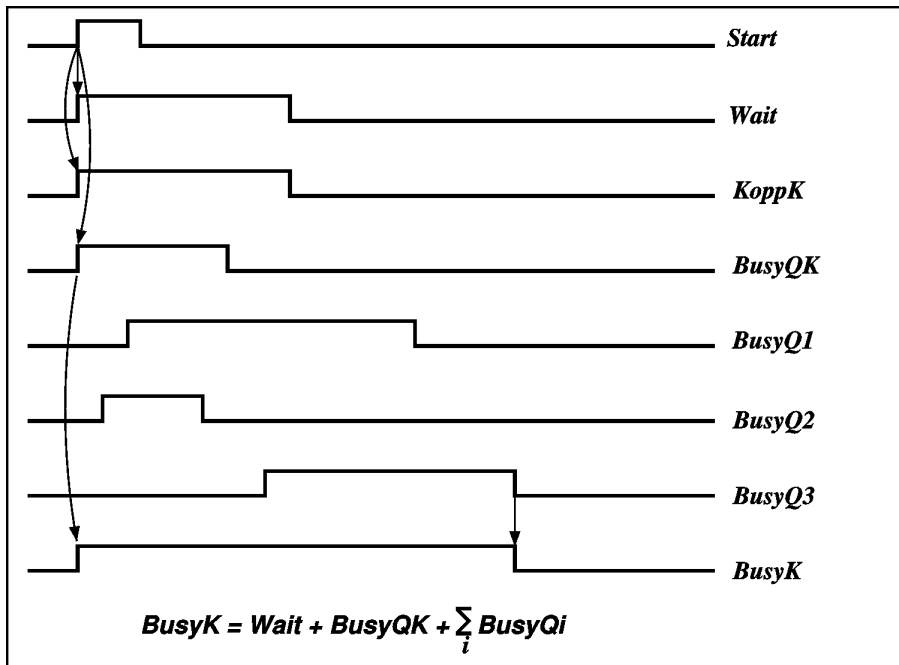


Abb. 3.4.3.1 Timing von gekoppelten Datequellen

3.5 Test-Datenquelle (TST)

3.5.1 Aufgabe

Die Testdatenquelle erzeugt 16-Bit Datenworte zu Testzwecken. Wahlweise liefert sie mit jedem Scannerzyklus ein Datenwort ab, die Daten werden also mit der maximal möglichen Geschwindigkeit übertragen, oder die Übertragung wird durch einen externen Takt gesteuert.

3.5.2 Bedienungselemente

set/count/reset: (Schalter)

- **set:** Alle 16 Datenbits sind gesetzt.
- **count:** Nach jeder Übertragung wird das Datenregister inkrementiert bzw. dekrementiert. Bei Überlauf wird mit 0 bzw. 65535 fortgesetzt.
- **reset:** Alle 16 Datenbits sind gelöscht.

ext/int: (Schalter)

- **ext:** Die Übertragung eines Datenwortes wird durch einen externen Takt angestoßen.
- **int:** Mit jedem Routing-Zyklus wird ein Datenwort übertragen.

prio/norm: (Schalter)

- **prio:** Die Übertragung erfolgt im Vorrang-Modus (Siehe Abschnitt 3.1 [Routing-Steuerung (RST)], Seite 5.).
- **norm:** Die Übertragung erfolgt im Normalmodus.

inc/alt/dec: (Schalter)

- **inc:** Im Modus 'Count' wird nach jeder Übertragung das Datenregister inkrementiert.
- **alt:** Im Modus 'Count' wird nach jeder Übertragung das Datenregister alternierend inkrementiert/dekrementiert.
- **dec:** Im Modus 'Count' wird nach jeder Übertragung das Datenregister dekrementiert.

Ext-Takt: (BNC-Buchse, TTL-Eingang)

Takteingang zum externen Anstoßen einer Übertragung (positive Flanke).

3.6 Bus-Signal Anzeige (BUS)

Die Anzeigekarte übernimmt vom Routing-Bus die Daten- und Kontroll-Bits und zeigt sie über LED's auf der Frontplatte an.

Anzeige:

D0 ...D15	Daten-Bits
frei	
frei	
frei	
frei	
Kopp-Bit	Kopplungs-Bit
LFT-Bit	Lifetime-Bit

3.7 Routing-Funktionsüberwachung (Alarm)

Es werden die beiden Scanner-Takte *TV* und *TN* der Routing-Steuerung (Siehe Abschnitt 7.2.5 [Routing-Steuerung], Seite 46.) überwacht, und falls sie länger als ca. 0.5s ausbleiben, ohne dass das Routing gestoppt ist, wird Alarm gegeben. Der Alarm wird beim nächsten Scanner-Schritt wieder aufgehoben.

Anzeige (rote LED): leuchtet bei Alarm

BNC-Ausgänge für High- und Low-Signale bei Alarm
(2.5kOhm nach +5V; 330Ohm nach 0V)

3.8 Totzeitanzeige

Diese Karte bildet ein zeitliches Mittel über ein digitales Totzeitsignal (TTL-Pegel) und zeigt die Totzeit auf einem Drehspulinstrument mit 100er-Skala an. Die Karte erwartet das Totzeitsignal auf der P-Bus-Leitung 28a von ihrer direkten linken Nachbarkarte. Diese muss deshalb auf 28c ein entsprechendes Signal verdrahtet haben. Für die üblichen Interface-Karten ist 28a-28c gebrückt, so dass das Signal *BusyQ* als Totzeitsignal zur Verfügung steht.

4 Standard-Interface-Karten

4.1 16-Bit-Digitaleingabe (Laben-ADC-Interface)

4.1.1 Aufgabe, Funktion

Die Digitaleingabe bietet eine universelle Schnittstelle zur Eingabe von 16 Daten-Bits. Sie ist so ausgelegt, dass Laben ADC's direkt angeschlossen werden können. Die Verbindung zur Datenquelle wird durch einen 37-poligen ERNI-Stecker hergestellt. Die Kabellänge sollte 1.5 m nicht überschreiten. Alle von der Datenquelle kommenden Leitungen sind mit 1 kOhm Pull-up-Widerständen versehen. Die Leitungen zur Datenquelle können mit max. 15 mA bei high Pegel und 24 mA bei low belastet werden.

Achtung:

Dieses Interface wurde früher auch zum Anschluss von Silena-ADCs verwendet. Durch einen Timer auf der alten Rechneranpassung war dafür gesorgt, dass *AcceptedB* mindestens 1us lang ist. Diese Verlängerung (Silena-Änderung) war notwendig, da der Silena-ADC eine (im Manual nicht spezifizierete) Zeit von ca. 0.5us für ein Reset benötigt. Diese Silena-Änderung wurde in die VME-Rechneranpassung nicht mit übernommen, deshalb ist dieses Interface nicht mehr zum Anschluss von Silena-ADCs geeignet. Bitte das spezielle Silena-ADC-Interface verwenden.

4.1.2 Signalbeschreibung

Mit **Stop** wird die Datenquelle gestoppt und gecleart.

Mit **Ready** zeigt die Datenquelle an, dass sie bereit ist ein Datenwort zu übergeben.

Mit **Request** wird die Datenquelle aufgefordert ihre Daten auf die Datenleitungen zu geben. Prinzipiell können die Daten immer an den Datenleitungen anstehen. Falls jedoch Pegeländerungen auf vielen Datenleitungen gleichzeitig auftreten, führt dies unweigerlich zu Störungsspitzen auf den Steuerleitungen, die zu Fehlfunktionen führen können (z.B. verfrühte Anmeldung von **Ready**).

Mit **Accepted** zeigt die Digitaleingabe an, dass die Daten übernommen wurden. Je nach Auslastung des Rechners kann das **Accepted** u.U. recht lange auf sich warten lassen. Normalerweise löscht die Datenquelle mit **Accepted** das anstehende **Ready**, notwendig ist dies jedoch nicht, falls beim nächsten Routing-Zyklus sofort ein weiteres Datenwort übertragen werden soll.

Mit **Busy** zeigt die Datenquelle an, dass sie beschäftigt ist mit der Erzeugung eines Datenwortes. **Busy** kann je nach Quelle mit der Vorder- oder Rückflanke von **Accepted** oder sonst wann verschwinden. Es wird bei einer eventuellen Totzeitmessung verarbeitet (Siehe Abschnitt 3.3 [Interface-Steuerung (IFS)], Seite 8.).

Da je Datenleitung ein Strom von ca. 5 mA fließt, ist für eine Masseleitung von ausreichendem Querschnitt zu sorgen für die Rückführung der $16 * 5 \text{ mA} = 80 \text{ mA}$.

Tab. 4.1.2.1 Schnittstelle zur Datenquelle

37 poliger ERNI-Stecker

Die nicht aufgeführten Anschlüsse sind unbelegt.

Signale, die mit '^' beginnen, werden von der Digitaleingabe erzeugt.

Signal	ERNI-Pin
Daten-Bit 0\	19
Daten-Bit 1\	18
Daten-Bit 2\	17
Daten-Bit 3\	16
Daten-Bit 4\	15
Daten-Bit 5\	14
Daten-Bit 6\	13
Daten-Bit 7\	12
Daten-Bit 8\	11
Daten-Bit 9\	10
Daten-Bit 10\	9
Daten-Bit 11\	8
Daten-Bit 12\	7
Daten-Bit 13\	6
Daten-Bit 14\	5
Daten-Bit 15\	4
Ready\	3
^Request	20
^Accepted	23
^Stop\	22
Busy\	2
^Gate	1
Masse	

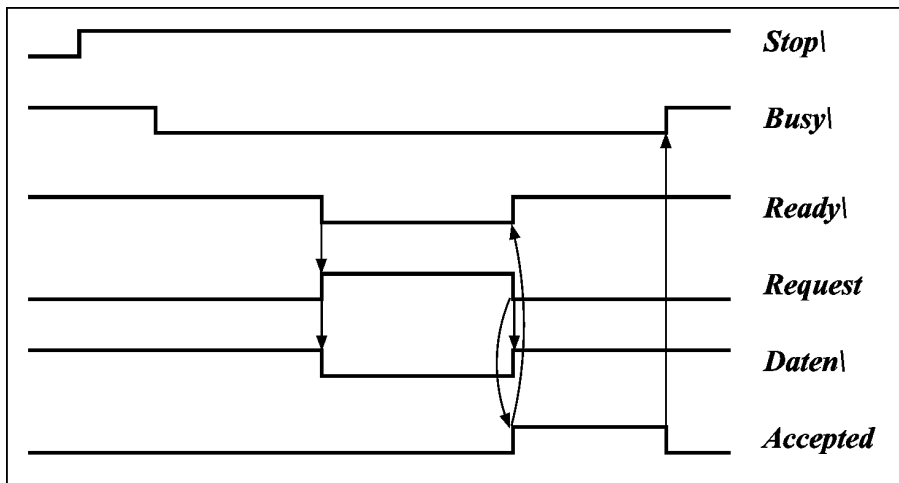


Abb. 4.1.2.2 Timing zwischen Digital-Eingabe und Datenquelle

4.2 Silena-ADC-Interface

4.2.1 Aufgabe, Funktion

Das Silena-ADC-Interface ist eine modifizierte 16-Bit-Digitaleingabe. Es berücksichtigt die spezielle Steckerbelegung des Silena-ADCs, so dass die Kabelverbindung mit einem 37-poligen Flachbandkabel hergestellt werden kann. Ferner berücksichtigt es den Umstand, dass der Silena-ADC ein unterschiedlich langes (bis zu 500ns, nicht im Manual spezifiziert) Accepted-Signal benötigt.

Selbstverständlich kann diese Interface-Karte auch als universelle Schnittstelle zur Eingabe von 16 Daten-Bits verwendet werden. Die Verbindung zur Datenquelle wird durch einen 37-poligen ERNI-Stecker hergestellt. Die Kabellänge sollte 1.5 m nicht überschreiten. Alle von der Datenquelle kommenden Leitungen sind mit 1 kOhm Pull-up-Widerständen versehen. Die Leitungen zur Datenquelle können mit max. 15 mA bei high Pegel und 24 mA bei low belastet werden.

4.2.2 Signalbeschreibung

Mit **Stop** wird der ADC gestoppt und gecleart.

Mit **Ready** zeigt der ADC an, dass er bereit ist ein Datenwort zu übergeben.

Request ist nicht verdrahtet, der Silena-ADC gibt deshalb seine Daten immer auf die Datenleitungen.

Mit **Accepted** zeigt das Interface an, dass die Daten übernommen wurden. **Accepted** steht so lange an bis das **Ready**-Signal verschwindet. Dieses Verhalten von **Accepted** ist notwendig, da der Silena-ADC ein im Manual nicht spezifiziertes, unterschiedlich langes (bis 500ns) **Accepted** benötigt.

Mit **Busy** zeigt der ADC an, dass er beschäftigt ist mit der Erzeugung eines Datenwortes. Es wird bei einer eventuellen Totzeitmessung verarbeitet (Siehe Abschnitt 3.3 [Interface-Steuerung (IFS)], Seite 8.).

Da je Datenleitung ein Strom von ca. 5 mA fließt, ist für eine Masseleitung von ausreichendem Querschnitt zu sorgen für die Rückführung der $16 * 5 \text{ mA} = 80 \text{ mA}$.

Tab. 4.2.2.1 Schnittstelle zum Silena ADC

37 poliger ERNI-Stecker

Die nicht aufgeführten Anschlüsse sind unbelegt.

Signale, die mit '^' beginnen, werden vom Interface erzeugt.

Signal	ERNI-Pin
Daten-Bit 0\	1
Daten-Bit 1\	2
Daten-Bit 2\	3
Daten-Bit 3\	4
Daten-Bit 4\	5
Daten-Bit 5\	6
Daten-Bit 6\	7
Daten-Bit 7\	8
Daten-Bit 8\	9
Daten-Bit 9\	10
Daten-Bit 10\	11
Daten-Bit 11\	12
Daten-Bit 12\	13
Daten-Bit 13\	18
Daten-Bit 14\	37
Daten-Bit 15\	19
Ready\	21
^Request	nicht verdrahtet
^Accepted	22
^Stop\	20
Busy\	15
^Gate	35
Masse	36

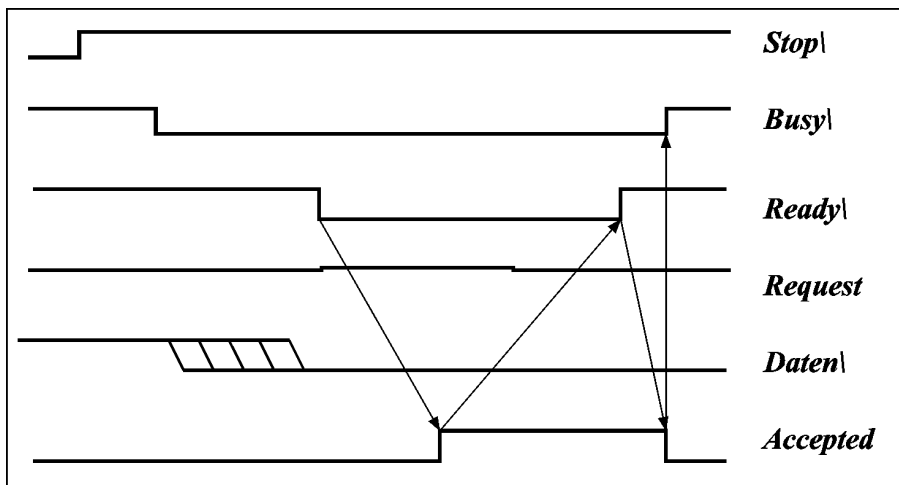


Abb. 4.2.2.2 Timing zwischen Interface und ADC

4.3 Uhr/Zähler (UHR)

4.3.1 Aufgabe

Die Uhrkarte enthält eine 16 Bit Relativzeituhr, deren Zeitregister während des Betriebes (on flight) abgefragt werden kann. Sie kann mit einem internen quarzstabilisierten Takt von 10 MHz oder extern mit bis zu 4 MHz betrieben werden. Der externe Betrieb ist gleichbedeutend mit einem Betrieb als Zähler. Beim Betrieb als Uhr besteht die Möglichkeit parallel zur eigentlichen Messungen eine zeitabhängige Totzeitmessung laufen zu lassen, um eine nachträgliche rechnerische Totzeitkorrektur zu ermöglichen. Bei Kopplung der Uhr mit anderen Datenquellen kann die gemeinsame Totzeit der gekoppelten Quellen gemessen werden.

4.3.2 Bedienungselemente

Int/Ext: (Schalter)

In Stellung *Int* wird die Uhr mit einem internen quarzstabilisierten 10 MHz Takt betrieben. In Stellung *Ext* kann die Uhr mit einem externen Takt bis maximal 4 MHz betrieben werden.

Takt: (TTL-Eingang)

Über diesen Eingang kann der Uhr ein externer Takt bis zu 4 MHz zugeführt werden. Dieser wird auf den internen 10 MHz Takt aufsynchronisiert. Die positive Flanke taktet die Uhr. Sie wird durch die Synchronisation innerhalb 100 ns unscharf. Taktbreite und -pause müssen mindestens 125 ns betragen.

Reset: (TTL-Eingang)

Über diesen Eingang kann die Uhr auf Null gesetzt werden.

High oder offener Eingang: *Reset*

min. Pulsbreite: 200 ns

Uhr: (TTL-Eingang)

Über diesen Eingang erfolgt die Abfrage der Uhr. Die positive Flanke des Abfragesignals veranlasst die Übernahme des Inhaltes des Uhrregisters in ein Pufferregister, wo es bis zur Übertragung zum Rechner gespeichert wird. Die Abfrage kann asynchron zum Takt erfolgen. Das Abfragesignal wird auf den internen 10 MHz Takt aufsynchronisiert. Von der Abfrage bis zur Übertragung des Datenwortes geht die Uhrkarte in den Zustand *Busy* und ist für weitere Abfragen gesperrt. Dieser Eingang ist gatebar über die Eingänge *GateQ* und *GateK* auf der IFS-Karte.

LFT: (TTL-Eingang)

Über diesen Eingang kann zeitabhängig die 'Lifetime' der Uhrkarte und eventuell gekoppelter Datenquellen gemessen werden. Er hat im wesentlichen die gleiche Wirkung wie eine Abfrage über den Eingang *Uhr* mit folgenden Unterschieden:

Die *Gate*-Signale wirken nicht auf diesen Eingang (änderbar). Das Datenwort hat zur Unterscheidung zusätzlich das *Lifetime-Bit* gesetzt. Die Abfrage ist gesperrt, wenn die IFS-Karte keine Lifetime (*LFTQ*) meldet.

Es wird zur Lifetime-Messung die gleiche Lifetime ($LFTQ$) verwendet wie auf der IFS-Karte. Diese Lifetime enthält konjunktiv die negierte Totzeit der Uhrkarte, das Signal $GateQ$ der IFS-Karte (Option), und im Kopplungsmodus $K2$ die Lifetime-Meldung der Kopplung

$$LFTQ = BusyQ \ \& \ GateQ(Option) \ \& \ (K2 \ + \ LFTK)$$

Achtung: Wird als externer Takt und zur Abfrage der Uhr (Uhr bzw. LFT) das gleiche Signal verwendet, so wird durch die Aufsynchronisierung der externen Signale auf den internen 10 MHz Takt mal der eine oder der andere Eingang früher sein. Soll ein Eingang mit Sicherheit vor dem anderen Eingang zum Zuge kommen, so müssen sie mindestens als Abstand die Taktbreite (100 ns) plus interne Laufzeitunterschiede (<50 ns) d.h. ca. 150 ns haben.

Die gleiche Zeitdifferenz ist einzuhalten zwischen einer Abfrage der Uhr und einem nachfolgenden $Reset$ -Signal, um mit Sicherheit ein Auslesen vor dem Löschen zu erreichen.

4.3.3 Beispiel: Lebensdauermessung mit Totzeitkorrektur

Über den Eingang Uhr wird der Zeitpunkt des Auftretens der beobachteten Ereignisse zum Rechner übertragen. Die Software erzeugt daraus ein Zeitspektrum, in dem sie die Zeit als Kanaladresse interpretiert und wie im MCA-Mode im betreffenden Kanal ein 'addiere 1' ausführt.

Über den Eingang LFT wird ein fester Takt nicht zu hoher Frequenz (damit nicht zu hohe zusätzliche Totzeit entsteht) eingegeben. Auf die gleiche Weise wie oben wird ein LFT-Spektrum erzeugt, aus dem für jeden Zeitpunkt die zugehörige Lifetime der Messung entnommen werden kann.

4.4 Up/Down-Zähler (U/D-ZLR)

4.4.1 Aufgabe

Diese Zählerkarte besitzt ein 16 Bit breites Zählerregister d.h. es kann 65535 Ereignisse registrieren. Bei Überlauf in Vorwärtsrichtung springt es auf 0 und in Rückwärtsrichtung auf 65535. Der Überlauf wird nicht angezeigt, es ist deshalb ggf. das Vorschalten eines Untersetzers zu empfehlen zur Vermeidung von Überläufen. Durch eine einfache Umverdrahtung sind jedoch auch (gekennzeichnete) Varianten entstanden, die bei Überlauf anhalten. Die Auflösungsgrenze für die Zählimpulse liegt bei ca. 200ns, d.h. 5 MHz max. Zählerfrequenz. Sollen geringere Zählimpulsabstände verarbeitet werden, bei statistisch ankommenden Signalen kann dies bereits bei geringeren Zählraten der Fall sein, so ist ein Untersetzter vorzuschalten (z.B. mit den Bausteinen 74196 (50MHz) oder 74S196 (100MHz)). Der Zählerstand kann asynchron zum Zählvorgang, d.h. ohne das Zählen zu beeinflussen, abgefragt werden. Sowohl der Zählereingang (*Count*) als auch der Eingang für das Auslesesignal (*Read*) können jeweils über Gate-Signale gesteuert werden. Alle Eingänge sind bis ca. 10V Dauerpegel gegen Überspannung geschützt und im Frequenzgang auf ca. 10 MHz begrenzt.

4.4.2 Funktion

Alle Eingangssignale werden vollständig auf verschiedene Phasen des internen 10 MHz-Taktes aufsynchronisiert, so dass der Karte die Eingangssignale völlig asynchron angeboten werden können, ohne Fehlfunktionen befürchten zu müssen. Aus diesem Aufsynchronisieren resultiert die Begrenzung der Zählfrequenz auf max. 5 MHz.

Mit dem *Read*-Signal wird der Inhalt des Zählregisters in ein Zwischenregister kopiert, wo er bis zur Übertragung zum Rechner gespeichert wird. Bis zum Abschluss der Übertragung befindet sich die Karte im Zustand *Busy* und weist weitere *Read*-Signale ab, d.h. der Inhalt des Zwischenregisters bleibt vom Zählvorgang unbeeinflusst. Das Zählen selber kann während des Übertragungsvorgangs völlig ungestört weitergeführt werden. Das aus dem *Read*-Vorgang resultierende *Busy*-Signal steht, bei entsprechender Verdrahtung (Siehe Abschnitt 3.3 [Interface-Steuerung (IFS)], Seite 8.), auf der IFS-Karte an der Buchse *Out* zur Verfügung. Die Lifetime-Messung auf der IFS-Karte basiert ebenfalls auf dem durch den *Read*-Vorgang erzeugten *Busy* und zeigt nicht etwa die Zählverluste an. Die Totzeitverluste beim Zählen sind durch Vorschalten von Untersetzern leicht in vernachlässigbare Bereiche zu zwingen (Siehe [\(undefined\)](#) [Totzeitbetrachtungen], Seite [\(undefined\)](#)).

4.4.3 Eingänge (BNC-Buchsen)

Count: Zähleingang
pos. flankengetriggert; ca. 5 MHz max.

Gate: Gate-Eingang für Count
High-Pegel: Gate offen

Read: Übertragen des Zählerinhaltes zum Rechner
pos. flankengetriggert; gatebar über den GateQ-Eingang auf der IFS-Karte

Reset: Rücksetzen (auf 0) des Zählers
High-Pegel: Reset (offener Eingang = High-Pegel!)

U/D\: Steuerung für Vorwärts/Rückwärts-Zählen
High-Pegel: Vorwärtszählen

Unbeschaltete Eingänge liegen auf High-Pegel!

Für alle Eingänge sind in Abhängigkeit von der Impulshöhe folgende minimalen Impulsbreiten erforderlich:

Pulshöhe	min. Pulsbreite
5 V	25 ns
2.5 V	50 ns

Falls die Signale *Read* und *Count* bzw. *Read* und *Reset* gleichzeitig (*Read* bis ca. 50ns später) eintreffen, so wird zuerst die *Read*-Funktion ausgeführt. Soll *Read* nach der konkurrierenden Funktion ausgeführt werden, so muss es mindestens ca. 150ns nach dieser eintreffen. Liegt *Read* ca. 50 bis 150ns nach *Count* bzw. *Reset*, so ist die Reihenfolge unbestimmt.

Das *Gate*-Signal muss mindestens 20ns vor dem *Count*-Signal anstehen, damit das Zählen freigegeben wird.

Bei einer Änderung des *U/D*-Signals im Zeitraum von 0 bis 100ns nach dem *Count*-Signal ist die Zählrichtung unbestimmt.

Die angegebenen Zeitwerte sind gültig für eine Bestückung mit 74LS-Bausteinen. Sie stellen Richtwerte dar, die aufgrund von Exemplarstreuungen variieren können. In kritischen Fällen wird empfohlen, respektvollen Abstand (ca. 20ns) von den oben angegebenen Werten zu halten.

4.5 12-Bit ADC (AD574)

4.5.1 Aufgabe

Diese Routing-Karte verwendet den AD583 Sample-and-Hold-Baustein und den AD574 12-Bit-ADC-Baustein von Analog Devices zur Digitalisierung von analogen Signalen. Der ADC arbeitet nach dem Verfahren der "Sukzessiven Approximation":

```
Konvertierungszeit : 15us - 35us
max. Abtastrate     : 26kHz
max. Linear.-Fehler: 1/4 LSB
Der Linearitätsfehler hat sich durch Alterung inzwischen deutlich
verschlechtert!
```

Durch zwei Schalter auf der Frontplatte (Uni/Bi, 10V/20V) können folgende Messbereiche gewählt werden:

```
                10V          20V
Uni:  0V - +10V   0V - +20V
Bi : -5V - +5V  -10V - +10V
```

4.5.2 Bedienungselemente

P1: (Poti)

Offset-Abgleich für Bipolar-Modus des AD574 ADC-Bausteins

P2: (Poti)

Full-Scale-Abgleich für Bipolar-Modus des AD574 ADC-Bausteins

P3: (Poti)

Offset-Abgleich für Unipolar-Modus des AD574 ADC-Bausteins

P4: (Poti)

Offset-Abgleich des AD583 Sample-and-Hold-Bausteins

Uni/Bi: (Schalter)

Einstellung von uni- oder bipolarem Modus

10V/20V: (Schalter)

Messbereichseinstellung 10V oder 20V

Analog: (Analog-Eingang)

Eingang für das zu messende Analog-Signal

Gate: (TTL-Eingang)

Gate-Eingang für das Sample-Signal

High-Pegel: Gate offen
(offener Eingang = High-Pegel)

Sample: (TTL-Eingang)
Eingang für das Trigger-Signal. Eine positive Flanke startet eine Konvertierung.

4.6 Zählerbatterie (ZLRS, ZLRx)

4.6.1 Aufgabe

Die Zählerbatterie benötigt wie jede andere Datenquelle eine IFS-Karte (Siehe Abschnitt 3.3 [Interface-Steuerung (IFS)], Seite 8.) direkt links neben der Zählersteuerung (**ZLRS**). Rechts anschließend sind die Zählerkarten (**ZLRx**) zu stecken bis zu einer maximalen Anzahl von 255 einzelnen Zählern. Jeder Zähler kann im allgemeinen nur ein Ereignis speichern und verwirft dann alle folgenden Ereignisse bis das gespeicherte Ereignis zum Rechner übertragen wurde. Die Zählersteuerung fragt reihum alle Zähler ab und sobald sie auf ein gespeichertes Ereignis stößt, überträgt sie die Nummer des Zählers zum Rechner und löscht den Speicher. Mit der Standard-Software zur Aufnahme von Spektren wird jedem Zähler ein Kanal im Spektrum zugeordnet. Die Nummerierung der Zähler beginnt bei der ersten Karte rechts von der Zählersteuerung und setzt sich auf den jeweils folgenden Zählerkarten fort. Im allgemeinen sind acht Zähler auf einer Karte untergebracht, wobei jedoch möglicherweise nicht alle von extern zugänglich sind (z.B. nur 4), weil die restlichen die Totzeitverluste ihrer Nachbarn registrieren.

Die Abfrage der Zähler erfolgt mit 5MHz. Ein voller Zyklus, in dem alle Zähler einmal abgefragt werden, dauert daher $N * 200\text{ns}$, falls keine Daten übertragen werden, wobei N die Anzahl der angeschlossenen Zähler ist. Werden Daten übertragen, so verlängert sich die Zeit. Die maximale Rate aller Zähler zusammen ist begrenzt durch die Übertragungskapazität der Verbindung zum Rechner.

Bei zu hohen Zählraten ist zur Vermeidung von Totzeitverlusten die Vorschaltung eines Untersetzers zu empfehlen (z.B. 1/10) (Siehe [\(undefined\)](#) [Vorgeschaltete Untersetzter], Seite [\(undefined\)](#)).

4.6.2 Funktion

Die Zählerbatterie besteht aus einer Zählersteuerung (**ZLRS**) und mehreren Zählerkarten (**ZLRx**) mit jeweils mehreren Zählern. Die Zählersteuerung fragt reihum alle Zähler mittels eines Abfrage-Bits (Select-Bit) ab (5MHz). Jede Zählerkarte enthält ein Schieberegister mit seriellem Eingang und parallelen Ausgängen, je einer für jeden Zähler, durch das dieses Abfrage-Bit geschoben wird. Ein Zähler ist selektiert, wenn das Bit auf dem ihm zugeordneten Ausgang erscheint. Falls er dann ein Ereignis gespeichert hat, meldet er über den PrivatBus ein 'Ready' zur Zählersteuerung. Diese überträgt daraufhin (*ReadyQ*) die in einem 8-Bit-Register mitgezählte Nummer des selektierten Zählers an den Rechner. Mit der Quittung auf die Übertragung (*AcceptedQ*) wird der selektierte Zähler gelöscht und die Abfrage fortgesetzt.

Der Abfragezyklus passt sich der Anzahl der vorhandenen Zähler an. Zu diesem Zweck übergibt jede Zählerkarte eine Anfangs- (*AM*) und Endmeldung (*EM*) an die Zählersteuerung. Erscheint mit dem folgenden Takt nach einer *EM* keine *AM*, so bedeutet dies für die Zählersteuerung, dass die nächstfolgende Karte fehlt. Sie setzt daraufhin den Nummernzähler (8-Bit-Register) auf Null, löscht die Schieberegister aller Zählerkarten und generiert ein neues 'Select-Bit'.

Falls der Nummernzähler überläuft aufgrund eines Funktionsfehlers, oder weil mehr als 255 Zähler angeschlossen sind, so wird dies über eine LED auf der Frontplatte angezeigt.

Ferner wird der Abfragezyklus abgebrochen und bei Zähler Null wieder begonnen. Die Fehleranzeige ist nur durch einen 'Stop' zu löschen.

Die Zählersteuerung meldet Busy (*BusyQ*) solange das einsynchronisierte *Ready* eines Zählers oder die Quittung des Routing (*AcceptedQ*) ansteht. Solange *BusyQ* ansteht ist der Schiebetakt gesperrt.

Ein manueller oder Rechner-Stop (*StopQ*) erzeugt ein Reset-Signal (*Reset*), das die Synchronisier-FF's für die Ready-Meldung und den gerade angewählten Zähler zurücksetzt. Da der Schiebetakt weiterläuft, werden bei anstehendem *StopQ* reihum alle Zähler immer wieder gelöscht, ohne dass dabei Daten übertragen werden.

4.6.3 Zählerkarten

4.6.3.1 ZLR1 - Vierfachzähler mit Überlaufregistrierung (TTL-Eingänge)

Die Karte enthält 8 Zähler, wobei jedoch nur die Zähler 0,2,4,6 extern zugänglich sind. Die Zähler 1,3,5,7 dienen dazu, die Totzeitverluste jeweils ihrer Vorgänger zu registrieren. Auf diese Weise können Totzeitkorrekturen durchgeführt werden.

Über einen 'Gate'-Eingang können die Zähler an- und abgestellt werden. Das *Gate*-Signal beeinflusst nicht bereits gespeicherte Ereignisse. Der *Gate*-Eingang ist konjunktiv (UND) verknüpft mit dem *Gate*-Signal von der IFS-Karte, d.h. beide Signale müssen auf High-Potential liegen, wenn die Zähler arbeiten sollen. Offene Anschlüsse liegen auf high.

Über eine LED wird angezeigt, wenn einer der Zähler ein Ereignis an das Routing überträgt.

Bedienungselemente, Anzeigen

0, 2, 4, 6: (BNC-Buchsen, TTL-Eingänge)

Eingänge für die vier Zähler 0,2,4,6. Gezählt werden die Low-High-Übergänge. Falls der geradzahlige Zähler bereits ein Ereignis speichert, wird der nächstfolgende ungeradzahlige Zähler gesetzt. Ist auch dieser belegt, so geht das Ereignis verloren.

Anzeige: (LED)

Die LED leuchtet immer, wenn einer der 8 Zähler ein Ereignis an den Rechner überträgt.

4.7 Messung der Breiten- und Abstandsverteilung von Signalen (BAV)

4.7.1 Aufgabe, Funktion

Die BAV-Karte misst mit Hilfe der Uhr-/Zähler-Karte (s.h 4.2) die Breite eines TTL-Signals oder den Abstand von zwei aufeinander folgenden TTL-Signalen. Die zeitliche Auflösung der Messung ist die der Uhr-Karte, bei internem Takt also 100ns. Die maximal messbaren Breiten bzw. Abstände sind durch ein Poti auf der Frontplatte im Bereich von ca. 50us bis 2ms einstellbar. Nach Ablauf des Zeitlimits wird die Karte zurückgesetzt und kann nach ca. 100ns das nächste Signal verarbeiten, um eine neue Messung zu beginnen. Werden die von der Uhr-Karte übertragenen Messdaten mit einem Vielkanal-Programm (z.B. MCA) erfasst, so erhält man als 'Spektrum' die zeitliche Verteilung von Breite oder Abstand.

Die BAV-Karte muss direkt rechts neben der Uhr-Karte stecken und erhält von dieser über die P-Bus-Leitung 28a das *BusyQ*-Signal. Solange die Uhr-Karte busy meldet, werden von der BAV-Karte keine Eingangssignale verarbeitet.

Breitenverteilung

Es wird der zeitliche Abstand zwischen ansteigender und abfallender Flanke des Eingangssignals gemessen. Bei nicht zu hoher Signallrate kann jedes Signal gemessen werden.

Abstandsverteilung

Es wird der zeitliche Abstand zwischen den ansteigenden Flanken zweier aufeinander folgender Eingangssignale gemessen. Daher kann bestenfalls nur jeder zweite Abstand gemessen werden.

4.7.2 Bedienungselemente

Breite/Abstand: (Schalter)

Umschaltung von Breite- auf Abstands-Messung

Limit: (10-Gang-Poti)

Einstellung des Zeitlimits (50us - 2ms) für die einzelne Messung

Reset: (BNC-Ausgang)

Ansteuerung des Reset-Eingangs der Uhr-Karte

Uhr: (BNC-Ausgang)

Ansteuerung des Uhr-Eingangs der Uhr-Karte

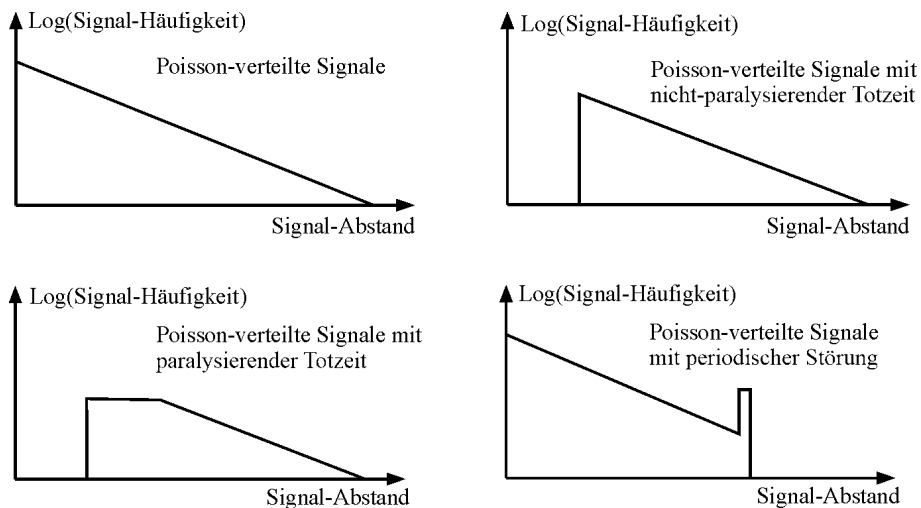
Puls: (BNC-Eingang)

Eingang für das zu messende TTL-Signal

4.8 Messung der Abstandsverteilung von Signalen (SAV)

4.8.1 Aufgabe

Bei vielen Experimenten (z.B. Wirkungsquerschnittsmessungen) ist es wichtig zu wissen in wie weit die erwartete Poisson-Verteilung der Ereignisse gestört ist z.B. durch Totzeit, Nachimpulse des Detektors, periodische Störungen usw.. Als eine geeignete Diagnosemöglichkeit bietet sich das Messen der Abstandsverteilung der Ereignisse an. Für Poisson-verteilte Ereignisse sind die Abstände exponentiell abfallend verteilt, in logarithmischer Darstellung erhält man also eine Gerade. Damit können Abweichungen von der Poisson-Verteilung rasch erkannt werden. Selbst mit dem Auge nicht mehr sichtbare Abweichungen können leicht durch einen Fit mit statistischer Wichtung aufgespürt werden, wenn das reduzierte Chi-Quadrat nicht in der Nähe von 1 liegt (Chi-Quadrat-Test, sh. Tab.).



Die Abstände zeitlich statistischer Signale (Poisson-Verteilung) sind exponentiell abfallend verteilt, d.h. in Log-Darstellung als abfallende Gerade.

Durchlaufen solche Signale eine nicht-paralysierende Totzeit, d.h. Signale, die in eine Totzeit fallen verlängern diese nicht, so erhält man wiederum die gleiche exponentielle Verteilung, aber um die Totzeit verschoben und um die Totzeitverluste vermindert.

Für eine paralysierende Totzeit, bei der jedes Signal, das in die Totzeit seines Vorgängers fällt, nur die Totzeit verlängert aber selbst nicht durchkommt, ist das der Totzeit nachfolgende mehr oder weniger ebene Dach charakteristisch. Mit zunehmender Zählrate wird der Abfall zunächst steiler und dann wieder flacher bis schließlich kein Ereignis mehr passieren kann.

Bei einer periodischen Störung in Poisson-verteilten Signalen können nur maximale Abstände in der Länge der Periode der Störung auftreten.

Mehr zu Totzeitverlusten und Abstandsverteilungen finden Sie unter Kapitel 6 [Zählratenstatistik], Seite 36.

4.8.2 Funktion

Es wird der zeitliche Abstand zwischen den ansteigenden Flanken zweier aufeinander folgender Eingangssignale gemessen. Für zufällige Ereignisse könnte man die Messung zu irgendeinem beliebigen Zeitpunkt starten und würde immer die richtige Abstandsverteilung erhalten, da die Vorgeschichte nicht in die Messung eingeht. Da jedoch auch nicht zufällige Signale wie Nachimpulse von Detektoren, periodische Störungen usw. richtig vermessen werden sollen, wird der exakte Abstand zwischen zwei aufeinanderfolgenden Signalen gemessen, was einen deutlich höheren Hardware-Aufwand bedeutet, da jedes Signal gleichzeitig Stop und Start einer Messung ist.

Die Zeitmessung erfolgt mit einer 20Mhz Quarzuhr, d.h. mit einer Auflösung von 50ns. Niedrigere Zeittakte sind durch Austausch des Quarzoszillators möglich. Für höhere Frequenzen muss jedoch ein schnellerer ispLSI-Baustein (1032E-100MHz) eingesetzt werden.

Die Zeitmessung erfolgt mit 16 Bits, bei einer Taktfrequenz von 20MHz können also Pulsabstände bis 3.27675ms gemessen werden. Darüber hinaus gehende Abstände werden im letzten Kanal (65535) gesammelt. Der Kanal 0 entspricht dem Abstand 0, der aus verständlichen Gründen nicht gemessen werden kann. Der Übergang von einem Kanal zum nächsten erfolgt linear gleitend, d.h. Abstände der Länge $n+p$ (n ganz, $p < 1$) tragen $1-p$ Anteile zum Kanal n und p Anteile zum Kanal $n+1$ bei.

!!! Bei Verwendung des Messprogrammes MCA mit dem Display-Programm Peak ist zu beachten, dass die Kanäle von 1 ab gezählt werden!!!

Totzeitverluste

Damit nach einer erfolgten Abstandsmessung nicht erst auf die Übertragung der Daten gewartet werden muss und dadurch das nächste Intervall verloren geht, ist dem Uhrregister ein 2-stufiger Fifo-Speicher nachgeschaltet. Gemäß der Poisson-Verteilung der eintreffenden Signale können jedoch zufälligerweise mehrere kurze Intervalle hintereinander eintreffen, so dass die Kanalkapazität des Übertragungsweges überschritten wird und Daten verloren gehen. Bei statistischen Pulsen verzerren diese Verluste nicht die Abstandsverteilung bei periodischen Pulsen können allerdings Interferenzen auftreten (s.u.)

Da der Fifo-Speicher die statistischen Schwankungen ausgleicht, minimiert er deutlich die Totzeitverluste (sh. Tab.)

Interferenzen bei periodischer Pulsfolge

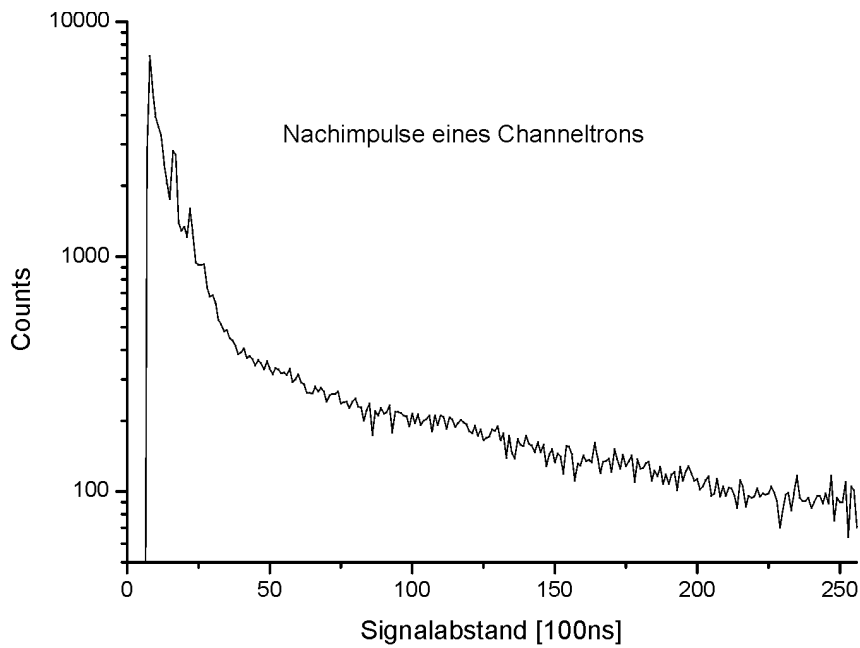
Bei gewissen periodischen Mustern von kurzen und langen Abständen können Interferenzen mit der internen Ablaufsteuerung auftreten, die die gemessene Abstandsverteilung verzerren. Bei einer Fifo-Tiefe von 2 würde z.B. eine periodische kurz-kurz-lang Abstandsfolge dazu führen, dass jeder zweite lange Abstand durch Totzeit verloren geht. Kurz bedeutet kürzer als die halbe Datenübertragungszeit und lang bedeutet länger als die dreifache Datenübertragungszeit. Eine kurz-lang Abstandsfolge würde hingegen richtig wiedergegeben.

4.8.3 Bedienungselemente

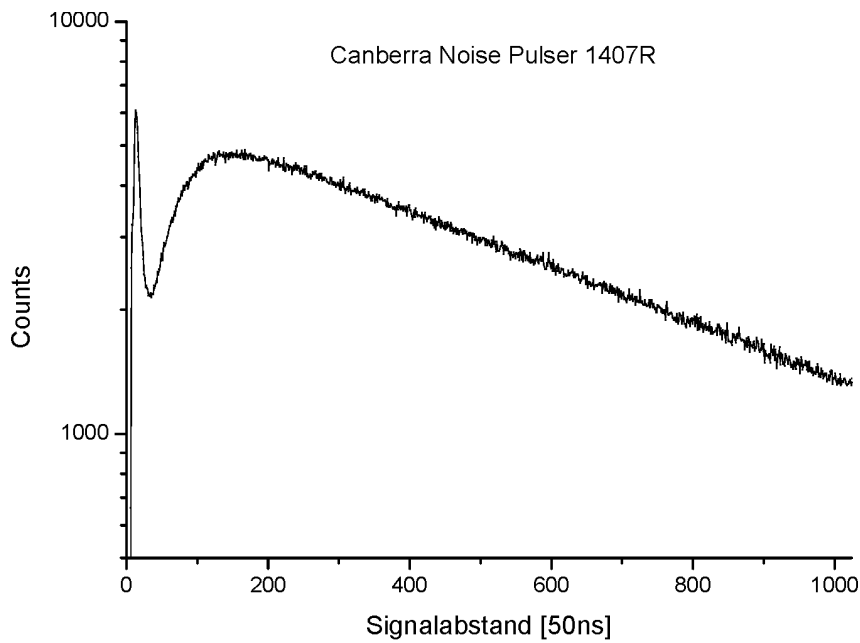
Puls: (BNC-Eingang)

Eingang für das zu messende TTL-Signal > 20ns

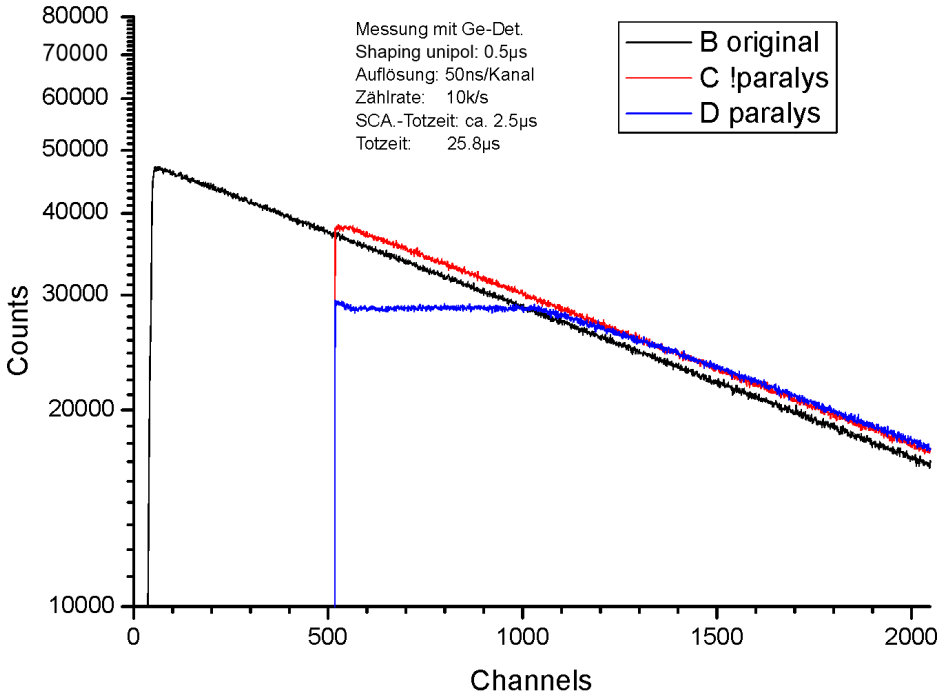
4.8.4 Beispiele für Abstandsverteilungen



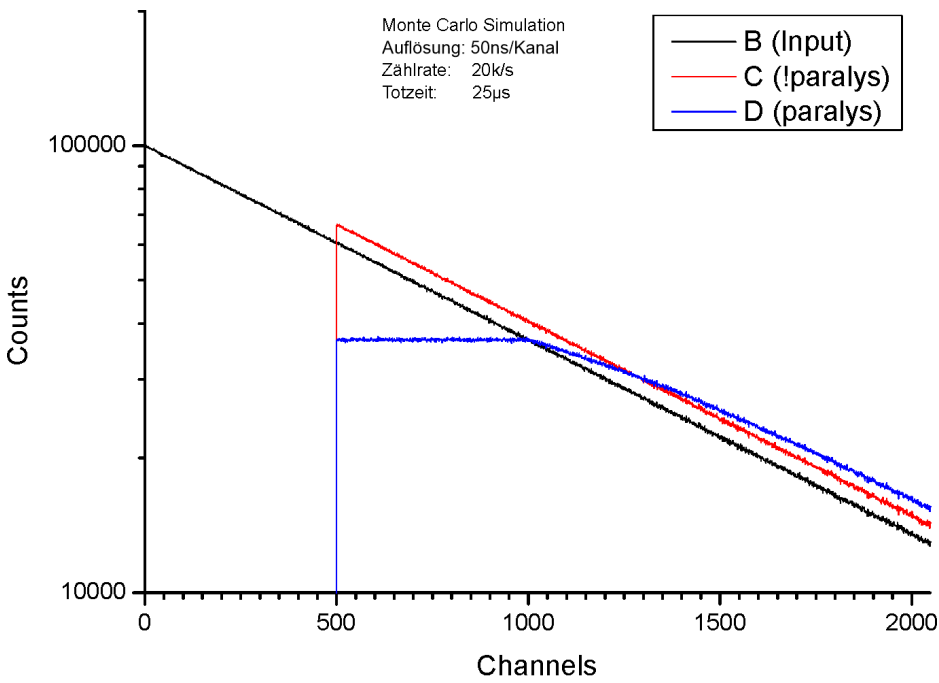
Totzeit (500ns) und Nachimpulse (bis ca. 4 μ s) eines Channeltrons.



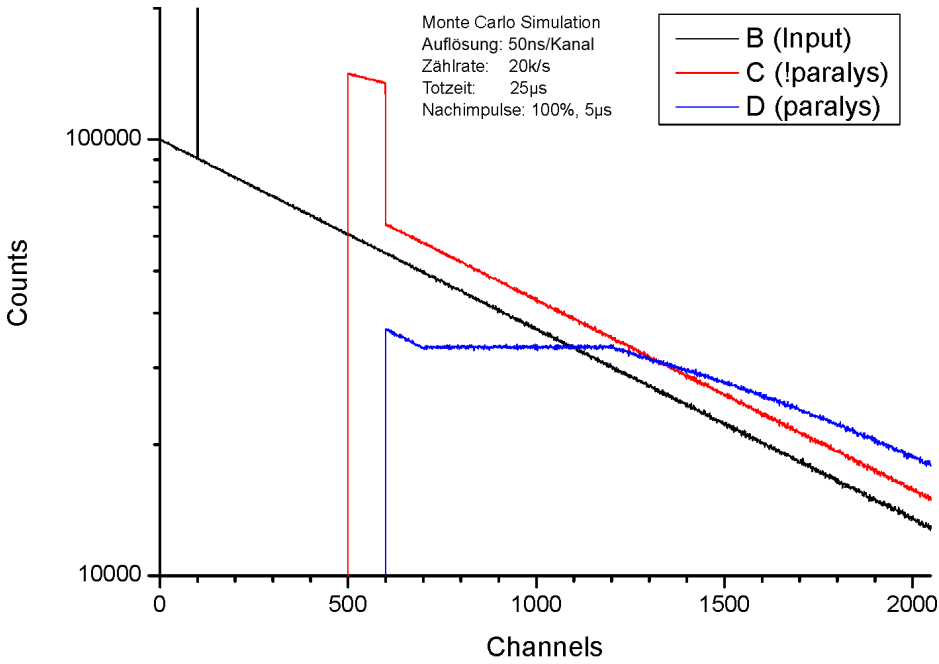
Der "Noise Pulsers" 1407R von Canberra zeigt eine markante Abweichung von der Poisson-Statistik unterhalb von 10 μ s.



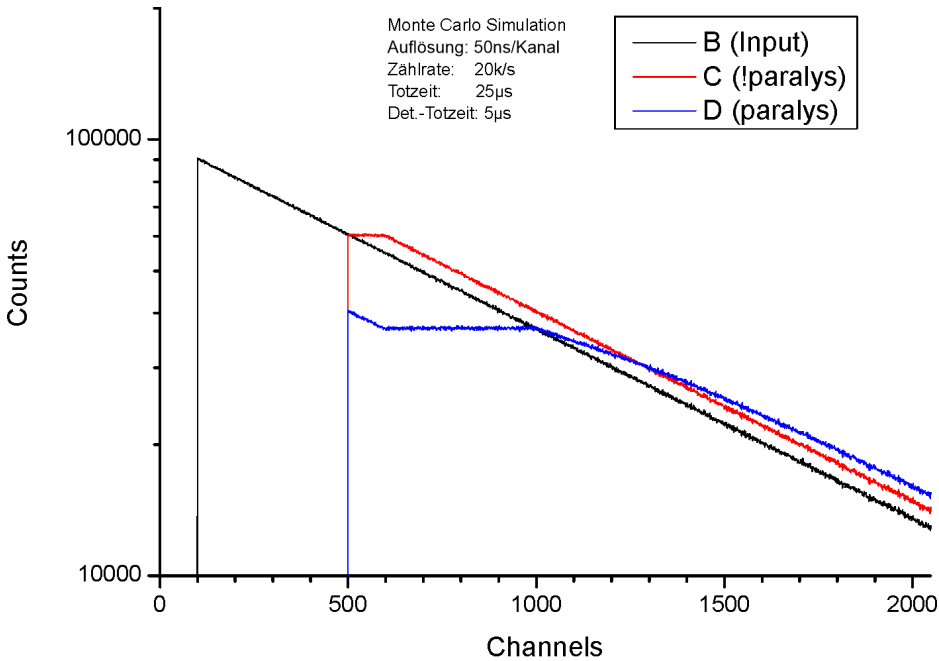
Ge-Detektorsignale bei Verwendung eines Hauptverstärkers (, 500ns Shaping, unipolar) mit "Pole-Zero-Cancellation".



Monte-Carlo-Simulation der Abstandsverteilung nach einer Totzeit bei Poisson-verteilter Eingangsrate.



Monte-Carlo-Simulation der Abstandsverteilung nach einer Totzeit bei Poisson-verteilter Eingangsrate mit 100% Nachimpulsen.



Monte-Carlo-Simulation der Abstandsverteilung nach einer Totzeit bei Poisson-verteilter Eingangsrate, die bereits eine vorausgegangene nicht paralysierende Totzeit durchlaufen hat.

4.8.5 Tabellen

Chi-Quadrat Test (1-P)

(1-P): Wahrscheinlichkeit, dass redChi^2 kleiner ist

redChi ²	Freiheitsgrade					
	200	400	600	800	1000	1200
0.900	0.1582	0.0749	0.0380	0.0200	0.0107	0.0058
0.920	0.2151	0.1273	0.0800	0.0518	0.0341	0.0227
0.940	0.2813	0.1998	0.1488	0.1135	0.0878	0.0686
0.960	0.3549	0.2912	0.2473	0.2135	0.1863	0.1637
0.980	0.4333	0.3970	0.3709	0.3497	0.3317	0.3158
1.000	0.5133	0.5094	0.5077	0.5066	0.5059	0.5054
1.020	0.5917	0.6196	0.6418	0.6606	0.6769	0.6916
1.040	0.6657	0.7197	0.7590	0.7899	0.8153	0.8366
1.060	0.7329	0.8039	0.8504	0.8836	0.9083	0.9271
1.080	0.7918	0.8699	0.9146	0.9425	0.9606	0.9728
1.100	0.8417	0.9182	0.9551	0.9746	0.9854	0.9915

Totzeitverluste bei statistischen Pulsen

Verluste: $1 - \exp(-RT) * (1 + RT + RT^2/2 + \dots + RT^n/n!)$

R: Datenrate

T: Totzeit = 3000ns mittel, 5200ns max, nicht paralyisierend

n+1: Fifo-Tiefe

Bei $RT > 1$ wird die Kanalkapazitaet ueberschritten und es kommt zum Datenstau.

RT \	1	2	3	4 Fifo-Tiefe
0.00	0.0000	0.0000	0.0000	0.0000
0.02	0.0198	0.0002	0.0000	0.0000
0.04	0.0392	0.0008	0.0000	0.0000
0.06	0.0582	0.0017	0.0000	0.0000
0.08	0.0769	0.0030	0.0001	0.0000
0.10	0.0952	0.0047	0.0002	0.0000
0.15	0.1393	0.0102	0.0005	0.0000
0.20	0.1813	0.0175	0.0011	0.0001
0.25	0.2212	0.0265	0.0022	0.0001
0.30	0.2592	0.0369	0.0036	0.0003
0.35	0.2953	0.0487	0.0055	0.0005
0.40	0.3297	0.0616	0.0079	0.0008
0.45	0.3624	0.0754	0.0109	0.0012
0.50	0.3935	0.0902	0.0144	0.0018
0.60	0.4512	0.1219	0.0231	0.0034
0.70	0.5034	0.1558	0.0341	0.0058
0.80	0.5507	0.1912	0.0474	0.0091
0.90	0.5934	0.2275	0.0629	0.0135
1.00	0.6321	0.2642	0.0803	0.0190

5 Spezielle Interface-Karten

Diese Interface-Entwicklungen werden hier nur aufgelistet und sind in den Unterlagen für die einzelnen Experimente im Detail beschrieben. Diese Aufstellung enthält auch Oldies, in der Hoffnung, sie eines Tages wieder recyceln zu können.

5.1 Elektron-Positron-Spaltfragment-Winkelverteilungs-Experiment (K5)

EPOS-Steuerung

EPOS-Messprogramm

Eingabe des LB-Triggers und Generierung von Gate- und Delay-Signalen für die "in beam" und "out of beam" Messphasen.

5.2 Ion-Ion-Stoß-Experiment (Labor 016)

LIC-Interface

IIM-, IIC-, LIC-Messprogramme

Umschalter IIM-, IIC-, LIC-Messungen; Kontrolle der LIC-Positions-Laufzeit-Messung

TAKT-Karte

IIC-, IIF-, IIP-, LIC-, LIP-Messprogramme

Takte zum Auslesen der Zählerkarten; Totzeitmessung

IIM-Interface

IIM-, LIC-Messprogramme

Interface zum SSL-2401 Position-Computer

LII-Interface

LII-, LIP-Messprogramme

FIFO-gepufferte Dateneingabe für Pulsed-Beam-Koinzidenz-Experimente; Interface zum SSL-2401 Position-Computer

5.3 Elektron-Ion-Stoß-Experiment (Labors 017, SI)

ESS-Kontrolle

ESS-Messprogramm

Totzeit-Überwachung.

ESS-Kanalnummer-Eingabe

ESS-Messprogramm

Interface für parallele Eingabe der aktuellen Kanalnummer.

DRM-Interface

DRM-Messprogramm

Interface zum SSL-2401 Position-Computer

6 Zählratenstatistik

Die Zählratenstatistik ist inzwischen zu einem eigenen Paper geworden.

HTML-Version:

<http://www.strz.uni-giessen.de/ExpHelp/statistik/statistik.html>

PDF-Version:

<http://www.strz.uni-giessen.de/ExpHelp/statistik/statistik.pdf>

7 Technische Details

7.1 Routing-Bus

7.1.1 Routing-Bus Signale

Der Routing-Bus (Tab. 7.1.1.1) besteht aus zwei Teilen, dem 'Allgemeinen Bus', der über die ganze Breite des Überrahmens geht und dem 'Privat-Bus', der erst durch das Stecken benachbarter Karten entsteht und auf diese begrenzt ist.

Allgemeiner Bus (A-Bus)

Der allgemeine Bus belegt die Anschlüsse 1a, 1c usw. bis 21c. Die Anschlüsse mit dem Suffix a und c führen im Gegensatz zum Privat-Bus zu verschiedenen Bus-Leitungen und dürfen nicht gebrückt werden.

- **Stromversorgung (Standard):**

- + 5V / 10A
 - +15V / 1A
 - 15V / 1A

- **DO - D15 (Datenleitungen):**

Sechszehn Datenleitungen stehen zur Verfügung. Low auf einer Leitung bedeutet, dass das Daten-Bit gesetzt ist. Offene Leitung heißt, das Daten-Bit ist nicht gesetzt.

- **Kopp-, LFT-Bit:**

Diese beiden Bits sind Kennungs-Bits, die von den Interfaces gesetzt werden können. Die Standard-Interface-Steuerung (IFS-Karte, Siehe Abschnitt 7.2.3 [Interface-Steuerung], Seite 45.) nutzt sie zur Anzeige von gekoppelten Daten bzw. bei der Lifetime-Messung.

- **10 MHz-Clock/-Clock\:**

Quarzstabilisierter Mastertakt der Routing-Steuerung.

Tastverhältnis: 1:2

Stabilität: 10ppm

- **1 MHz-Clock:**

Vom Mastertakt abgeleiteter 1MHz Takt.

Tastverhältnis: 1:2

- **PAddr1..3 (Steckplatzadresse):**

Für jeweils zwei benachbarte Steckplätze ist eine gemeinsame Steckplatzadresse 0-7 verdrahtet (Abb. 2.2.0.1). Die Interface-Steuerungen müssen so gesteckt werden, dass sie verschiedene Adressen einnehmen, da sie nur dann eindeutig zu adressieren sind.

- **SAddr0..3 (Scanner-Adresse):**

Ein Steckplatz gilt als angewählt, wenn die Bits 1-3 von Steckplatz- und Scanner-Adresse übereinstimmen. Nur dann darf die Interface-Steuerung mit der Routing-Steuerung in Kontakt treten. Das Bit *SAddr0* erlaubt noch das Ansprechen zweier

unterschiedlicher Datenregister eines Eingabekanals. Die Scanner-Adresse wird den übertragenen Daten von der Rechneranpassung als Quellenkennung beigefügt, um sie unterscheidbar zu machen (Siehe [\[Rechneranpassung\]](#), Seite [\[Rechneranpassung\]](#)).

- **TN, TV** (Freigabetakte für den Adressvergleich):
TN, TV erscheinen verzögert zu der Scanner-Adresse, so dass sichergestellt ist, dass die Adress-Bits eingeschwungen sind und keine Vergleichsfehler auftreten können. Ferner ermöglichen *TN, TV* eine Unterscheidung welcher der beiden Scanner unterschiedlicher Priorität gerade aktiv ist.
 - TN*: Normal-Scanner
 - TV*: Vorrang-Scanner
- **PrioReqB** (Vorrang Anmeldung):
 Über diese Leitung kann bei der Routing-Steuerung ein vorrangiger Bearbeitungswunsch angemeldet werden. Die Routing-Steuerung fragt in einer bestimmten Phase jedes Scan-Zyklus diese Leitung ab und schaltet auf den gewünschten Scanner.
- **ReadyB** (Data Ready); **RequestB** (Data Request); **AcceptedB** (Data Accepted)
 Mit diesen drei Signalen synchronisiert sich die Datenübertragung eines vom Scanner angewählten Eingabekanals. Die Interface-Steuerung meldet *ReadyB*, falls sie Daten übertragen will. Dies hat ein Stoppen des Scanners zur Folge. Mit *RequestB* wird die Datenquelle aufgefordert die Daten auf den Bus zu geben. Beendet wird die Übertragung mit *AcceptedB*. Mit diesem Signal wird im allgemeinen auch die Datenquelle zurückgesetzt, die dann das nächste Datenwort aufnehmen kann.
- **StopB**:
 Dieses Signal teilt allen Eingabekanälen gleichzeitig mit, dass die Routing-Steuerung gestoppt wurde.

Privat-Bus (P-Bus)

Der Privat-Bus baut sich erst durch Stecken der Karten auf. Eine fehlende Karte oder das Fehlen entsprechender Brücken auf den Karten unterbrechen den Privat-Bus. Er reicht von den Leitungen 22 bis 32 einschließlich. Die Leitungen mit dem Suffix a führen zur linken Nachbarkarte und die mit c zur rechten. Soll ein Bus aufgebaut werden, so sind die Anschlüsse a und c einer Leitung zu brücken. Die individuelle Nutzung des Privat-Busses ist den einzelnen Kartenbeschreibungen zu entnehmen (z.B. RST-, IFS-, KPL- Karten).

Tab. 7.1.1.1 Data-Routing-Bus

	a	Pin	c	
Spannungs-	(digital) +5V	1	+5V (digital)	Spannungs-
Versorgung	(digital) 0V	2	0V (digital)	Versorgung
		D0\ 3	D1\	
		D2\ 4	D3\	
		D4\ 5	D5\	
		D6\ 6	D7\	
		D8\ 7	D9\	
		D10\ 8	D11\	
		D12\ 9	D13\	
		D14\ 10	D15\	
A-Bus	Kopp-Bit\	11	LFT-Bit\	A-Bus
	SAddr0\	12	1MHz-Clock	
	SAddr1\	13	PAddr1\	
	SAddr2\	14	PAddr2\	
	SAddr3\	15	PAddr3\	
	RequestB	16	ReadyB\	
	StopB\	17	AcceptedB	
	10MHz-Clock\	18	10MHz-Clock	
	TV	19	TN	
	(analog) 0V	20	PrioReqB\	
	(analog) +15V	21	-15V (analog)	
		22		
		23		
		24		
		25		
P-Bus		26		P-Bus
		27		
		28		
		29		
		30		
		31		
		32		

(Signalnamen mit '\': aktiv low)

7.1.2 Routing-Bus Abschluss

Der Bus-Abschluss ist normalerweise ganz links auf der Rückseite der ersten VG-Steckerleiste installiert. Es bestehen jedoch auch Sonderlösungen durch von vorne gesteckte Karten, insbesondere wenn der Überrahmen in einen Data-Routing-Bus und einen Control-Routing-Bus aufgeteilt ist.

Der Bus-Abschluss ist je nach Aufgabe der Bus-Leitungen als aktiver Abschluss (ca. 3V) oder durch Pulldown-Widerstände realisiert (Abb. 7.1.2.1). Der 27 Ohm Widerstand, mit denen die Pulldowns zusammengefasst sind, ist experimentell ermittelt und verbessert die Signale ganz erheblich. Insbesondere vermindert er ein Übersprechen, das im Data-Routing zum Ausfall einzelner 10MHz-Takte beim Schalten der *SADR(0:3)*-Leitungen geführt hatte. Warum das Ganze gerade in dieser Beschaltung ordentlich läuft, weiß keiner so recht. Die von den Interface-Karten erzeugten Signale können mittels Treiber-Bausteinen mit mindestens 15mA (besser 24mA) Low-Signalstrom (z.B. 'LS245, 'LS373, 'LS374) auf den Bus gegeben werden.

Bei der Entwicklung von Interface-Karten bitte stets darauf achten, dass sowohl **Senders** als auch **Empfänger-Bausteine möglichst nahe am Bus platziert werden!** Denn ein solcher Bus ist eine sehr heikle Hf-Übertragungsstrecke (ca. 20MHz), die man durch falsch aufgebaute Steckkarten empfindlich stören kann.

Abb. 7.1.2.1 Routing-Bus-Abschluss

```

D00\ 3a <--220hm--|
D01\ 3c <--220hm--|
D02\ 4a <--220hm--|
D03\ 4c <--220hm--|
D04\ 5a <--220hm--|
D05\ 5c <--220hm--|
D06\ 6a <--220hm--|
D07\ 6c <--220hm--|          +-----+
D08\ 7a <--220hm--|          | aktiver |
D09\ 7c <--220hm--+-----+ | Abschluss|
D10\ 8a <--220hm--|          | 3 Volt  |
D11\ 8c <--220hm--|          +-----+
D12\ 9a <--220hm--|          |
D13\ 9c <--220hm--|          ===
D14\ 10a <--220hm--|
D15\ 10c <--220hm--|
Kopp-Bit\ 11a <--220hm--|
LFT-Bit\ 11c <--220hm--|
1MHz-Clock 12c <--220hm--|
ReadyB\ 16c <--220hm--|
PrioReqB\ 20c <--220hm--|
PADR1\ 13c )
PADR2\ 14c ) Steckplatzkodierung 0V/5V
PADR3\ 15c )
SADR0\ 12a <--220hm--|
SADR1\ 13a <--220hm--|          +-----+
SADR2\ 14a <--220hm--|          | passiver |
SADR3\ 15a <--220hm--+-----+ | Abschluss|
RequestB\ 16a <--220hm--|          | 27 Ohm  |
StopB\ 17a <--220hm--|          +-----+
AcceptedB 17c <--220hm--|          |
10MHz-Clock\ 18a <--220hm--|          ===
10MHz-Clock 18c <--220hm--|
TV 19a <--220hm--|
TN 19c <--220hm--|
(analog) 0V 20a <--220hm--|

```

7.2 Komponenten und Schnittstellen

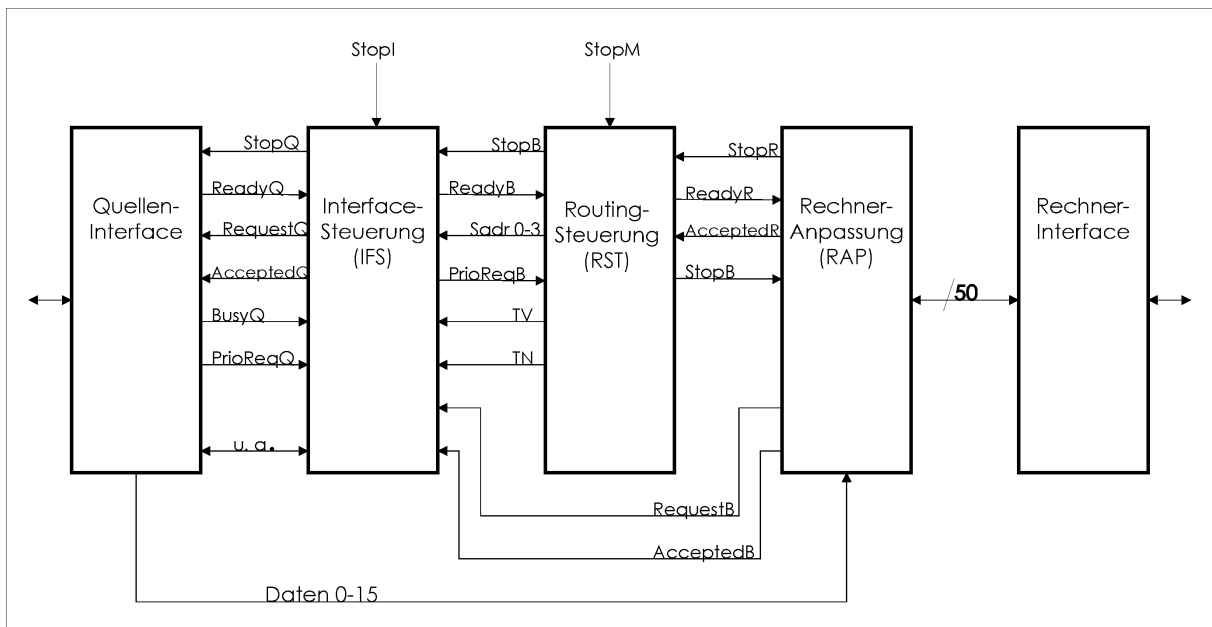


Abb. 7.2.0.1 Komponenten und Schnittstellen des Routing

7.2.1 Datenquellen <-> Quellen-Interfaces

Eine solche Schnittstelle existiert immer dann, wenn ein bereits vorhandenes Gerät angeschlossen werden soll (z.B. ein Silena ADC wird mittels Silena-ADC-Interface an das Routing angepasst) und ist i.a. ausschließlich für diesen Zweck ausgelegt.

7.2.2 Quellen-Interface <-> Interface-Steuerung

Die IFS-Karte baut nach rechts einen Privat-Bus den IFS-Bus zum Quellen-Interface hin auf. Er endet normalerweise beim Quellen-Interface, wobei dieses jedoch aus mehreren Karten bestehen darf. Alle Signalnamen auf dem IFS-Bus enden mit Q. Die Signalbezeichnungen sind in positiver Logik angegeben, d.h. wenn an einer Leitung 'high' anliegt, so ist das zugehörige Signal wahr. Ein Backslash "\" bedeutet die Negierung eines Signals: BUSY\ = NOT (BUSY). Wenn also eine Datenquelle Busy meldet, dann liegt 'low' auf der Leitung BUSY\. Mit '^' gekennzeichnete Signale werden auf der IFS-Karte erzeugt.

^LFTQ\: (Lifetime Quelle)

Dieses Signal wird auf der IFS-Karte erzeugt und stellt die Lifetime (=NOT(Totzeit)) zur Verfügung. *LFTQ* setzt sich konjunktiv (UND) zusammen aus der negierten Busy-Meldung

der Datenquelle (*BusyQ*), dem Eingang *GateQ* der IFS-Karte (Option), und im Kopplungsmodus *K2* der Lifetime-Meldung *LFTK* der Kopplung.

$$LFTQ = BusyQ \wedge * GateQ(Option) * (K2 * LFTK) \wedge$$

PrioReqQ: (Vorrang Quelle)

Dieses Signal zeigt der IFS-Karte an, dass die Quelle mit Vorrang bedient zu werden wünscht. Es kann statisch anstehen, mit *ReadyQ* wird es wirksam. *PrioReqQ* schaltet die IFS-Karte in den Kopplungsmodus *Frei*. Eine Datenquelle, die mit Vorrang läuft, kann deshalb nicht an einer Datenkopplung teilnehmen.

~GateSQ: (Gate Quelle)

Mit diesem Signal gated die IFS-Karte die angeschlossene Datenquelle auf- und zu.

BusyQ: (Busy Quelle)

Mit diesem Signal meldet die Datenquelle ihr Busy an die IFS-Karte.

~StopQ: (Stop Quelle)

Dieses Signal teilt der Datenquelle mit, dass ihr keine Daten vom Routing abgenommen werden. Als Ursachen hierfür kommen in Frage:

- Stop durch den Rechner.
- Die RST-Karte ist manuell gestoppt.
- Die IFS-Karte ist manuell gestoppt.

Normalerweise clear dieses Signal die Datenquelle.

ReadyQ: (Data Ready Quelle)

~RequestQ: (Data Request Quelle)

~AcceptedQ: (Data Accepted Quelle)

Diese drei Signale bilden zusammen einen Hand-Shake-Dialog (Abb. 7.2.2.2) zur Übernahme der Daten von der Quelle. Mit *ReadyQ* teilt die Datenquelle der IFS-Karte mit, dass sie ein Datenwort übertragen möchte. Dieses Signal muss zumindest bis zum Beginn von *RequestQ* anstehen. Mit *RequestQ* fordert die IFS-Karte die Datenquelle auf, für die Dauer des Signals ihre Daten auf den Bus zu geben. Mit *AcceptedQ* wird der Datenquelle mitgeteilt, dass die Daten übernommen wurden.

Daten(0-15): (Datenleitungen)

Die Daten laufen nicht über die IFS-Karte sondern werden direkt auf den Routing-Bus gegeben. Sie sind über Treiber-Bausteine mit mindestens 24mA Low-Signalstrom (z.B. 'LS245, 'LS373, 'LS374) auf die Leitungen zu geben.

KoppK: (Kopplungssignal)

BusyK: (Busy Kopplung)

LFTK: (Lifetime Kopplung)

Diese Signale gehören dem Kopplungs-Bus an und werden zusammen mit der Kopplungskarte beschrieben (Siehe $\langle \text{undefined} \rangle$ [Kopplungskarte <-> Interface-Steuerungen], Seite $\langle \text{undefined} \rangle$).

Tab. 7.2.2.1 Schnittstelle zur Datenquelle (IFS-Bus)

Signal	Funktion	Leitung	IFS	Quelle	KPL
LFTQ	Lifetime	25	c	a	
PrioReqQ\	Vorrang	26	c	a	
GateSQ	Gate	27	c	a	
BusyQ\	Busy	28	c	a	
StopQ	Stop	29	c	a	
AcceptedQ	Data	30	c	a	
RequestQ\	Data Request	31	c	a	
ReadyQ\	Date Ready	32	c	a	
KoppK\	Kopplungssignal	23	a,c	a,c	a
BusyK\	Kopplung	24	a,c	a,c	a
LFTK	Lifetime	22	a,c	a,c	a

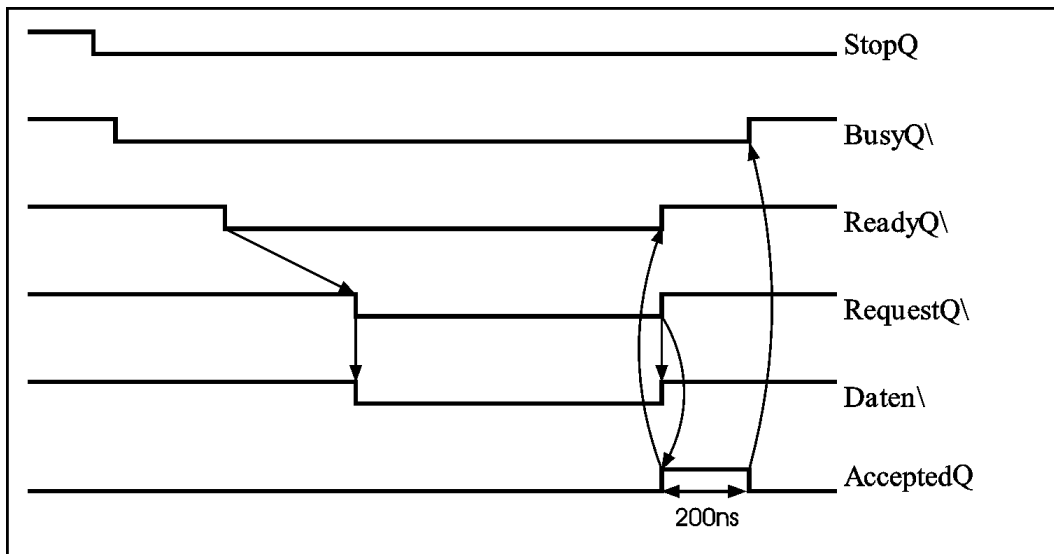


Abb. 7.2.2.2 Timing zwischen IFS-Karte und Datenquelle

7.2.3 Interface-Steuerung

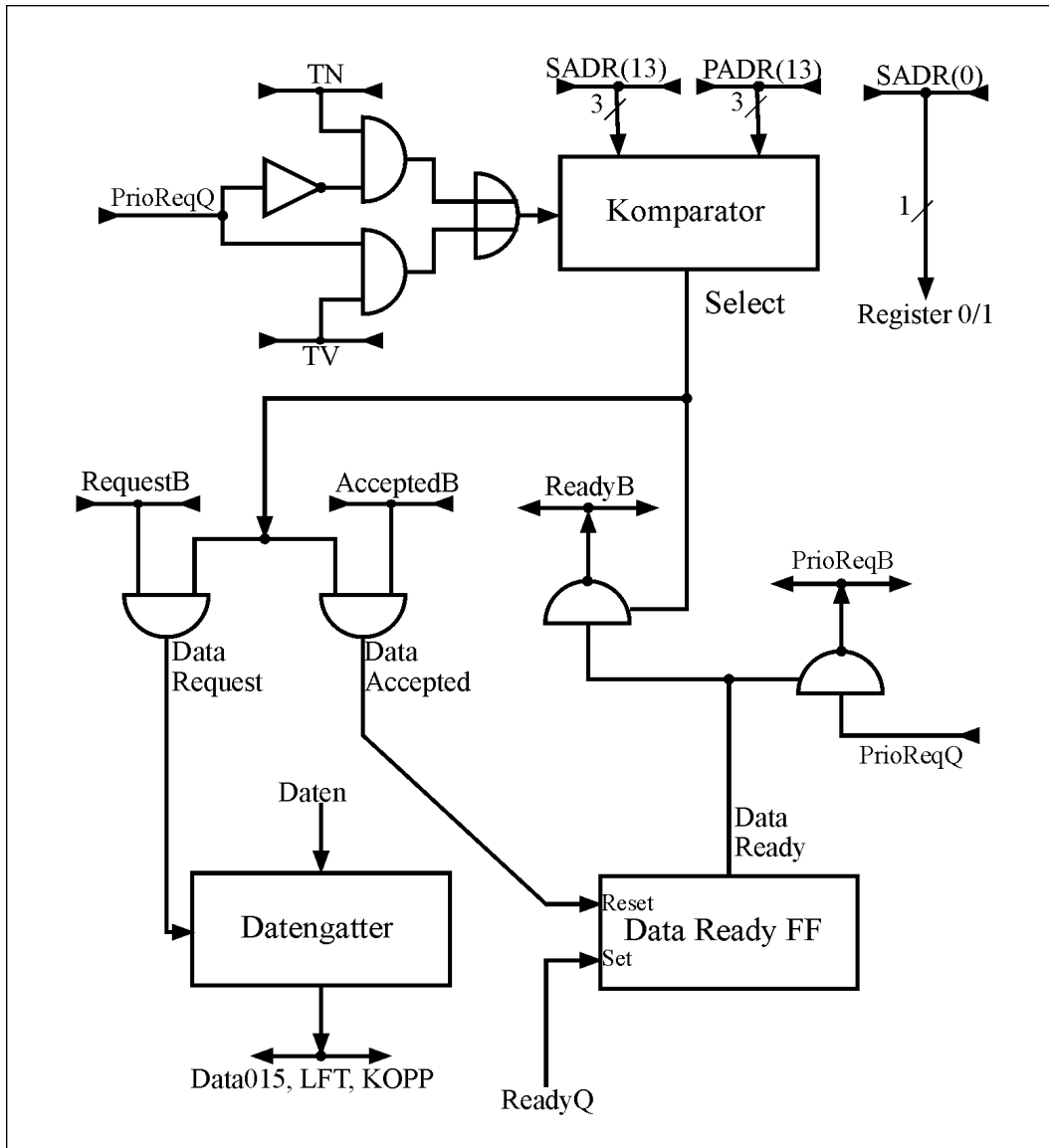


Abb. 7.2.3.1 Interface-Steuerung

7.2.4 Interface-Steuerung <-> Routing-Steuerung

Die Routing-Steuerung verständigt sich mit der Interface-Steuerung durch einen Hand-Shake-Dialog über Signalleitungen des Routing-Busses:

Signalnamen ohne Backslash '\': aktiv high

Signalnamen mit Backslash '\': aktiv low

Ein '^' vor dem Signalnamen: die Routing-Steuerung ist die Quelle des Signals

^10 MHz-Clock,-Clock

Quarzstabilisierter 10MHz Mastertakt der Routing-Steuerung.

Tastverhältnis: 1:2

Stabilität: 10ppm

^1 MHz-Clock

vom Mastertakt abgeleiteter 1MHz-Takt.

Tastverhältnis: 1:2

^SAddr0..3 (Scanner-Adresse)

Scanner-Adresse (4 Bits) zur Abfrage der Interface-Steuerungen.

ReadyB

Die abgefragte Interface-Steuerung meldet sich bei der Routing-Steuerung mit *ReadyB*, falls sie ein Datenwort übertragen will.

^StopB

Dieses Signal teilt allen Interface-Steuerungen mit, dass die Routing-Steuerung gestoppt wurde (manuell oder durch Rechner).

^TN, ^TV

Freigabetakte für den Adressvergleich auf den Interface-Steuerungen für Normal- (*TN*) bzw. Vorrangmodus (*TV*).

PrioReqB

Anmeldung eines vorrangigen Bearbeitungswunsches bei der Routing-Steuerung durch die Interface-Steuerungen. Dieses Signal muss mit 'Open-Collector-Technik' (wired OR) beschaltet werden.

7.2.5 Routing-Steuerung

Im Mai 2000 wurde eine Neuentwicklung der Routing-Steuerung fertiggestellt und nach und nach zum Einsatz gebracht. Sie verwendet die alten Frontplatten. Die Schaltung ist im

wesentlichen auf einem hochintegrierten, programmierbaren Baustein untergebracht. Ihre Funktionalität und Betriebssicherheit wurden verbessert.

Die Routing-Steuerung ist mit zwei alternativ laufenden Scannern unterschiedlicher Priorität ausgestattet, mit denen die Eingabekanäle abgefragt werden. Sie schickt zu diesem Zweck über vier Bus-Leitungen (*SAddr0..3*) nacheinander die Scanner-Adressen 0-15 an alle Interface-Steuerungen. Diese vergleichen die drei höchstwertigen Bits der Scanner-Adressen mit ihrer Steckplatzadresse (*PAddr1..3*), die für jeden Steckplatz fest verdrahtet ist. Bei Gleichheit gilt der Eingabekanal als angewählt und kann bei Bedarf mit der Routing-Steuerung kommunizieren.

Es besteht die Möglichkeit, einen oder mehrere Eingabekanäle mit Vorrang vor den anderen abfragen zu lassen. Dazu müssen diese Eingabekanäle einen Vorrangwunsch (*PrioReqB*) bei der Routing-Steuerung anmelden, sobald sie bereit sind ein Datenwort zu übertragen. Diese Anmeldung kann zu jeder Zeit asynchron erfolgen. Nach jedem Scanner-Schritt prüft die Routing-Steuerung die *PrioReqB*-Leitung ab und setzt den Betrieb mit dem gewünschten Scanner (Vorrang- bzw. Normal-Scanner) fort. Der jeweils ruhende Scanner behält seine Position bei, so dass er an der unterbrochenen Stelle weiterlaufen kann. Der im Vorrangmodus arbeitende Eingabekanal muss sich des Freigabetaktes *TV* bedienen im Gegensatz zu *TN* für den Normalmodus.

Ein Stop der Routing-Steuerung kann vom Rechner her erfolgen (*StopR*) oder über einen Schalter auf der Frontplatte (*StopM*). Beide bewirken ein Anhalten der Daten-Scanner. Ferner wird der Stop den Eingabekanälen mitgeteilt (*StopB*) und über die Interface-Steuerung den Datenquellen (*StopQ*).

Da das gleichzeitige Schalten aller Scanner-Adress-Bits zu massiven Störungen auf dem Routing-Bus geführt hat (einzelne 10MHz Takte wurden verstümmelt), ist die Folge der Scanner-Adressen so gewählt, dass sich stets nur möglichst wenig Bits gleichzeitig ändern. Da die IFS1-Karte das niederwertigste Bit als Clock-Signal verwendet, muss dieses eine 0-1-0 Folge haben. Unter dieser Voraussetzung und der Forderung nach der Möglichkeit für einen verkürzten Scanner-Umlauf, war eine Beschränkung auf 2-Bit-Wechsel möglich.

Die Scanner-Adressen sind in vier Gruppen zusammengefasst, die sich am Anfang und Ende auch gegenüber einigen anderen Gruppen nur um maximal zwei Bits unterscheiden. Sie können deshalb in Einer-, Zweier- und Vierer-Kombinationen verknüpft werden.

Gruppe 0, Adressen 0 - 3

```
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
```

Gruppe 1, Adressen 4 - 7

```
0 1 1 0
0 1 1 1
0 1 0 0
0 1 0 1
```

Gruppe 3, Adressen 12 - 15

```
1 1 0 0
```

```

1 1 0 1
1 1 1 0
1 1 1 1

```

Gruppe 2, Adressen 8 - 11

```

1 0 1 0
1 0 1 1
1 0 0 0
1 0 0 1

```

Erlaubte Kombinationen von Gruppen mit maximal 2-Bit-Wechsel sind:

0, 1, 2, 3, 0-1, 0-2, 1-3, 2-3, 0-1-3-2

Bei unzulässigen Kombinationen, und falls kein Jumper gesetzt ist, wird 0-1-3-2 ausgeführt.

Es ist zu beachten, dass die verschiedenen Eingabekanäle von der Routing-Steuerung nicht nach einem 'first in - first out' Verfahren bearbeitet werden, sondern durch einen Scanner reihum abgefragt werden. Dies kann zur Folge haben, dass die einzelnen Eingabekanäle nicht in der Reihenfolge zum Rechner übertragen werden, in der sie zur Übertragung bereit waren. Die damit zusammenhängenden Probleme der Verkoppelung von Datenquellen z.B. bei Koinzidenzexperimenten sind jedoch berücksichtigt (Siehe [\[Kopplungskarte <-> Interface-Steuerungen\]](#), Seite [\(undefiniert\)](#)).

Da den 4 Bits der Scanner-Adresse nur 3 Bits der Steckplatzadresse gegenüberstehen, kann das niederwertigste Bit der Scanner-Adresse zur Adressierung zweier Register verwendet werden. Die Standard-Interface-Steuerung (IFS-Karte) macht bei der Lifetime-Messung von dieser Möglichkeit Gebrauch.

Die von der Interface-Steuerung zugeführten Signale *ReadyB* und *PrioReqB* werden sorgfältig einsynchronisiert, so dass sie asynchron zu den internen Vorgängen auftreten dürfen.

Das Protokoll mit den Eingabekanälen und der Rechneranpassung ist auf der Routing-Steuerung als Zustandsautomat realisiert (Abb. 7.2.5.1, 7.2.5.2):

- Im Leerlauf durchläuft der Zustandsautomat die Zustände Z0-Z1-Z3. Beim Übergang nach Z1 wird von der Routing-Steuerung durch die Taktsignale *TN* bzw. *TV* der Vergleich von Scanner-Adresse und Steckplatzadresse freigegeben (A2). Ist ein Eingabekanal angewählt und bereit zur Datenübertragung, so teilt er dies der Routing-Steuerung durch das Signal *ReadyB* über den Bus mit.

Bei der Rückkehr nach Z0 wird die nächste Scanner-Adresse geschaltet und das Vorrangsignal (*PrioReqB*) einsynchronisiert (A4).

- Antwortet ein Eingabekanal mit einer Data-Ready-Meldung (*ReadyB*), so wird diese beim Übergang nach Z2 mit A3 einsynchronisiert (*ReadyR*) und in Z2 zur Datenübertragung verzweigt (Z21, Z22). Dabei wird das Abfragesignal (*TN* bzw. *TV*) erneuert (A5, A2).

Ferner wird *ReadyR* an die Rechneranpassung weitergegeben zur Anmeldung einer Datenübertragung. Sobald die Rechneranpassung bereit ist, ein Datenwort aufzunehmen, meldet sie sich mit einem Data-Request-Signal (*RequestB*) und veranlasst damit ein Öffnen der Datengatter des angewählten Eingabekanals (Quellen-Interface). Aus den 16 Daten-Bits, dem LFT- und Kopp-Bit und den 4 Bits der Scanner-Adresse setzt

die Rechneranpassung ein Datenwort zusammen (Siehe $\langle \text{undefined} \rangle$ [Rechneranpassung], Seite $\langle \text{undefined} \rangle$.) und überträgt es zum Rechner. Der Empfang der Daten wird von der Rechneranpassung mit den Data-Accepted-Signalen $AcceptedB$ und $AcceptedR$ quittiert. Auf der Routing-Steuerung bewirkt die Rückflanke von $AcceptedR$ ein Weiterstarten des Daten-Scanners unabhängig davon ob $ReadyB$ noch ansteht oder nicht. Die Datenquelle wird durch $AcceptedB$ im allgemeinen zu einer Normierung veranlasst, zur Vorbereitung auf das nächste Datenwort.

In Z21 wird auf das Data Accepted ($AcceptedR$) der Rechneranpassung gewartet und in Z22 auf dessen Ende.

Bei der Rückkehr nach Z0 wird die nächste Scanner-Adresse geschaltet und das Vorrangsignal ($PrioReqB$) einsynchronisiert (A4).

- Wenn in Z0 festgestellt wird (E4), dass das Vorrangsignal sich geändert hat, so wird ein Wechsel des Daten-Scanners durchgeführt (Z01, Z02, Z03). Dabei werden die vier Adress-Bits nacheinander umgeschaltet, um Störungen auf dem Bus gering zu halten (A6, A7, A8, A9).

Timing:

Der Zustandsautomat wird mit 10MHz getaktet. Daraus ergibt sich folgendes Timing (zum Vergleich die Werte für die ausgemusterte Routingsteuerung).

		<i>neu</i>	<i>alt</i>
Adressfortschaltung im Leerlauf	A =	300ns	400ns
Datenübertragung	D =	400ns	500ns
Scanner-Prioritätswechsel	P =	400ns	0ns
<i>4 Gruppen, 1 Sanner-Umlauf:</i>			
Leerlauf	(16A)	4800ns	6400ns
1 Prio 0 Datenübertragung	(16A + D)	5200ns (192kHz)	7100ns (141kHz)
1 Prio 1 Datenübertragung	(16A + 2P + D)	6000ns (167kHz)	7100ns (141kHz)
<i>2 Gruppen, 1 Sanner-Umlauf:</i>			
Leerlauf	(8A)	2400ns	
1 Prio 0 Datenübertragung	(8A + D)	2800ns (357kHz)	
1 Prio 1 Datenübertragung	(8A + 2P + D)	3600ns (278kHz)	
<i>1 Gruppe, 1 Sanner-Umlauf:</i>			
Leerlauf	(4A)	1200ns	
1 Prio 0 Datenübertragung	(4A + D)	1600ns (625kHz)	
1 Prio 1 Datenübertragung	(4A + 2P + D)	2400ns (417kHz)	

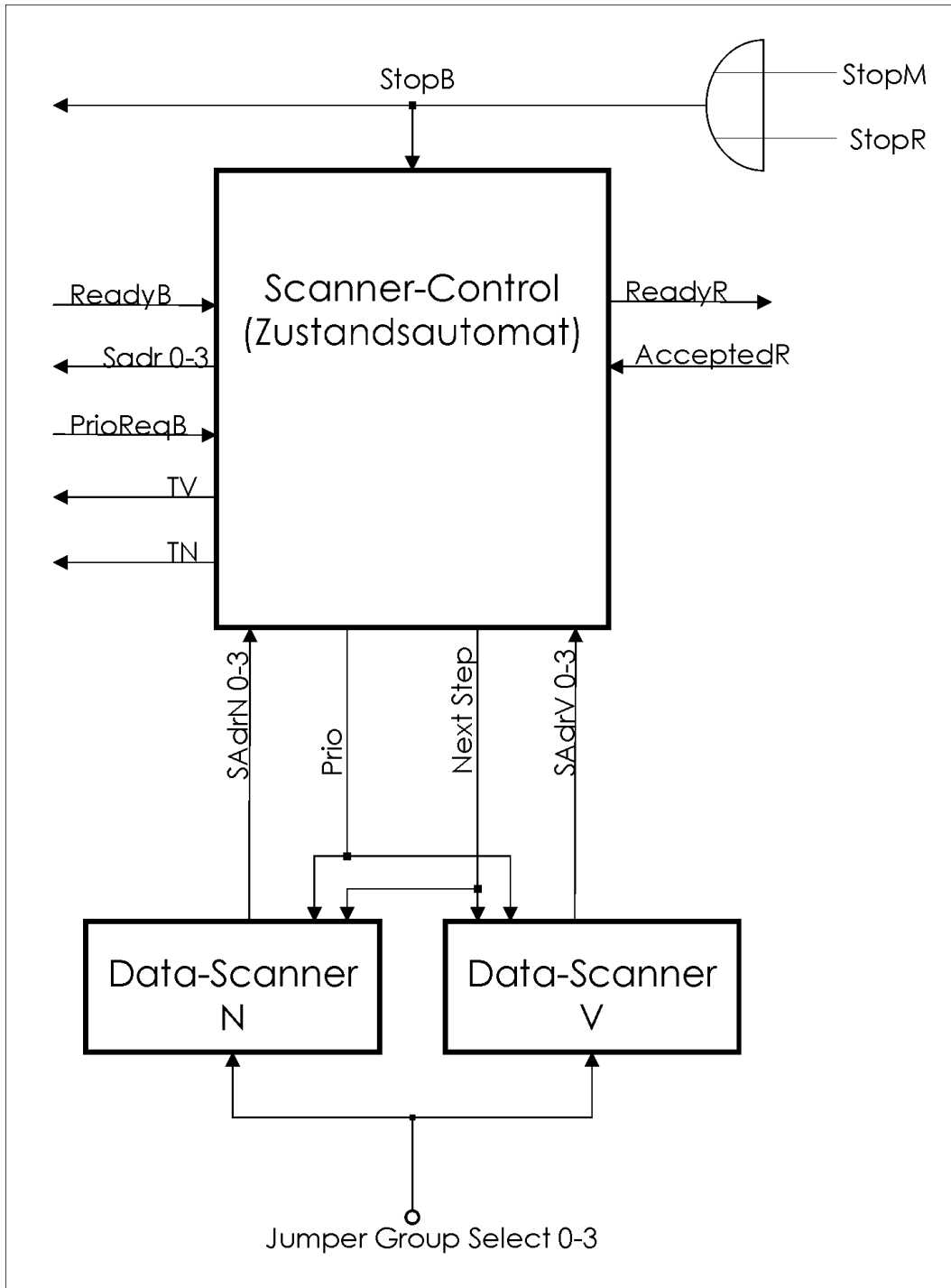
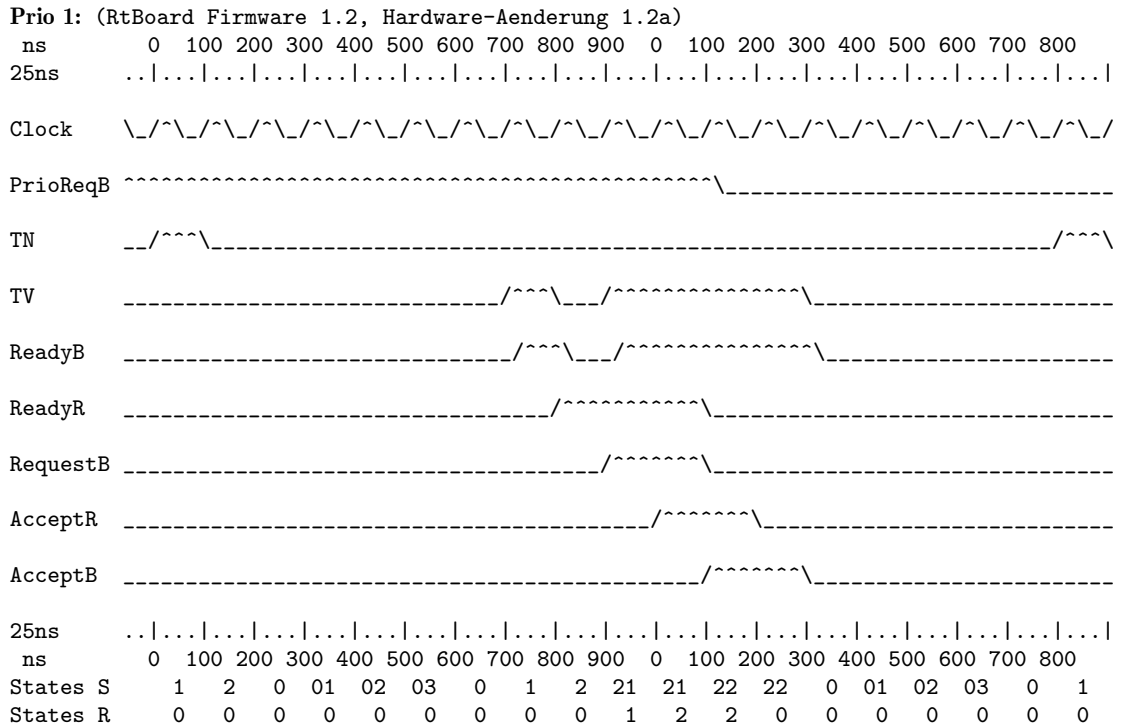


Abb. 7.2.5.1 Routingsteuerung

Abb. 7.2.5.3c Data Transfer Timing IFS1 Board / Control Board / Routing Board



7.2.6 Routing-Steuerung <-> Rechneranpassung

Die Routing-Steuerung verständigt sich mit der Rechneranpassung durch einen Hand-Shake-Dialog, zum Teil über Leitungen des Privat-Busses nach rechts. Die Routing-Steuerung muss deshalb stets direkt links von der Rechneranpassung stecken. Beide Karten sollten vorzugsweise ganz rechts im Überrahmen untergebracht werden. (Siehe Abschnitt 2.2 [Überrahmen], Seite 4.).

Signalnamen ohne Backslash '\': aktiv high

Signalnamen mit Backslash '\': aktiv low

Ein '^' vor dem Signalnamen: die Routing-Steuerung ist die Quelle des Signals

^ReadyR (Data Ready)

Mit diesem Signal wird die Rechneranpassung von der Routing-Steuerung aufgefordert, ein Datenwort zu übernehmen. Es steht solange an bis die Rechneranpassung *AcceptedR* sendet.

RequestB (Data Request)

Mit diesem Signal zeigt die Rechneranpassung dem angewählten Eingabekanal an, dass das Datenwort auf den Bus zu schalten ist.

AcceptedB (Data Accepted)

Mit diesem Signal an den Eingabekanal quittiert die Rechneranpassung den Empfang des Datenwortes.

AcceptedR (Data Accepted)

Mit diesem Signal von der Rechneranpassung wird der Routing-Steuerung mitgeteilt, dass die Übertragung beendet ist. Diese nimmt *ReadyR* zurück und startet den aktuellen Scanner nach dem Ende von *AcceptedR*.

StopR

Stop der Routing-Steuerung durch den Rechner.

^SAddr0..3 (Scanner-Adresse)

Die Scanner-Adresse wird von der Rechneranpassung dem Datenwort als Quellenkennung beigefügt (Siehe [\[Rechneranpassung\]](#), Seite [\[unbestimmt\]](#)).

Tab. 7.2.6.1 Privat-Bus-Belegung (RST-Bus)

Signal	Leitung	RST	RAP
AcceptedR	24	c	a
StopR	25	c	a
ReadyR	27	c	a

7.2.7 USB Anpassung

Mit dem UpGrade auf *VxWorks 7* werden die MVME-Rechner nicht mehr unterstützt, es besteht jedoch die Möglichkeit für eine Rechneranpassung über USB, so dass PCs als Laborrechner zum Einsatz kommen können.

Die USB-Anbindung wurde mit einem *FTDI FT2232H MINI MODULE* realisiert. Aus der Auswahl von Industriestandards, die der *FT2232H* IC bietet, ist insbesondere der folgende wegen der großen FIFOs für die Rechneranpassung geeignet:

```
Asynchronous FIFO Interface Mode
Dual Port FIFO TX Buffer (4 Kbytes per interface)
Dual Port FIFO RX Buffer (4 Kbytes per interface)
```

7.2.7.1 FT2232H Mini Modul (USB)

Der FT2232H Baustein des Mini Moduls muss mit Hilfe des *FTDI-FT_PROG* Windows-Programmes über USB konfiguriert werden.

Mit '*DEVICES -> Scan and Parse*' wird der Ist-Zustand des EEPROM ausgelesen und kann editiert werden.

Und mit '*DEVICES -> Program*' wird der EEPROM programmiert.

Mit '*File -> Save As Template*' kann die Konfiguration eines bereits programmierten FT2232H auf einen neuen übertragen werden.

Die Data-Routing Konfiguration unterscheidet sich vom Control-Routing nur in der Serial Number:

```
USB Device Descriptor
  Custom VID/PID:   FTDI Default
  Vendor ID:       0403
  Product ID:      6010
  USB Version:     USB 2.0

USB Config Descriptor -> bmAttributes
  Bus Powered:     yes
  Self Powered:    no
  Max Bus Power:   150 mA
  USB Remote Wakeup: no
  Pull Down IO...: no

USB String Descriptors
  Manufacturer:    FTDI
  Product Descr.:  FT2232H Mini Modul
  Ser. Num. Enabled: yes
  Auto Gen. Ser. No: no
  Serial Number:   IAMP-DataRout
  Ser. Num. Prefix:

Hardware Specific
  DPRDRV          : 0
  Suspend on DBus7 Low: no
```

```

Port A/B -> Hardware: 245 FIFO
Port A/B -> Driver:   Virtual COM Port
I/O Pins -> Group.. -> Slow Slew: no
I/O Pins -> Group.. -> Drive:     8mA
I/O Pins -> Group.. -> Schmitt...: no

```

Durch eine Fehlbedienung kann der EEPROM unbrauchbar werden. Dann hilft nur ein 'Disaster Recovery'. Von FTDI gibt es eine Anleitung dazu. Benötigt wird ein intaktes Mini Modul und das defekte muss vermutlich aus der Schaltung raus...

7.2.7.2 Rechneranpassung (USB)

Das Mini Modul bietet zwei Ports. Über Port-1 erfolgt die Kontrolle des Datentransfers und die Statusabfrage. Port-0 dient dem Datentransfer.

Die Rechneranpassung verfügt über eine *StandAlone*-Funktion um das Data-Routing für Tests ohne USB-Anschluss betreiben zu können. Dabei werden alle Datenquellen permanent abgefragt wenn der USB-Anschluss fehlt.

```

" ***** Data Routing Control Port Programmierung *****
" Ein Kommando an den Control Port besteht aus einem Byte, in dem nur die
" drei hoechstwertigen und die zwei niederwertigsten Bits ausgewertet werden:
"   "***** Control commands"
"   [x,x,x,x,x,x,0,0] "Send status"
"   [x,x,x,x,x,x,1,1] "Send status"
" Es werden zwei Bytes zurueck gesendet:
" 1. Header Byte: [0,1,0,0,0,0,1,0]
" 2. Status Byte: [r,s,f,m,x,x,x,x]
" r = 1: 'Reset Routing' ist gesetzt
" s = 1: 'Stop Data Transfer' ist gesetzt
" f = 1: 'FPGA Testdaten' ist gesetzt
" m = 1: manueller Stop Datentransfer
"
"   [r,s,f,x,x,x,0,1] "Clear control bits"
" r = 1: 'Reset Routing' wird geloescht
" s = 1: 'Stop Data Transfer' wird geloescht
" f = 1: 'FPGA Testdaten' wird geloescht
"
"   [r,s,f,x,x,x,1,0]; "Set control bits"
" r = 1: 'Reset Routing' wird gesetzt
" s = 1: 'Stop Data Transfer' wird gesetzt
" f = 1: 'FPGA Testdaten' wird gesetzt
" *****
" ***** Data Routing Data Port Programmierung *****
" * Das einzige Kommando an den Data Port besteht nur aus einem Byte
"   um eine Pause fuer den Datentransfer ein und aus zu schalten um
"   Buffer-Ueberlaeufe zu verhindern.
"   [x,x,x,x,x,x,1] Transferpause einschalten
"   [x,x,x,x,x,x,0] Transferpause ausschalten
"

```

```

"
" * Der Datentransfer des Data Ports besteht aus vier Bytes:
"   1. Header Byte: [0,1,1,0,0,0,1,0]
"
"   2. Status Byte: [F,0,I,I,I,I,I,I]
"   F == 1: USB Fifo overflow ???
"   [I,I,I,I,I,I]: Data ID des uebertragenen Datenwortes
"
"   3. Data High Byte:
"   4. Data Low Byte:
" Als Datenwort wird das Routing Datenregister uebertragen.
" *****

```

7.2.8 VME Anpassung

7.2.8.1 Rechneranpassung (VME)

Die VME Rechneranpassung ist mit einem 2048 Wort tiefen und 3 Byte breiten Fifo ausgerüstet, um vorübergehende Unterbrechungen der Datenübertragung (z.B. durch Störungen) überbrücken zu können. Der Zustand des Fifo (halbvoll, voll), nach dem ein Datenwort gespeichert wurde, wird jedem Datenwort als Status-Bits mitgegeben. Ferner steht der aktuelle Zustand des Fifo (leer, halbvoll, voll, Fehler) stets dem Rechner zur Abfrage zur Verfügung.

Mittels der Zustandssignale (leer, halbvoll, voll) der drei Fifo-Bausteine wird überwacht, ob die drei Bausteine noch synchron sind. Im Fehlerfall wird ein Status-Bit gesetzt, das von der Software nur durch ein 'Stop Datenübertragung' gelöscht werden kann. Diese Überwachung erkennt einen Fifo-Fehler u. U. nicht sofort, sondern erst, wenn die Zustands-Flags der drei Bausteine nicht mehr gleich sind.

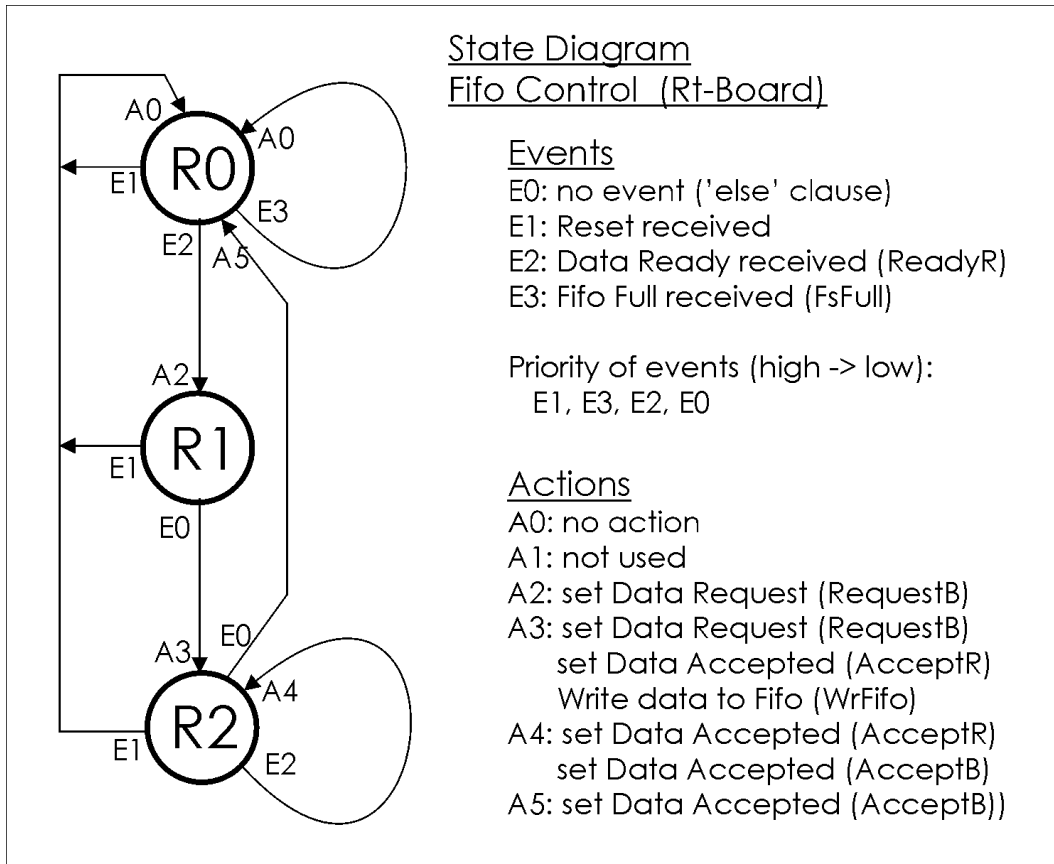


Abb. 7.2.7.1 Zustandsdiagramm Fifo Control

7.2.8.2 Rechneranpassung <-> Rechner-Interface (VME)

Hardware-Verbindung Rechneranpassung -> Rechner-Interface

Die Datenübertragung von der Rechneranpassung zum Rechner-Interface geschieht über ein 50-poliges Flachbandkabel. Tests mit einem 40m langen Kabel haben gezeigt, dass bis zu dieser Länge noch eine gesicherte Übertragung möglich ist. Möglichst kurze Kabellängen sind jedoch zu empfehlen, da die Kabellänge in die Übertragungsgeschwindigkeit eingeht (s.u.) und wegen höherer Störungssicherheit.

Tab. 7.2.8.1 Steckerbelegung 50 pol. Verbindungskabel

Pin	Signal	Pin	Signal
1,2,...,50	Ground		Ground
3	Fifo full	27	Routing online P
5	Fifo failure	29	Routing stopped
7	Fifo half full	31	Data Available bit 1
9	Routing online N	33	Data Available bit 0
11	not used	35	Data 7
13	not used	37	Data 6
15	Data Request bit 0	39	Data 5
17	Stop Routing	41	Data 4
19	Data Request bit 1	43	Data 3
21	IP-Board ID bit 2	45	Data 2
23	IP-Board ID bit 1	47	Data 1
25	IP-Board ID bit 0	49	Data 0

Verbindungs-Check Rechneranpassung -> Rechner-Interface

Um Verwechslungen der IP-Boards (Rechner-Interfaces) für Data-Routing und Control-Routing zu erkennen (sie unterscheiden sich nur in unterschiedlicher Firmware), bieten die IP-Boards der Rechneranpassung eine 3 Bit Board-ID an, die mit der Board-ID der Rechneranpassung verglichen wird. Die Rechneranpassung schaltet nur auf Online wenn beide IDs gleich sind. Das Ergebnis des Vergleichs wird dem Rechner-Interface über die komplementären Leitungen 'Routing online P/N' mitgeteilt und kann vom Rechner als Status-Bit abgefragt werden. Ein negatives Ergebnis (Routing offline) entsteht ebenfalls, wenn das Routing abgeschaltet ist oder das Verbindungskabel fehlt.

Data Transfer Protokoll Rechneranpassung -> Rechner-Interface

Rechneranpassung:	Rechner-Interface:
Sender	Empfänger
Data Transfer Slave	Data Transfer Master
10 Mhz Clock	8 Mhz Clock

Ein vollständiger Übertragungszyklus (Abb. 7.2.8.2, 7.2.8.3) überträgt nacheinander drei Bytes (16 Bits Daten, 8 Bits Status). Der Empfänger (Rechner-Interface) zeigt mit 'DataRequestN' (N = 1,3,2) an, dass er zur Aufnahme von Daten-Byte N bereit ist. Der Sender (Rechneranpassung) antwortet mit 'DataAvailableN' und der Übertragung von

Daten-Byte N. Mit 'DataRequest0' und 'DataAvailable0' wird die Übertragung beendet. Erst wenn 'DataRequest0' UND 'DataAvailable0' anstehen kann eine neue Übertragung starten. 'DataRequest1' wird vom Sender ignoriert, solange keine Daten vorhanden sind.

N ist mit jeweils zwei Leitungen binär kodiert. Die Reihenfolge von N ist so gewählt ($N = 1,3,2$), dass sich immer nur ein Bit ändert, um bei Flankenüberschneidungen falsche N zu vermeiden.

Das Eintreffen eines unerwarteten 'DataRequestN' bzw. 'DataAvailableN' bei Sender oder Empfänger führt zu einem Abbruch der Datenübertragung durch 'DataRequest0' bzw. 'DataAvailable0'. Noch nicht gesendete Bytes eines Datenwortes gehen dabei verloren.

Die Übertragung wird überwacht und im Falle von Störungen wiederholt. Zur Erkennung von Störungen auf dem Verbindungskabel werden 3 (Sender) bzw. 2 (Empfänger) Leitungen mit konstanten und komplementären Signalpegeln auf Signaleinbrüche sowohl nach High als auch nach Low überwacht. Störungen werden dadurch erkannt und registriert, dass zwischen zwei !Clock-Signalen eine überwachte Leitung sowohl High als auch Low war. Das Ergebnis wird mit einem nachfolgenden Clock überprüft und gibt Auskunft ob die mit dem vorhergehenden Clock (zwischen den beiden !Clock) übernommenen Signale gestört waren. Im Störfall werden 'DataRequestN' bzw. 'DataAvailableN' solange aufs Neue übernommen, bis dies fehlerfrei geklappt hat.

Anfangs wurde ein deutlich schnelleres, jedoch nicht ausreichend fehlersicheres Übertragungsprotokoll verwendet:

Die Übertragung eines Datenwortes ($3 * 1\text{Byte}$) startet mit dem Senden vom 'DataRequest'. Die Übertragung der Bytes erfolgt mit 3 Strobe-Signalen 'DataAvailable' (200ns, Pause 200ns). Um mögliche Störungen der Handshake-Signale durch den Datentransfer zu vermeiden, wird 'DataAvailable' sowohl mit Clock als auch mit !Clock übernommen und nur weiter verarbeitet, wenn beide Ergebnisse gleich sind. Bei länger andauernden Störungen kann dabei jedoch ein 'DataAvailable' völlig verloren gehen, was mit einem Timer überwacht werden muss und zu einem Datenverlust führt.

Die neue, fehlersichere Übertragung mit Handshake-Protokoll benötigt mehr Zeit als die alte mit Strobe-Signalen und darüberhinaus geht die Kabellänge in die Übertragungszeit ein. Folgende Zeiten werden für den Transfer von der Rechneranpassung zum Rechner-Interface (Anfang erstes bis Ende letztes Data-Byte) benötigt. Durch das Einsynchronisieren auf die unterschiedlichen Clock-Signale von Rechneranpassung und Rechner-Interface entstehen min/häufigste/max Zeiten:

altes Protokoll:	1200ns	Handshake-Laufzeiten
neues Protokoll 1m Kabel:	1600/1700/(1800)ns	$6 * 1\text{m} * 5\text{ns/m} = 30\text{ns}$
neues Protokoll 5m Kabel:	1700/1800/1900ns	$6 * 5\text{m} * 5\text{ns/m} = 150\text{ns}$
neues Protokoll 40m Kabel:	2900ns	$6 * 40\text{m} * 5\text{ns/m} = 1200\text{ns}$

EMV-Tests:

Durch periodisches Ein- und Ausschalten einer Weller 21021 Lötstation wurden in einem halb drumherum gelegten 50-pol., ca. 5m Verbindungskabel EM-Störungen erzeugt und deren Auswirkungen mit VMETST, Test Data-Routing, 10^7 Transfers, inkrementierend, beobachtet.

Fehlertyp 1: Wortfehler in der Form:

n: expected: ABCD read: ABCD+1

Fehlertyp 2: Bitfehler in der Form:

n: expected: ABCD read: XYCD

n+1: expected: XYCD+1 read: ABCD+1

Fehlertyp 3: Data Errors:

beim Transfer gingen Bytes verloren

Fehlertyp 4: Data Errors Falschmeldung:

da als Status 0xffff gelesen wird, wird fälschlicherweise
ein Data Error gemeldet

Ergebnisse für altes (schneller) und neues (verbesserte Fehlersicherung)
Transfer-Protokoll und die verschiedenen IP-Carrier (10⁷ Transfers):

IP-Carrier	Protokoll	Typ1	Typ2	Typ3+4
VIPC616	alt	10	2	0
VIPC616	neu	1,4	1,4	0
IPC01	neu	2	5	1

Weitere Tests haben gezeigt, dass das IPC01 deutlich empfindlicher ist und insbesondere den Typ 4 auch ohne laufenden Daten-Transfer immer wieder bringt. Dabei ist sehr häufig das Auslesen des Status durch den Computer gestört (Status = 0xffff), was dann fälschlicherweise als Data-Error interpretiert wird (nur IPC01)! Passiert auch ohne Datentransfer und mit offenem Kabel!

Einige Windungen des Kabels um einen Ferrit-Ring bringen möglicherweise eine Verbesserung.

Das Verbinden der Abschirmflächen unter den IP-Boards mit Masse (Lötunkte für Brücken waren vorgesehen, aber nicht gebrückt) bringt möglicherweise eine Verbesserung. Das Einlöten des zweiten Steckers zur IPC01 Seite hin (mit geerdeten Abschirmflächen) beseitigt schließlich diesen Fehlertyp 4 weitgehend, mit Ferrit-Ring (3 Windungen) auf der IPC01-Seite sogar vollständig! Danach taucht aber der Typ 2 offensichtlich verstärkt auf, und ebenso immer wieder FIFO Errors! Der geänderte VMETST liest jetzt die Daten zweimal aus und vergleicht. Dies hat gezeigt, dass beim Auslesen keine Fehler auftreten, sondern die Daten schon falsch übertragen werden. Das Problem hat sich jetzt möglicherweise auf die Routing Seite verlagert?! Ein Ferrit-Ring (3 Windungen) auf der Routingseite vermindert die Fehler deutlich (10*⁸ Transfers, neues Protokoll, 5m Kabel):

(Wk = kleiner Ferrit, Wg = großer Ferrit)

Carrier	Ferrit	Typ1	Typ2	Typ3	Typ4	FIFO-Errors
IPC01	kein	2*10	14*10	1*10	0	ja
IPC01	3Wg IPC01	45	51	0	0	?
IPC01	3Wg Routing	1	0	0	2	nein
IPC01	1Wk IPC01+ 3Wg Routing	4	1	0	0	nein
IPC01	3Wg IPC01+ 3Wg Routing	1	6	0	1	ja
IPC01	1Wg IPC01+ 3Wg Routing	1	5	0	3	nein
IPC01	1Wg IPC01+ 1Wg Routing	viele	viele	0	0	mehrere

Der Erfolg hängt offensichtlich auch von der Anordnung der Ferrits und dem Typ ab.

Großer, grüner Ferrit:						
IPC01	4Wg Routing	2	6	0	6	nein
Großer, schwarzer, zerbrochener Ferrit:						
IPC01	4Wg Routing	ja	ja	ja	ja	ja

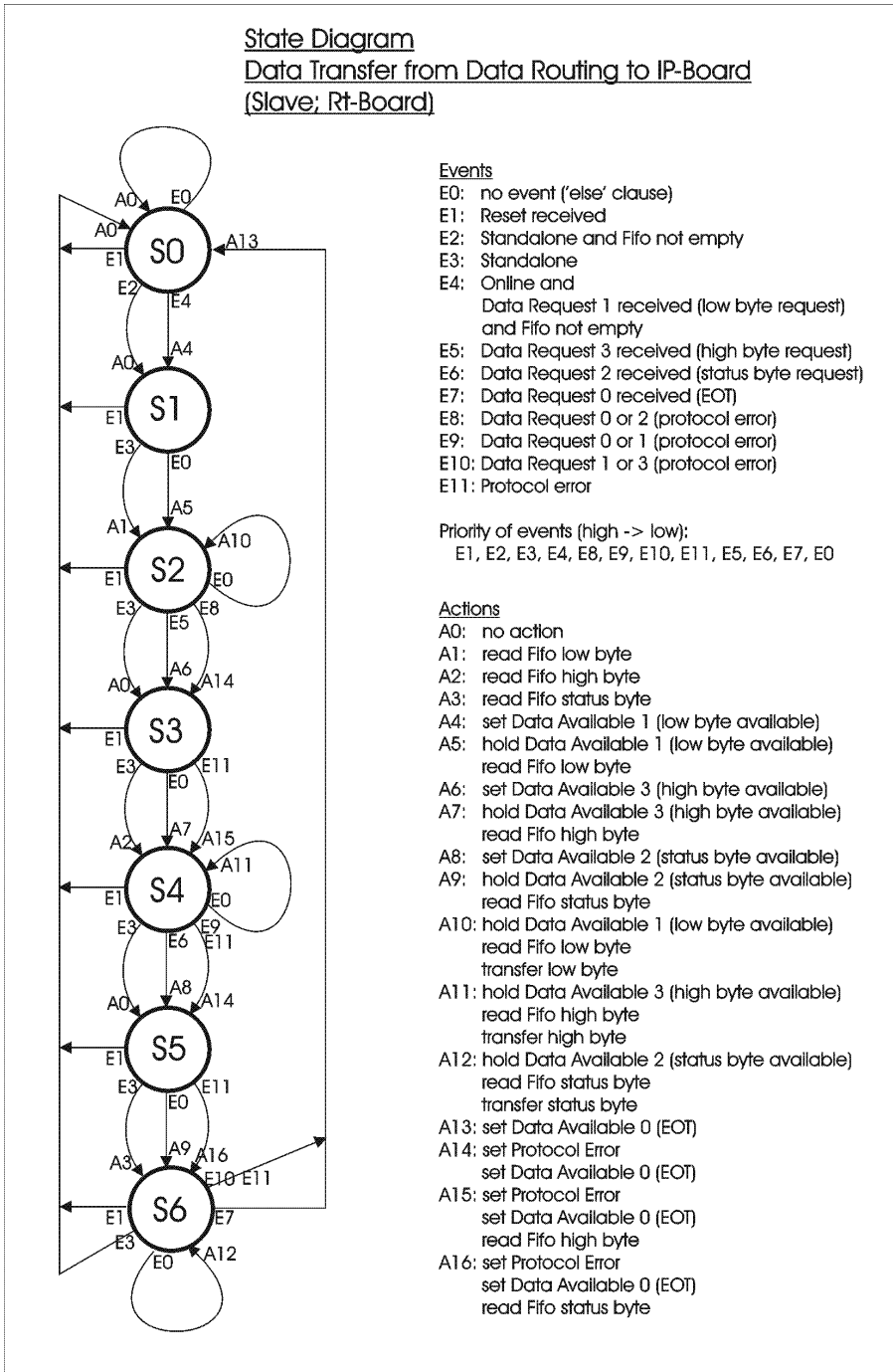


Abb. 7.2.8.2 Data Transfer Zustandsautomat Rechneranpassung

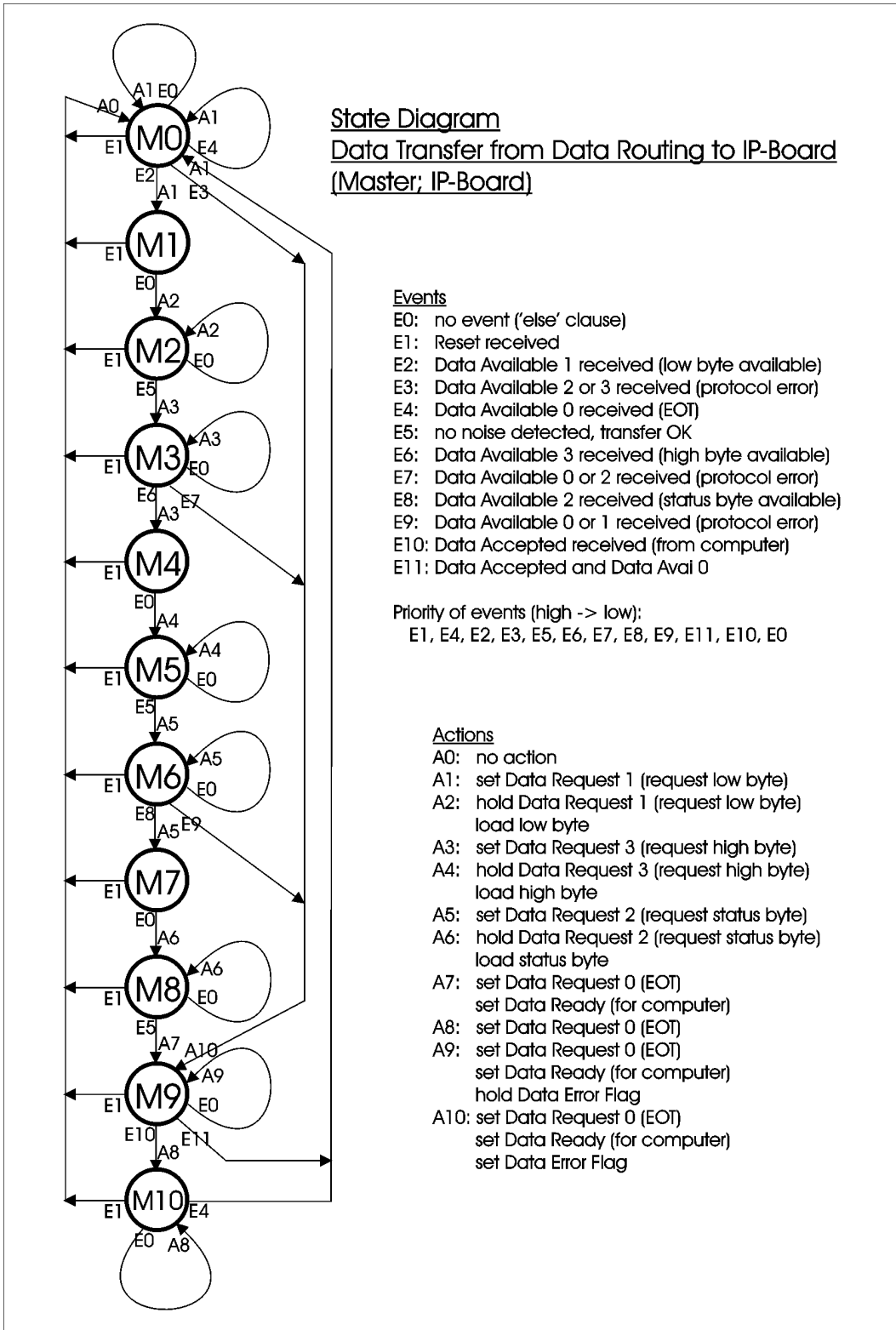


Abb. 7.2.8.3 Data Transfer Zustandsautomat Rechner-Interface

7.2.8.3 Rechner-Interface (VME)

Das rechnerseitige Interface ist auf einem doppelten IP-Board (Industrie Pack) untergebracht. Es wird das gleiche IP Board Layout verwendet wie für das Control-Routing Interface, und unterscheidet sich lediglich in der Programmierung des ispLSI-Bausteins. Industrie Pack Carrier gibt es für eine ganze Reihe verschiedener Plattformen (z.B. VME, PC). Zur Zeit sind VME-Boards als Carrier im Einsatz (GreenSpring VIPC616, MicroSys IPC01, Motorola CPU-Boards MVME162, MVME172, MVME172P).

Das Rechner-Interface empfängt von der Routing-Seite 2 Daten-Bytes (Data00-15) und ein Status-Byte (Data16-23), das den Status zur Zeit der Entstehung des Datenwortes enthält (Status00-07). Bei der Übertragung zum Rechner wird ein weiteres Status-Byte hinzugefügt, das den gerade aktuellen Status enthält (Status08-15).

Das IP-Board besitzt je ein 16Bit-Register für Status und Daten, die mit und ohne Löschen vom Rechner gelesen werden können. Solange sie nicht gelöscht sind, werden keine neuen Daten vom Routing übertragen.

Die Ein- und Ausgabe zu dem IP-Board erfolgt in 16 Bit Worten. Ein gültiges Datenwort und Routing-seitiges Status-Byte kann nur gelesen werden, wenn im Statuswort folgende Bedingungen erfüllt sind:

- NOT Data transfer error
- Routing online
- NOT Data transfer stopped
- Data ready
- NOT Data transfer busy
- NOT Fifo failure

Tab. 7.2.9.1 Programmierung des Rechner-Interfaces

```

*** Address decoding:
  A2 A1 A0   (Read)
    0  0  0   Read Status only
    0  1  0   Read Status and clear Data
    1  0  0   Read Data only
    1  1  0   Read and clear Data
  A2 A1 A0   (Write)
    0  0  0   Write Control

*** Bit      Control word
    15       Stop data transfer
    14-00    not used

*** Bit   Data word   Status word
    15    Data15      Data transfer error
                        - die Daten sind bei der Übertragung verloren
                        gegangen
    14    Data14      Routing online
                        - das Routing ist angeschlossen und angeschaltet

```

13	Data13	Data transfer stopped - die Übertragung ist durch Rechner oder manuell gestoppt
12	Data12	Data ready - es steht ein Datenwort zum Lesen bereit
11	Data11	Fifo full (at Status read time) - das Fifo auf der Rechneranpassung meldet aktuell 'voll'
10	Data10	Fifo half full (at Status read time) - das Fifo auf der Rechneranpassung meldet aktuell 'halb voll'
09	Data09	Data transfer busy - es wird gerade ein Datentransfer von der Rechneranpassung zum IP-Board durchgeführt
08	Data08	Fifo failure - Funktionsstörung des Fifo auf der Rechneranpassung
07	Data07	Fifo full (at Data time) (Data23) - nach Ablage der Daten im Fifo war dieses voll
06	Data06	Fifo half full (at Data time) (Data22) - nach Ablage der Daten im Fifo war dieses mehr als halb voll
05	Data05	LFT-Bit (Data21) - Lifetime Kennungs-Bit
04	Data04	K-Bit (Data20) - Kopplungs Kennungs-Bit
03	Data03	Scanner addr. bit 3 (Data19) - Routing-Steckplatz-Kennungs-Bit 3
02	Data02	Scanner addr. bit 2 (Data18) - Routing-Steckplatz-Kennungs-Bit 2
01	Data01	Scanner addr. bit 1 (Data17) - Routing-Steckplatz-Kennungs-Bit 1
00	Data00	Scanner addr. bit 0 (Data16) - Routing-Steckplatz-Kennungs-Bit 0

subsection Kopplungskarte <-> Interface-Steuerungen

Der Kopplungs-Bus baut sich von der Kopplungskarte ausgehend nach links als Privat-Bus auf. Alle Karten sollten möglichst die Pins a und c der Leitungen 22, 23, 24 gebrückt haben, um den Kpl-Bus nicht zu unterbrechen.

LFTK: (Lifetime Kopplung, Leitung 22)

Über diese Leitung meldet die Kopplung ihre Totzeit. Es wird dazu das Lifetime-Signal *LFTQ* der zur Kopplungskarte gehörigen IFS-Karte verwendet, die im Modus *Frei* läuft ($K2=0$). Statt *BusyQ* übergibt die Kopplung jedoch *BusyK* an die IFS-Karte.

high: Kopplung ist frei

$$LFTK = BusyK \setminus * GateQ \text{ (falls verdrahtet)}$$

KoppK\: (Kopplungssignal, Leitung 23)

Über diese Leitung bietet die Kopplung den angeschlossenen IFS-Karten nach dem Start das Kopplungssignal an.

Low: Kopplung aktiv

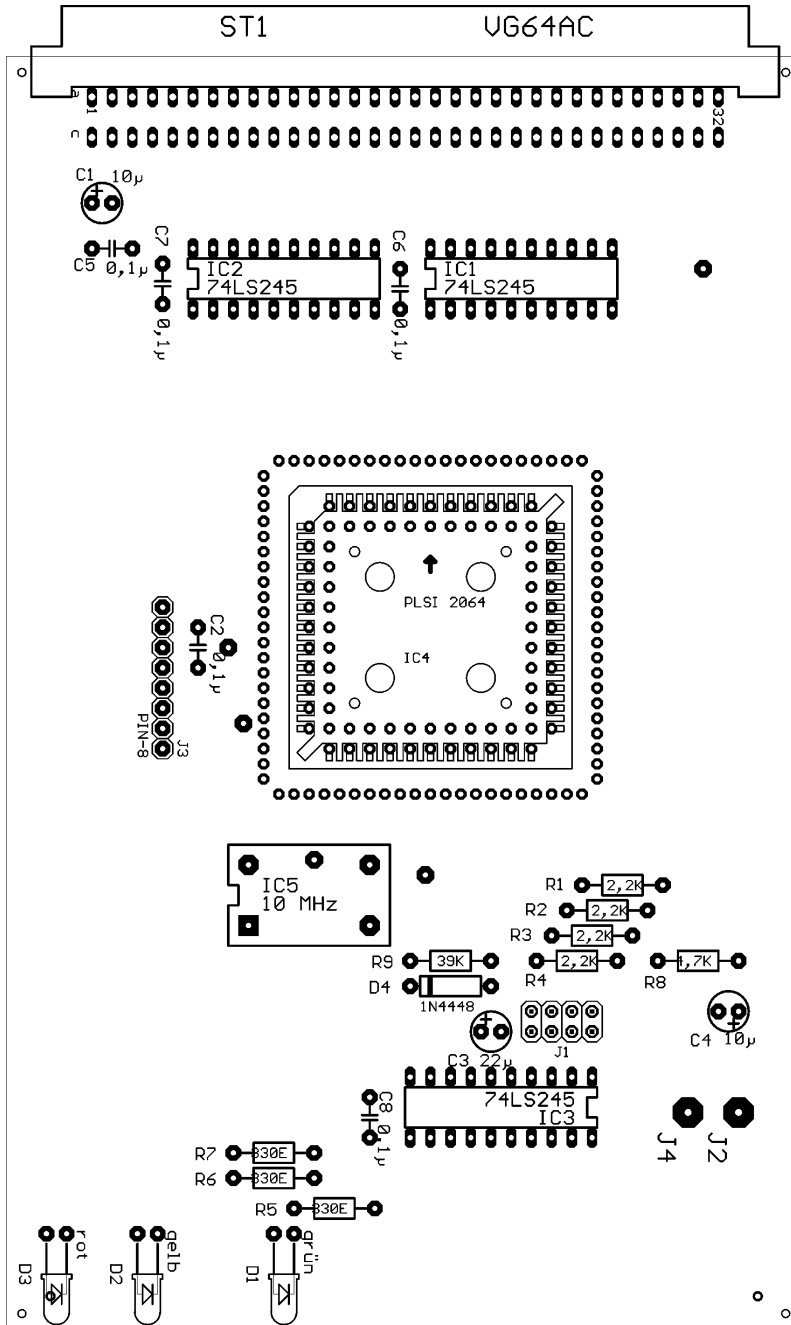
BusyK\: (Busy Kopplung, Leitung 24)

Auf diese Leitung geben die IFS-Karten der gekoppelten Quellen und die Kopplungskarte selber in 'wired-or'-Schaltung ihr Busy.

7.3 Schaltungsunterlagen

7.3.1 Board Interface-Steuerung

7.3.2 Board Routing-Steuerung



Last Update: 23.06.2000 08:26:20 Dateiname: Data Routing Control

Abb. 7.3.r.2 Layout Routing-Steuerung

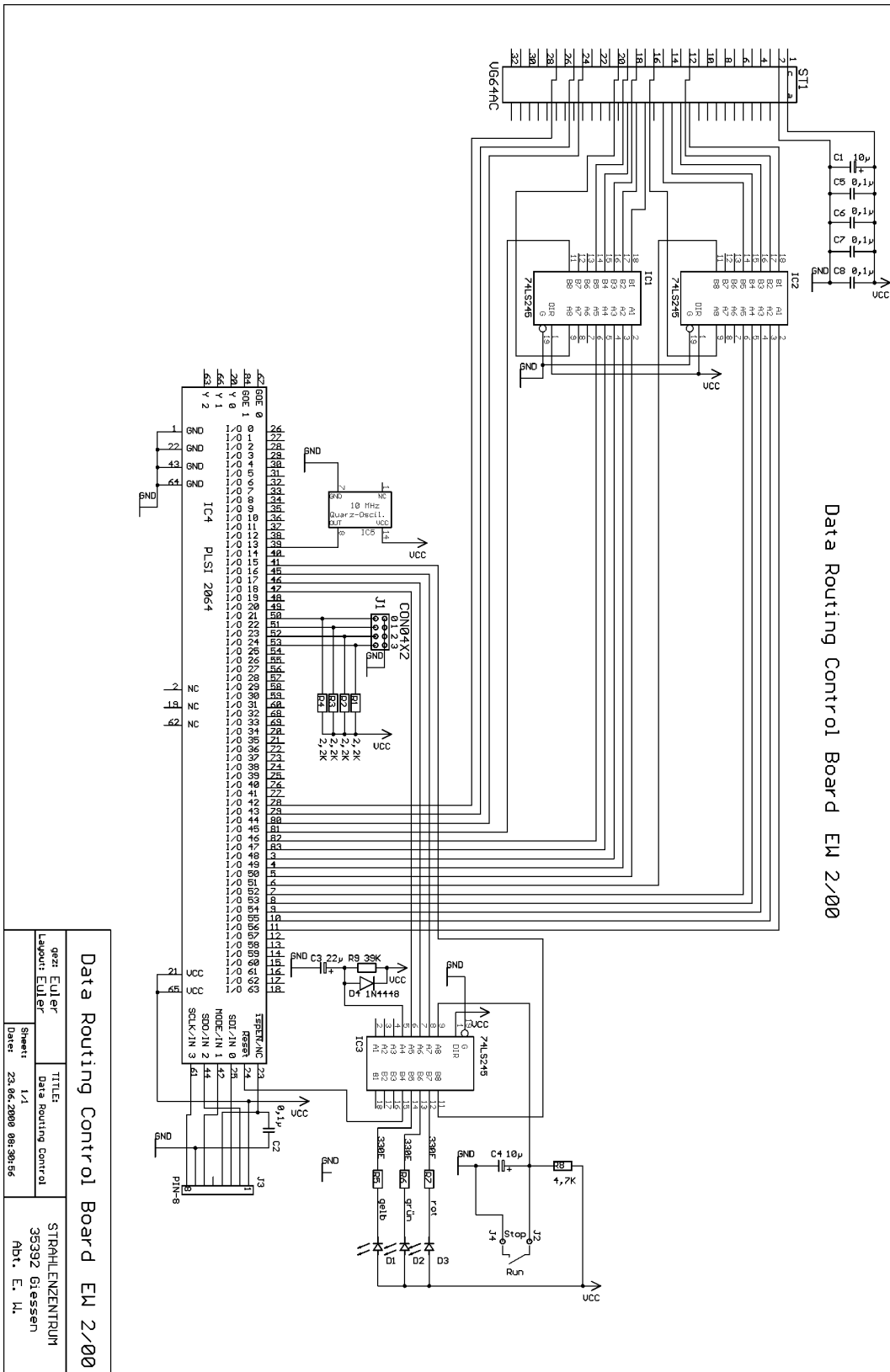


Abb. 7.3.r.1 Schaltplan Routing-Steuerung

7.3.3 Board Rechneranpassung

7.3.4 Board Rechner-Interface

8 Oldies

8.1 Alte Routing-Steuerung

Diese Variante der Routing-Steuerung war an TR86-, PDP11- und VME-Systemen im Einsatz. Sie ist inzwischen durch eine Karte mit einem hochintegrierten, programmierbaren Baustein und verbesserter Funktionalität ersetzt worden.

8.1.1 Funktion

Die Routing-Steuerung koordiniert die Eingabe der Daten von den bis zu acht anschließbaren Eingabekanälen. Ein Eingabekanal besteht aus den drei funktionellen Gruppen:

- Interface-Steuerung
- Datenquellen-Interface
- Datenquellen

Im allgemeinen sind diese Funktionen auf getrennten Karten untergebracht, bzw. die Datenquellen können auch eigenständige Geräte am Experiment sein (z.B. spektroskopischer ADC). Es sind jedoch auch Einplatinen-Lösungen möglich, bei deren Einsatz aber unbedingt darauf geachtet werden muss, dass sie verschiedene Steckplatznummern belegen (Abb. 3.1.0.1).

Die Routing-Steuerung ist mit zwei alternativ laufenden Scannern unterschiedlicher Priorität ausgestattet, mit denen die Eingabekanäle abgefragt werden. Sie schickt zu diesem Zweck über vier Bus-Leitungen (*SADR_x*) nacheinander die Scanner-Adressen 0-15 an alle Interface-Steuerungen. Diese vergleichen die drei höchstwertigen Bits der Scanner-Adressen mit ihrer Steckplatzadresse (*PADR_x*), die für jeden Steckplatz fest verdrahtet ist. Bei Gleichheit gilt der Eingabekanal als angewählt und kann bei Bedarf einer Datenübertragung mit der Routing-Steuerung kommunizieren.

Der Vergleich von Scanner-Adresse und Steckplatzadresse wird von der Routing-Steuerung durch die Taktsignale *TN* bzw. *TV* freigegeben. Ist ein Eingabekanal angewählt und bereit zur Datenübertragung (*ReadyQ*), so teilt er dies der Routing-Steuerung durch das Signal *ReadyB* über den Bus mit. Dies führt nach doppelter Einsynchronisierung der *Ready*-Meldung (*ReadyC*, *ReadyS*) zu einem Stop des Taktgenerators, und einer Erneuerung des, wegen der Laufzeiten durch die Interface-Steuerung bereits beendeten, *TN* bzw. *TV* Signals mittels *ReadyS*. Ferner wird das *ReadyS* an die Rechneranpassung weitergegeben zur Anmeldung einer Datenübertragung. Sobald die Rechneranpassung bereit ist, ein Datenwort aufzunehmen bzw. an den Rechner weiterzuleiten, meldet sie sich mit dem Signal *RequestB* und veranlasst damit ein Öffnen der Datengatter des angewählten Eingabekanals (Quellen-Interface). Aus den 16 Daten-Bits, dem LFT- und Kopp-Bit und den 4 Bits der Scanner-Adresse setzt die Rechneranpassung ein Datenwort zusammen (Siehe \langle undefined \rangle [Rechneranpassung], Seite \langle undefined \rangle .) und überträgt es zum Rechner. Der Empfang der Daten wird von der Rechneranpassung mit den Signalen *AcceptedB* und *AcceptedR* quittiert. Auf der Routing-Steuerung bewirkt nach Einsynchronisierung von *AcceptedR* die Rückflanke von *AcceptedS* ein Weiterstarten des Taktgenerators unabhängig davon ob *ReadyB* noch ansteht oder nicht. Die Datenquelle

wird durch *AcceptedB* im allgemeinen zu einer Normierung veranlasst, zur Vorbereitung auf das nächste Datenwort.

Es besteht die Möglichkeit, einen oder mehrere Eingabekanäle mit Vorrang vor den anderen abfragen zu lassen. Dazu müssen diese Eingabekanäle einen Vorrangwunsch (*VANM*) bei der Routing-Steuerung anmelden, sobald sie bereit sind ein Datenwort zu übertragen. Diese Anmeldung kann zu jeder Zeit asynchron erfolgen. Nach jedem Scanner-Schritt *TZ* prüft die Routing-Steuerung die *VANM*-Leitung mit *TUV* ab und setzt den Betrieb mit dem gewünschten Scanner (Vorrang- bzw. Normal-Scanner) fort. Der jeweils ruhende Scanner behält seine Position bei, so dass er an der unterbrochenen Stelle weiterlaufen kann. Der im Vorrangmodus arbeitende Eingabekanal muss sich des Freigabetaktes *TV* bedienen im Gegensatz zu *TN* für den Normalmodus.

Ein Stop der Routing-Steuerung kann vom Rechner her erfolgen (*StopR*) oder über einen Schalter auf der Frontplatte. Beide bewirken ein Anhalten des Taktgenerators. Ferner wird der Stop den Eingabekanälen mitgeteilt (*StopB*) und über die Interface-Steuerung den Datenquellen.

Die Scanner laufen mit 2.5MHz um, jeder Eingabekanal erhält also im Leerlauf alle 6.4us zwei Abfragen (eine mit *SADR0=0* und eine mit *SADR0=1*). Soll ein Datenwort übertragen werden, so hält der Scanner für eine Zeit an, die im wesentlichen durch das Timing des Rechners oder der Rechneranpassung, falls diese einen FiFo Speicher besitzt, bestimmt wird (ca. 1 us).

Es ist zu beachten, dass die verschiedenen Eingabekanäle von der Routing-Steuerung nicht nach einem 'first in - first out' Verfahren bearbeitet werden, sondern durch einen Scanner reihum abgefragt werden. Dies kann zur Folge haben, dass die einzelnen Eingabekanäle nicht in der Reihenfolge zum Rechner übertragen werden, in der sie zur Übertragung bereit waren. Die damit zusammenhängenden Probleme der Verkoppelung von Datenquellen z.B. bei Koinzidenzexperimenten sind jedoch berücksichtigt (Siehe (undefined) [Kopplungskarte], Seite (undefined)).

Da den 4 Bits der Scanner-Adresse nur 3 Bits der Steckplatzadresse gegenüberstehen, kann das niederwertigste Bit der Scanner-Adresse zur Adressierung zweier Register verwendet werden. Bei Verwendung der Standard-Interface-Steuerung (IFS-Karte) macht diese bei der Lifetime-Messung von dieser Möglichkeit Gebrauch.

Alle von außen zugeführten Signale werden sorgfältig einsynchronisiert (meistens doppelt, um Schaltspitzen durch Flankenüberschneidungen zu vermeiden), so dass sie asynchron zu den internen Vorgängen zugeführt werden können.

8.1.2 Bedienungselemente, Anzeigen

Schalter: Run/Stop

In Stellung *Stop* wird die Abfrage der Eingabekanäle angehalten durch Stoppen des Taktgenerators. An die Eingabekanäle geht ein Stoppsignal. Der Schalter ist disjunktiv (ODER) verknüpft mit dem Stop vom Rechner.

Anzeige: Run

Wenn weder vom Rechner noch durch den Schalter *Run/Stop* ein Stop erfolgt, d.h. die Routing-Steuerung läuft, so leuchtet die Anzeige *Run*.

Anzeige: Ready

Solange die Routing-Steuerung den Übertragungswunsch eines Eingabekanals (*ReadyB*) bearbeitet, leuchtet die Anzeige *Ready*.

Anzeige: Vorrang

Solange die Routing-Steuerung im Vorrangmodus arbeitet, leuchtet die Anzeige *Vorrang*.

8.1.3 Schnittstellen

Signalnamen ohne Backslash '\': aktiv high

Signalnamen mit Backslash '\': aktiv low

Ein '^' vor dem Signalnamen: die Routing-Steuerung ist die Quelle des Signals

a) Schnittstelle zur Rechneranpassung

Die Routing-Steuerung verständigt sich mit der Rechneranpassung durch einen Hand-Shake-Dialog, zum Teil über Leitungen des Privat-Busses nach rechts. Die Routing-Steuerung muss deshalb stets direkt links von der Rechneranpassung stecken. Beide Karten sollten vorzugsweise ganz rechts im Überrahmen untergebracht werden. (Siehe Abschnitt 2.2 [Überrahmen], Seite 4.).

^ReadyS\ (Data Ready)

Mit diesem Signal wird die Rechneranpassung von der Routing-Steuerung aufgefordert, ein Datenwort zu übernehmen. Es steht solange an bis die Rechneranpassung *AcceptedR* sendet.

RequestB (Data Request)

Mit diesem Signal zeigt die Rechneranpassung dem angewählten Eingabekanal an, dass das Datenwort auf den Bus zu schalten ist.

AcceptedB (Data Accepted)

Mit diesem Signal quittiert die Rechneranpassung den Empfang des Datenwortes an den Eingabekanal. Durch einen Timer auf der Rechneranpassung wird dafür gesorgt, dass *AcceptedB* mindestens 1us lang ist. Diese Verlängerung (Silena-Änderung) wurde notwendig, da z.B. der Silena-ADC eine (nicht spezifizierte) Zeit von ca. 0.5us für ein Reset benötigt.

AcceptedR (Data Accepted)

Mit diesem Signal von der Rechneranpassung wird der Routing-Steuerung mitgeteilt, dass die Übertragung beendet ist. Diese nimmt *ReadyS* zurück und startet den aktuellen Scanner neu.

StopR

Stop der Routing-Steuerung durch den Rechner.

^SADR(0-3)\ (Scanner-Adresse)

Die Scanner-Adresse wird von der Rechneranpassung dem Datenwort als Quellenkennung beigefügt (Siehe [Rechneranpassung], Seite).

Privat-Bus-Belegung: (Tab. 3.2.4.1)

StopR: 25c

ReadyS: 27c

b) Schnittstelle zu den Eingabekanälen

\sim 10 MHz-Clock,-Clock

Quarzstabilisierter 10MHz Mastertakt der Routing-Steuerung.

Tastverhältnis: 1:2

Stabilität: 10ppm

\sim 1 MHz-Clock

vom Mastertakt abgeleiteter 1MHz-Takt.

Tastverhältnis: 1:2

\sim SADR(0-3) (Scanner-Adresse)

Scanner-Adresse (4 Bits) zur Abfrage der Eingabekanäle.

ReadyB

Der abgefragte Eingabekanal meldet sich bei der Routing-Steuerung mit *ReadyB*, falls er ein Datenwort übertragen will.

\sim StopB

Dieses Signal teilt den Eingabekanälen mit, dass die Routing-Steuerung gestoppt wurde (manuell oder durch Rechner).

\sim TN, \sim TV

Freigabetakte für den Adressvergleich auf den Interface-Steuerungen für Normal- (*TN*) bzw. Vorrangmodus (*TV*).

VANM

Anmeldung eines vorrangigen Bearbeitungswunsches bei der Routing-Steuerung durch die Eingabekanäle.

8.1.4 Routing-Bus

Der Routing-Bus (Tab. 3.2.4.1) besteht aus zwei Teilen, dem 'Allgemeinen Bus', der über die ganze Breite des Überraumens geht und dem 'Privat-Bus', der erst durch das Stecken benachbarter Karten entsteht und auf diese begrenzt ist.

Allgemeiner Bus (A-Bus)

Der allgemeine Bus belegt die Anschlüsse 1a, 1c usw. bis 21c (Tab. 3.2.4.1). Die Anschlüsse mit dem Suffix a und c führen im Gegensatz zum Privat-Bus zu verschiedenen Bus-Leitungen und dürfen nicht gebrückt werden.

Stromversorgung (Standard):

+ 5V / 10A
+15V / 1A
-15V / 1A

DO - D15 (Datenleitungen):

Sechzehn Datenleitungen stehen zur Verfügung. Low auf einer Leitung bedeutet, dass das Daten-Bit gesetzt ist. Offene Leitung heißt, das Daten-Bit ist nicht gesetzt.

Kopp-, LFT-Bit:

Diese beiden Bits sind Kennungs-Bits, die von den Interfaces gesetzt werden können. Die Standard-Interface-Steuerung (IFS-Karte) nutzt sie zur Anzeige von gekoppelten Daten bzw. bei der Lifetime-Messung.

10 MHz-Clock/-Clock\:

Quarzystabilisierter Mastertakt der Routing-Steuerung.
Tastverhältnis: 1:2
Stabilität: 10ppm

1 MHz-Clock:

Vom Mastertakt abgeleiteter 1MHz Takt.
Tastverhältnis: 1:2

PADR(1-3) (Steckplatzadresse):

Für jeweils zwei benachbarte Steckplätze ist eine gemeinsame Steckplatzadresse 0-7 verdrahtet (Abb. 2.0.0.1). Die Interface-Steuerungen müssen so gesteckt werden, dass sie verschiedene Adressen einnehmen, da sie nur dann eindeutig zu adressieren sind.

SADR(0-3) (Scanner-Adresse):

Ein Steckplatz gilt als angewählt, wenn die Bits 1-3 von Steckplatz- und Scanner-Adresse übereinstimmen. Nur dann darf die Interface-Steuerung mit der Routing-Steuerung in Kontakt treten. Das Bit *SADR0* erlaubt noch das Ansprechen zweier unterschiedlicher Datenregister eines Eingabekanals. Die Scanner-Adresse wird den übertragenen Daten von der Rechneranpassung als Quellenkennung beigefügt, um sie unterscheidbar zu machen (Siehe \langle undefined \rangle [Rechneranpassung], Seite \langle undefined \rangle).

TN, TV (Freigabetakte für den Adressvergleich):

TN, *TV* erscheinen verzögert zu der Scanner-Adresse, so dass sichergestellt ist, dass die Adress-Bits eingeschwungen sind und keine Vergleichsfehler auftreten können. Ferner

ermöglichen *TN, TV* eine Unterscheidung welcher der beiden Scanner unterschiedlicher Priorität gerade aktiv ist.

TN: Normal-Scanner

TV: Vorrang-Scanner

VANM (Vorrang Anmeldung):

Über diese Leitung kann bei der Routing-Steuerung ein vorrangiger Bearbeitungswunsch angemeldet werden. Die Routing-Steuerung fragt in einer bestimmten Phase jedes Scan-Zyklus (Abb. 3.2.4.6) diese Leitung ab und schaltet auf den gewünschten Scanner.

ReadyB (Data ready); **RequestB** (Data request); **AcceptedB** (Data accepted)

Mit diesen drei Signalen synchronisiert sich die Datenübertragung eines vom Scanner angewählten Eingabekanals. Die Interface-Steuerung meldet *ReadyB*, falls sie Daten übertragen will. Dies hat ein Stoppen des Scanners zur Folge. Mit *RequestB* wird die Datenquelle aufgefordert die Daten auf den Bus zu geben. Beendet wird die Übertragung mit *AcceptedB*. Mit diesem Signal wird im allgemeinen auch die Datenquelle geleert, die dann das nächste Datenwort aufnehmen kann.

StopB:

Dieses Signal teilt allen Eingabekanälen gleichzeitig mit, dass die Routing-Steuerung gestoppt wurde.

Privat-Bus (P-Bus) Der Privat-Bus baut sich erst durch Stecken der Karten auf. Eine fehlende Karte oder das Fehlen entsprechender Brücken auf den Karten unterbrechen den Privat-Bus. Er reicht von den Leitungen 22 bis 32 einschließlich. Die Leitungen mit dem Suffix a führen zur linken Nachbarkarte und die mit c zur rechten. Soll ein Bus aufgebaut werden, so sind die Anschlüsse a und c einer Leitung zu brücken. Die individuelle Nutzung des Privat-Busses ist den einzelnen Kartenbeschreibungen zu entnehmen (z.B. RST-, IFS-, KPL- Karten).

Tab. 8.1.4.1 Data-Routing-Bus

	a	Pin	c	
Spannungs-	(digital) +5V	1	+5V (digital)	Spannungs-
Versorgung	(digital) 0V	2	0V (digital)	Versorgung
		D0\ 3	D1\	
		D2\ 4	D3\	
		D4\ 5	D5\	
		D6\ 6	D7\	
		D8\ 7	D9\	
		D10\ 8	D11\	
		D12\ 9	D13\	
		D14\ 10	D15\	
A-Bus	Kopp-Bit\	11	LFT-Bit\	A-Bus
	SADRO\	12	1MHz-Clock	
	SADR1\	13	PADR1\	
	SADR2\	14	PADR2\	
	SADR3\	15	PADR3\	
	RequestB	16	ReadyB\	
	StopB\	17	AcceptedB	
	10MHz-Clock\	18	10MHz-Clock	
	TV	19	TN	
	(analog) 0V	20	VANM\	
	(analog) +15V	21	-15V (analog)	
		22		
		23		
		24		
		25		
P-Bus		26		P-Bus
		27		
		28		
		29		
		30		
		31		
		32		

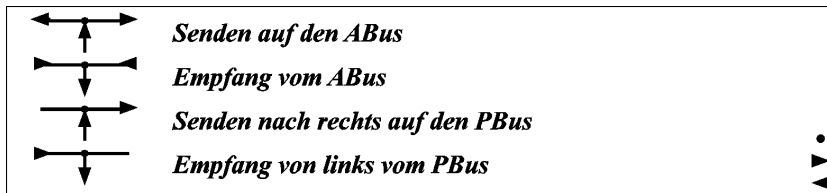


Abb. 8.1.4.2 Symbole in den Blockdiagrammen (Abb. 8.1.4.3 - 8.1.4.5)

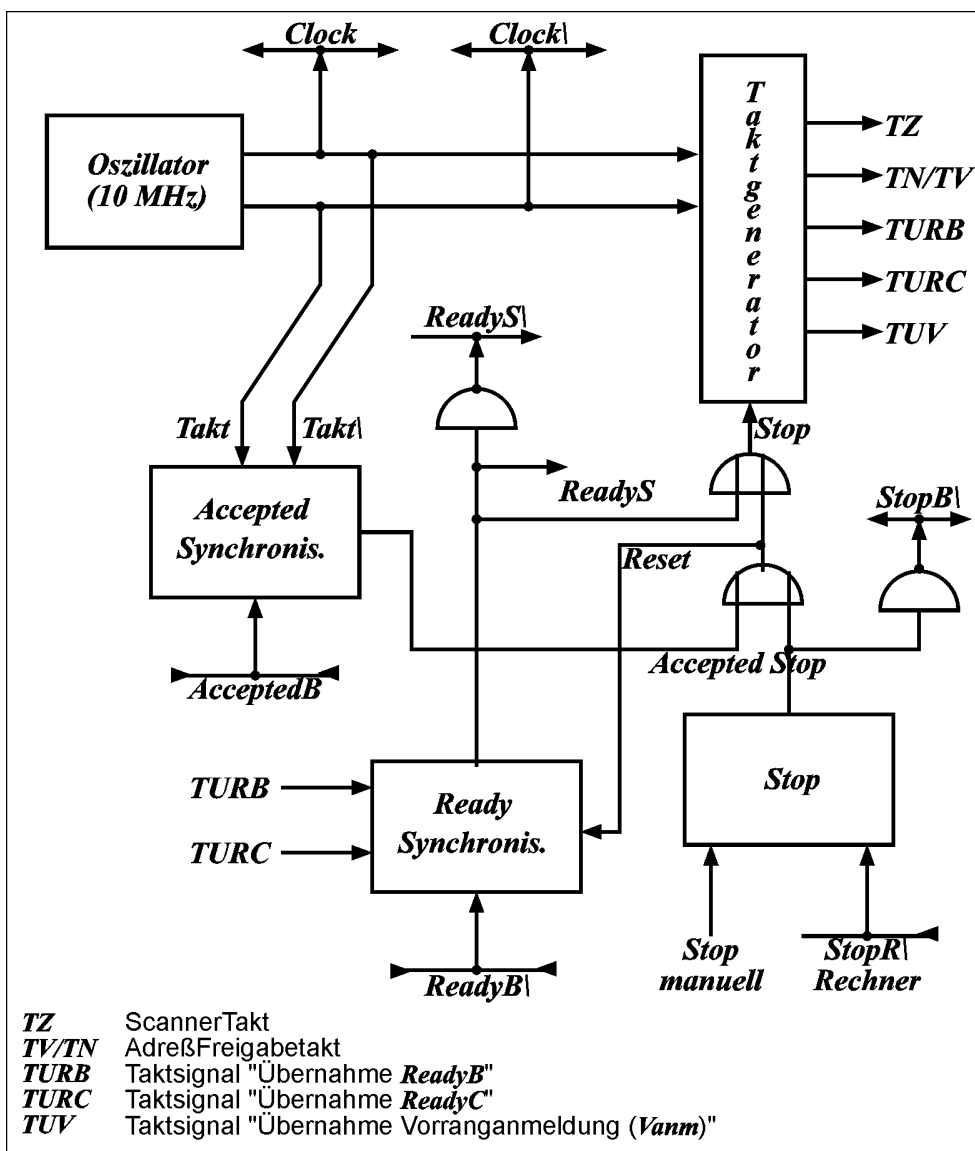


Abb. 8.1.4.3 Routing-Steuerung Taktgenerator

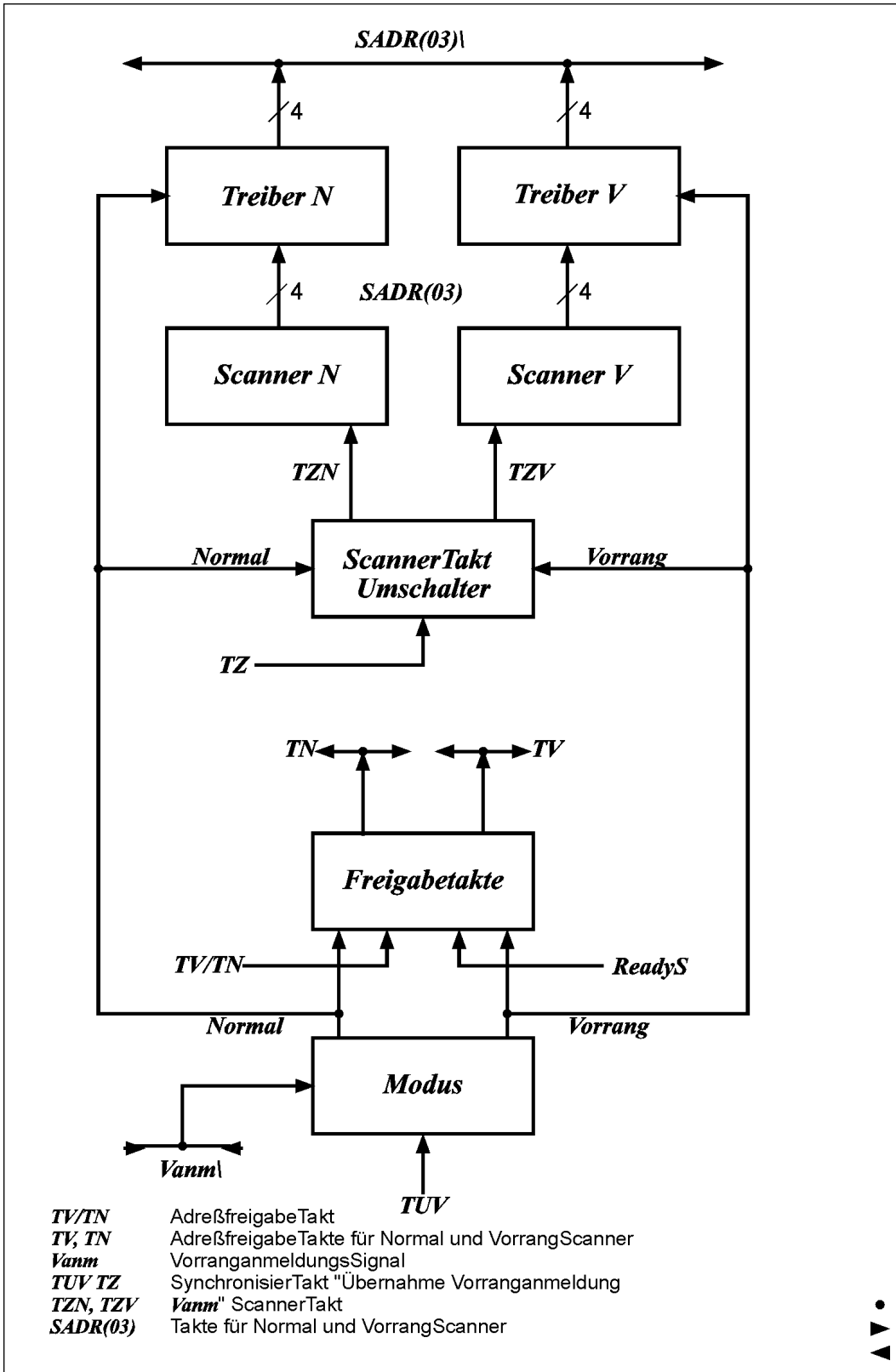


Abb. 8.1.4.4 Routing-Steuerung Scanner

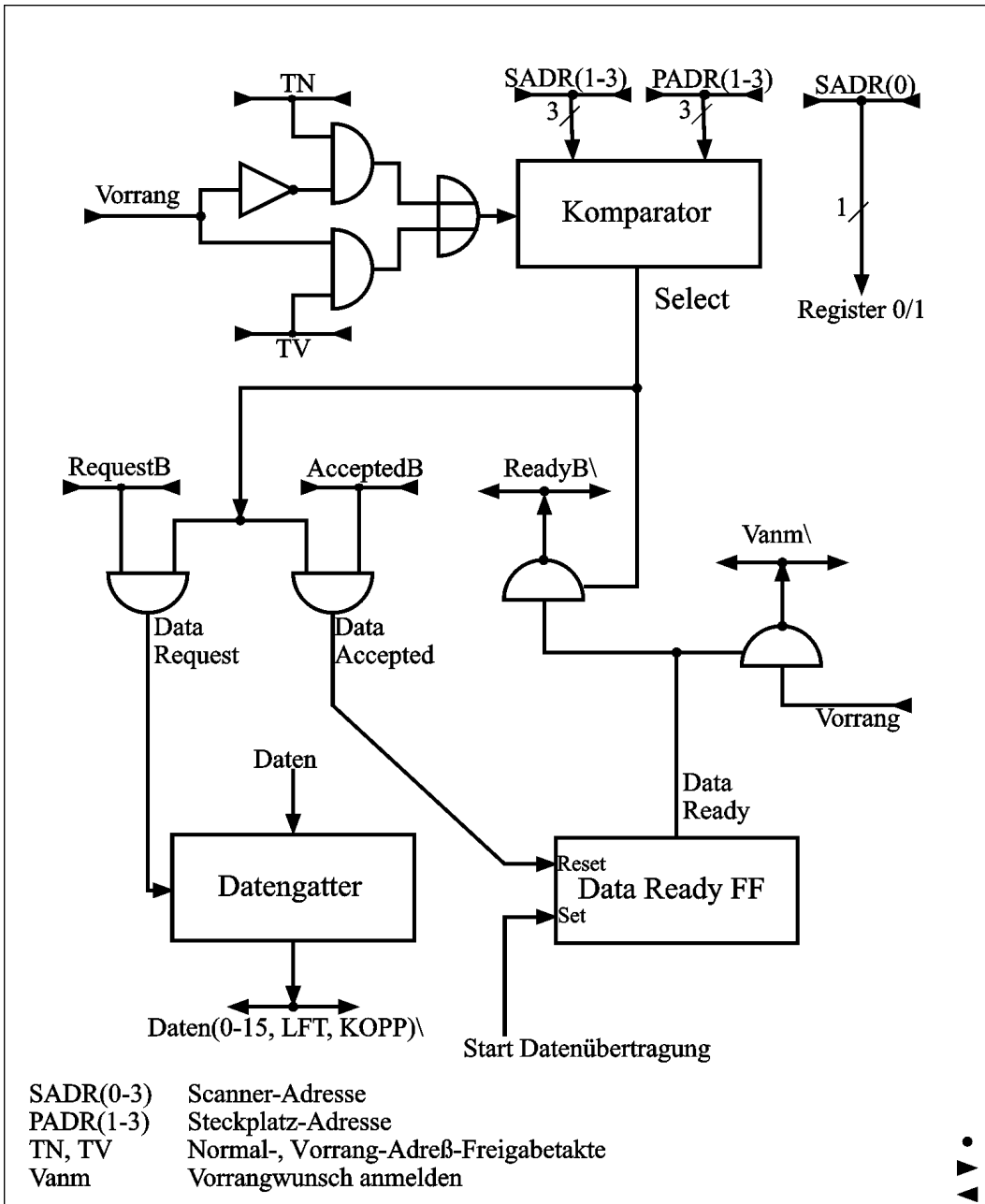


Abb. 8.1.4.5 Interface-Steuerung

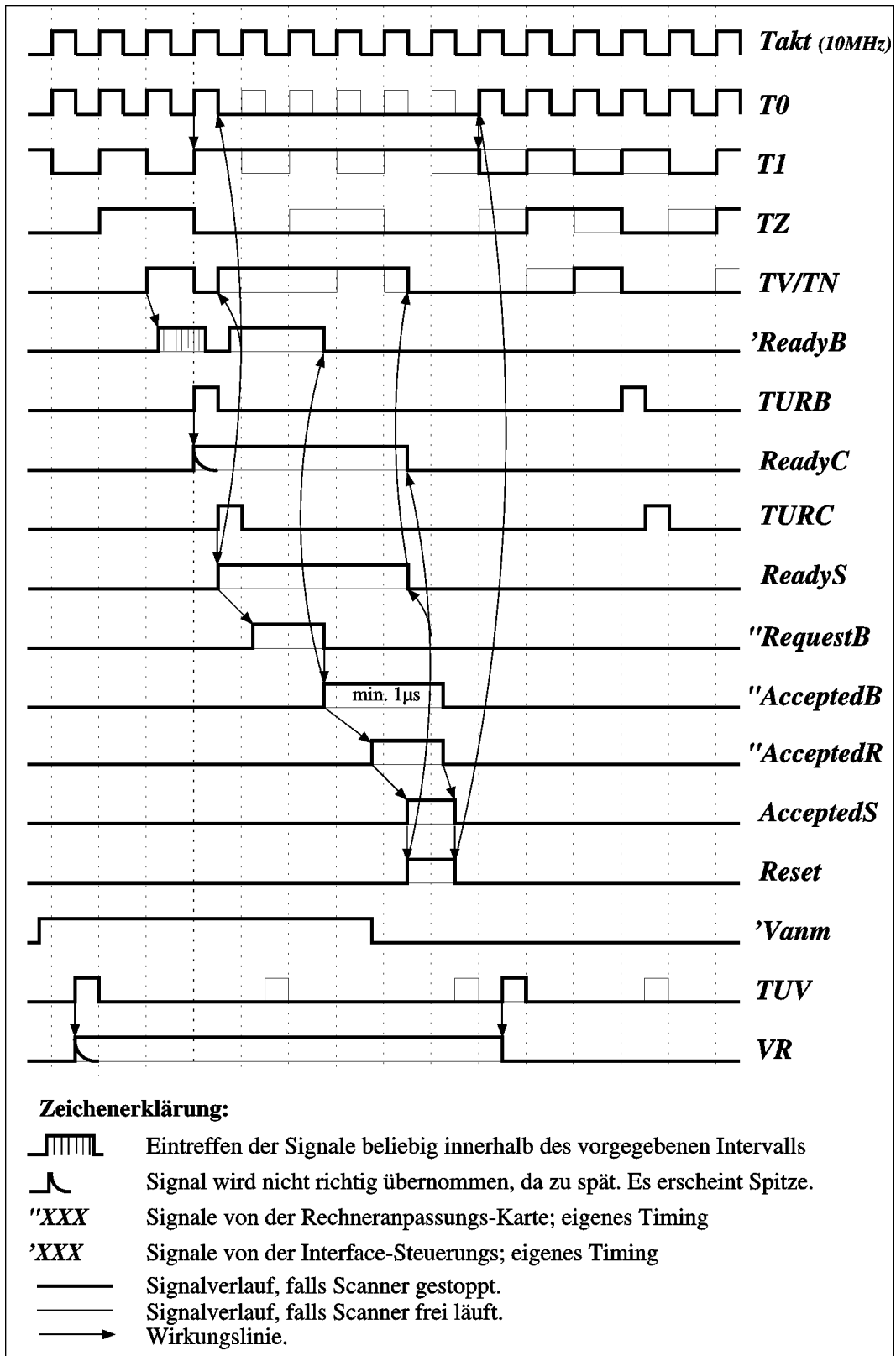


Abb. 8.1.4.6 Timing Routing-Steuerung

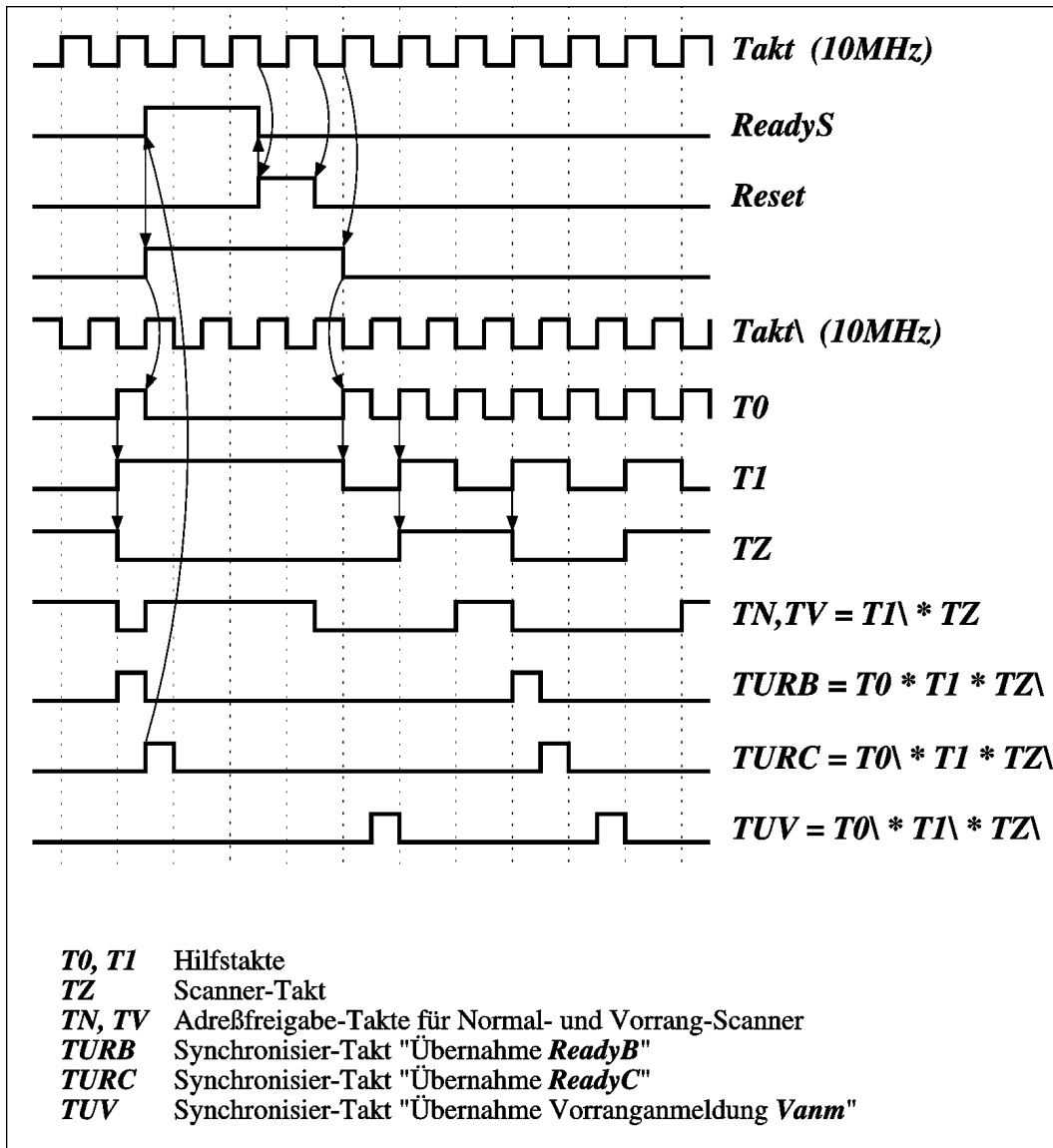


Abb. 8.1.4.7 Timing Taktgenerator

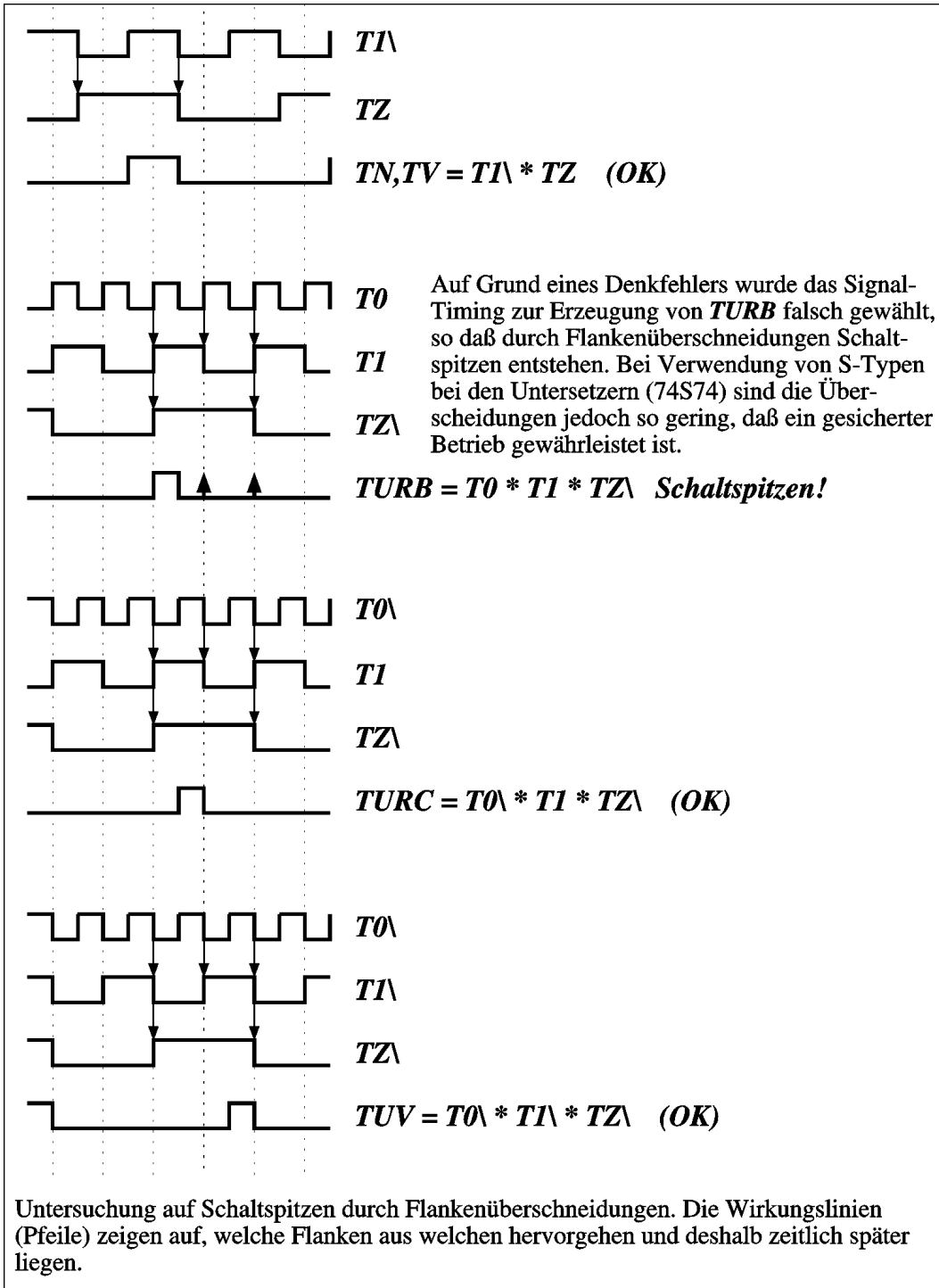


Abb. 8.1.4.8 Timing-Details Taktgenerator

8.2 Alte Rechneranpassungen

8.2.1 PDP11 - DRQ11-CA - Anpassung

Aufbau der Datenwörter

Es werden die 16 Daten-Bits der Eingabedaten, die 4 Scanner-Adress-Bits, das Lifetime-Bit und das Kopplungs-Bit auf zwei 16 Bit Eingabeworte aufgeteilt und mit je einem Wortkennungs-Bit versehen (Tab. 8.2.1.1). Die Wortkennungs-Bits erlauben der Software die Unterscheidung von erstem und zweitem Eingabewort und damit die Erkennung von Datenverlusten.

Tab. 8.2.1.1 Aufbau der Datenwörter (PDP11)

Bit	Wort 1	Wort 2		
0	D8	D0		
1	D9	D1		
2	D10	D2		
3	D11	D3		
4	D12	D4		
5	D13	D5		
6	D14	D6		
7	D15	D7		
8	H	SADR1		
9	H	SADR2	D0...15	Bits der Eingabedaten
10	H	SDAR3	SADRO...3	Scanner-Adress-Bits
11	H	H	LFT	Lifetime-Bit
12	H	SADRO	KOPP	Kopplungs-Bit
13	H	LFT	H	High-Pegel = "0"
14	H	KOPP	L	Low-Pegel = "1"
15	L	H	Bit 15	Wortkennungs-Bit

8.2.2 TR86 - CALAS - Anpassung

Von dieser inzwischen ersetzten Rechneranpassung existiert noch eine Reihe von Karten, deren spätere Verwendung z.B. an einem PC wegen ihrer einfachen Schnittstellen-Logik denkbar wäre. Deshalb wurde ihre ausführliche Beschreibung erhalten. Bei ihrem Einsatz muss jedoch die sogenannte Silena-Änderung auf der Routing-Steuerung berücksichtigt werden (s.h. Schaltungs-Unterlagen für PDP11-Rechneranpassung).

Zwei Varianten dieser Rechneranpassungs-Karte wurden entwickelt. Eine leitet die Datenworte unmittelbar, lediglich durch Leitungstreiber verstärkt zum Rechner weiter. Die andere schiebt die Datenworte zuerst durch einen 24 Bit breiten und 64 Worte tiefen FiFo-Speicher. Der Füllstand des FiFo-Speichers wird durch LED's angezeigt und steht über Ausgänge zur Steuerung des Experimentes zur Verfügung. Der Einsatz dieser Karte ist angebracht, wenn die Daten stoßweise auftreten, jedoch nicht mehr als 64 Datenworte auf einmal. Bei statistisch ankommenden Daten ist ihr Einsatz von geringer Wirkung, falls die nachfolgende Rechner-Elektronik durch einige Speicherregister die statistischen Spitzen bereits weitgehend ausglättet.

Funktion

Die Rechneranpassungs-Karte sorgt mittels 74LS245 für das Senden und Empfangen von Daten und Steuersignalen über das Kabel zwischen Routing und Rechner. Die Schnittstelle zum Rechner besteht aus 21 Datenleitungen und 4 Steuerleitungen, über die ein einfacher 'Hand-Shake-Dialog' abläuft (Abb. 8.2.2.1).

Mit dem Signal *Data Ready* meldet das Routing dem Rechner, dass Daten bereit stehen zur Übergabe. Der Rechner fordert die Daten mit *Data Request* an und quittiert deren Übernahme mit *Data Accepted*, worauf das Routing das *Data Ready* zurücknimmt und nach Beendigung von *Data Accepted* zum nächsten Eingabekanal weitergeht.

Zu diesen drei Hand-Shake-Signalen kommt noch ein *Stop*-Signal hinzu, mit dem der Rechner mitteilt, dass die Datenübertragung gestoppt ist.

Kritisch ist der Augenblick, in dem die Daten-Bits durchgeschaltet werden. Dies führt durch Übersprechen auf die Steuerleitungen durchweg zu erheblichen Störungsspitzen. Zwar sind die Polaritäten der Steuersignale so gewählt, dass diese Störungen keinen Einfluss auf die Funktion haben sollten, in der Praxis war dies jedoch nicht ausreichend. Folgende Effekte konnten beobachtet werden:

- Die Kabelleitungen (Länge ca. 150 cm) neigen zum Schwingen, wodurch nach dem Schalten der Daten-Bits Störungen beider Polaritäten auftreten. Insbesondere kann dadurch kurzzeitig *Data Accepted* ausgelöst werden, was ein verfrühtes Clearen von Routing-Steuerung und Datenquelle zur Folge hat.
- Es gibt Datenquellen, die ihre Daten noch kurz nach *Data Request* verändern, was zu störenden Low-high Übergängen auf den Datenleitungen führen kann. Dies führt zu den gleichen Folgen wie zuvor.
- Am Ende von *Data Request* können die nach high gehenden Datenleitungen auf der *Request*-Leitung eine solche Störung hervorrufen, dass die Daten wieder eingeschaltet werden. Diese Rückkopplung kann sich unendlich fortsetzen.

Zur Unterdrückung dieser Effekte wurden *Data Accepted* und *Data Request* gegeneinander verriegelt. Jedes dieser Steuersignale kann nur durchkommen, wenn das andere nicht ansteht (nicht bei FiFo-Version und älteren Rechneranpassungs-Karten). Ferner wurden alle

Steuerleitungen mit pull-up bzw. pull-down Widerständen und Kondensatoren belastet, um Störungen zu vermindern.

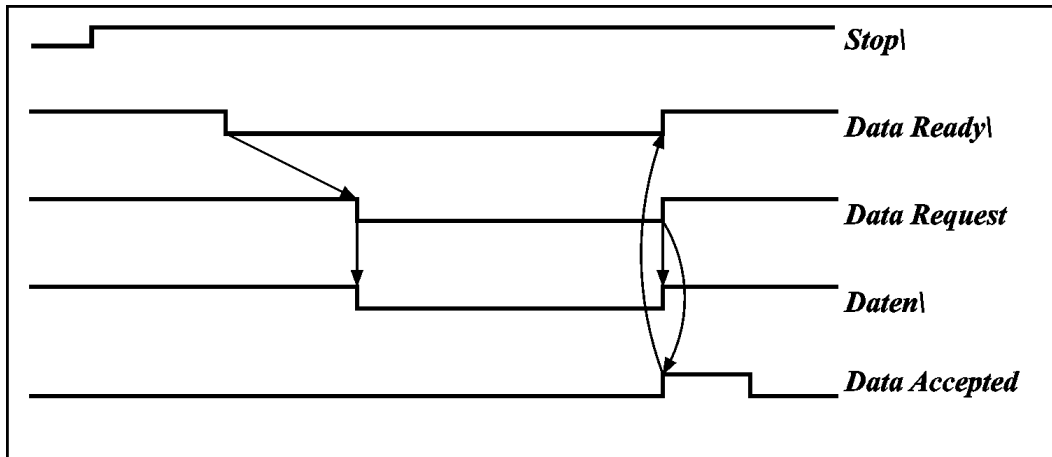


Abb. 8.2.2.1 Timing zwischen Routing und Rechner (CALAS)

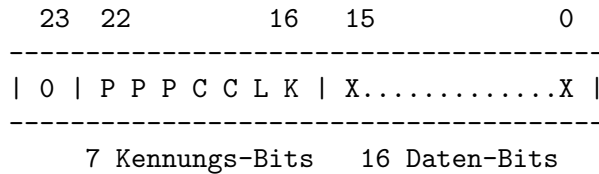
Aufbau des Datenwortes

Die Zusammenstellung des Datenwortes ist für beide Varianten der Rechneranpassungs-Karte gleich und der CALAS-Elektronik angepasst (Tab. 8.2.2.2). Die CALAS-Elektronik kennt 16 Daten-Bits (0-15), 7 Quellenkennungs-Bits (16-22) und ein Fehlerkennungs-Bit (23), mit dem das CALAS Übertragungsfehler anzeigt. Entsprechend überträgt die Rechneranpassungs-Karte 16 Daten-Bits, jedoch nur 5 Kennungs-Bits (3 Steckplatzadressen, Lifetime-Bit, Kopplungs-Bit). Mit den verbleibenden zwei Kennungs-Bits (CC-Kennung) kann die CALAS-Elektronik verschiedene direkt angeschlossene Datenquellen (z.B. Routings) unterscheiden.

Als Kennungs-Bits werden die drei höchstwertigen Bits der Scanner-Adresse (SADR(1-3)=Steckplatzadresse), sowie die den Datenquellen verfügbaren Bits Lifetime-Bit (LFT-Bit) und Kopplungs-Bit (Kopp-Bit) übertragen. Die tatsächliche Anordnung der Bits im Datenwort hängt noch von der Verdrahtung im Rechner ab. Eine Standardverdrahtung zeigt die folgende Tabelle:

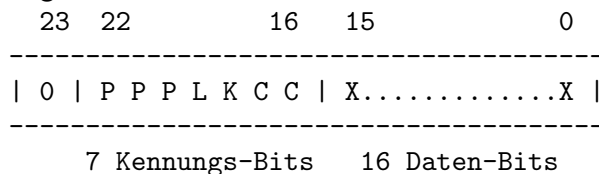
Tab. 8.2.2.2 Aufbau eines Datenwortes (CALAS)

Verdrahtungsvariante 1:



- CC: 00: direkt angeschlossene Datenquellen (z.B. Testkarte '02'H)
- 01:
- 10: Routing oder direkte Datenquellen
- 11:

Verdrahtungsvariante 2:



- CC: 10: direkt angeschlossene Datenquellen (z.B. Testkarte '02'H)
- 00:
- 01: Routing oder direkte Datenquellen
- 11:

PPP: Steckplatz der Datenquelle
 L: Lifetime-Bit
 K: Kopplungs-Bit

Variante mit FiFo-Speicher

Die Daten werden vom Routing entgegengenommen, durch einen 64 Worte tiefen und 24 Bit breiten FiFo-Speicher geschoben und anschliessend an den Rechner übergeben. Die Ein- und Ausgaben beim FiFo-Speicher laufen asynchron zueinander. Mit jeder Eingabe (*Ready*) und jeder Ausgabe (*Accepted*) wird ein Zähler hoch- bzw. runtergezählt zur Registrierung des Füllstandes des FiFo. Aus den drei Bits der Wertigkeit 16, 32, und 64 werden Signale abgeleitet, die den Füllstand anzeigen. Mit dem *Stop*-Signal (manuell oder Rechner) werden FiFo und Zähler gelöscht, um einen definierten Anfangszustand zu schaffen.

Anzeigen: 1/4 - 1/1 (LED's)

Über vier LED's wird der Füllstand des FiFo (1/4, 1/2, 3/4, 1/1) angezeigt. Eine LED, die leuchtet, zeigt an, dass der FiFo mindestens bis zu dem zugehörigen Wert gefüllt ist.

Anzeige	belegt
1/4	16-31
1/2	32-47
3/4	48-63
1/1	64

BNC-Buchsen: 0, 1/4, 1/2, 3/4 (TTL-Ausgänge) Zur Steuerung des Experimentes können diesen Ausgängen Signale entnommen werden, die anzeigen wieviel freie Plätze im FiFo-Speicher mindestens noch frei sind.

Ausgang high	min. frei
0	0
1/4	16
1/2	32
3/4	48

Simulatorkarte

Die Simulatorkarte ermöglicht das Testen einer Routing-Station ohne deren Anschluss an einen Rechner. Sie wickelt mit dem Routing den notwendigen Hand-Shake-Dialog ab, wertet die übergebenen Daten jedoch nicht aus. Die Breite der Hand-Shake-Signale *Data Request* und *Data Accepted* kann über Potentiometer auf der Karte eingestellt werden. (Die PDP11-DRQ11 Anpassung führt diesen Test selbständig aus, wenn die Verbindung zum Rechner offen ist.)

Die Simulatorkarte kann in zwei Modi betrieben werden, die auf der Karte durch Schalter einzustellen sind:

- Modus: über Routing-Bus
In diesem Modus übernimmt die Simulatorkarte die Funktion der Rechneranpassungs-Karte und des Rechners. Sie muss deshalb wie die Rechneranpassungs-Karte direkt rechts neben der Routing-Steuerung plziert werden, mit der sie sich über den Routing-Bus verständigt. Der frontseitige Kabelstecker bleibt unbenutzt.
- Modus: über Kabelstecker
In diesem Fall übernimmt die Simulatorkarte die Funktionen des Rechners. Vom Routing-Bus benötigt sie lediglich die Spannungsversorgung während Daten und Steuerungssignale von der Rechneranpassungs-Karte über den frontseitigen Kabelstecker zugeführt werden.