SIEMENS

ICs for Communications

ISDN Subscriber Access Controller for Terminals

ISAC®-S TE

PSB 2186

User's Manual 10.94

PEB 2186 Revision History: 10.94	
Previous Releases: **11.88; 3.89; 12.89; 02.95**	
Page	Subjects (changes since last revision)
	The present documentation is an editorial update of the Technical Manual 12.89

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at T_A = 25 °C and the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description are fulfilled.

For detailed technical information about **“Processing Guidelines”** and **“Quality Assurance”** for ICs, see our **“Product Overview”**.

Edition 10.94

This edition was realized using the software system FrameMaker®

Published by Siemens AG, Breech Belittler, Marketing-Communication, Banisters 73, D-81541 Munching.

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Introduction

The PSB 2186 ISAC®-S TE implements the four-wire S/T interface used to link voice/data terminals to an ISDN.

The PSB 2186 combines the functions of the S-Bus Interface Circuit (SBC: PEB 2080) and the ISDN Communications Controller (ICC: PEB 2070) on one chip.

The component switches B- and D-channels between the S/T and the ISDN Oriented Modular (IOM®) interfaces, the latter being a standard backplane interface for the ISDN-basic access.

The device provides all electrical and logical functions of the S/T interface, such as: activation/ deactivation, mode dependent timing recovery and D-channel access and priority control.

The HDLC packets of the ISDN D-channel are handled by the ISAC-S which interfaces them to the associated microcontroller. In one of its operating modes the device offers high level support of layer-2 functions of the LAPD protocol.

The ISAC-S is a CMOS device, available in a P-DIP-40, P-LCC-44, P-MQFP-64 package. It operates from a single + 5 V supply and features a power-down state with very low power consumption.

SIEMENS

ISDN Subscriber Access Controller for Terminals (ISAC®-S TE)

PSB 2186

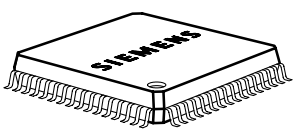
Preliminary Data

CMOS IC

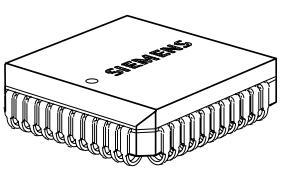
1 Features

Terminal IOM®-2 terminal specific version of the PEB 2086:

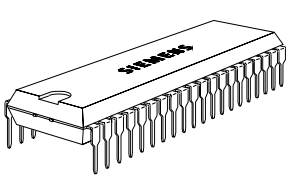
- Pin and software compatible to PEB 2086
- Compatible to PEB 2085 (Symmetrical Receiver)
- Full duplex 2B+D S/T interface transceiver according to CCITT I.430
- Conversion of the frame structure between the S/T interface and IOM-2
- Receive timing recovery
- D-channel access control
- Activation and deactivation procedures with automatic wake-up from power-down state
- Access to S and Q bits of S/T interface
- Adaptively switched receive thresholds
- Support of LAPD protocol
- FIFO buffer (2 x 64 bytes) for efficient transfer of D-channel packets
- 8-bit microprocessor interface, multiplexed or non-multiplexed
- Serial interface: IOM-2 interface including bit clock and strobe signal
- Implementation of IOM-2 MONITOR and C/I-channel protocol to control peripheral devices
- Microprocessor access to B- and intercommunication-channels
- Watchdog timer
- Advanced CMOS technology
- Low power consumption: standby: 8 mW; active: 80 mW



P-MQFP-64



P-LCC-44



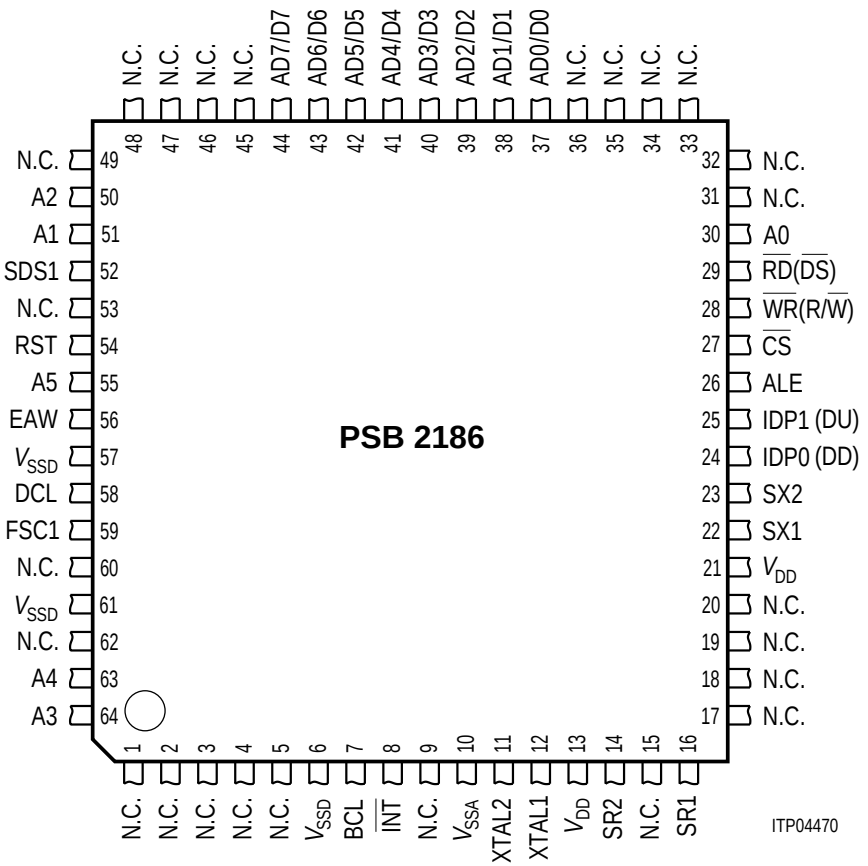
P-DIP-40

Type	Ordering Code	Package
PSB 2186H	Q67100-H6412	P-MQFP-64 (SMD)
PSB 2186N	Q67100-H6390	P-LCC-44 (SMD)
PSB 2186P	Q67100-H6389	P-DIP-40

The PSB 2186, ISAC-S TE is software compatible to the PEB 2085, ISAC-S.

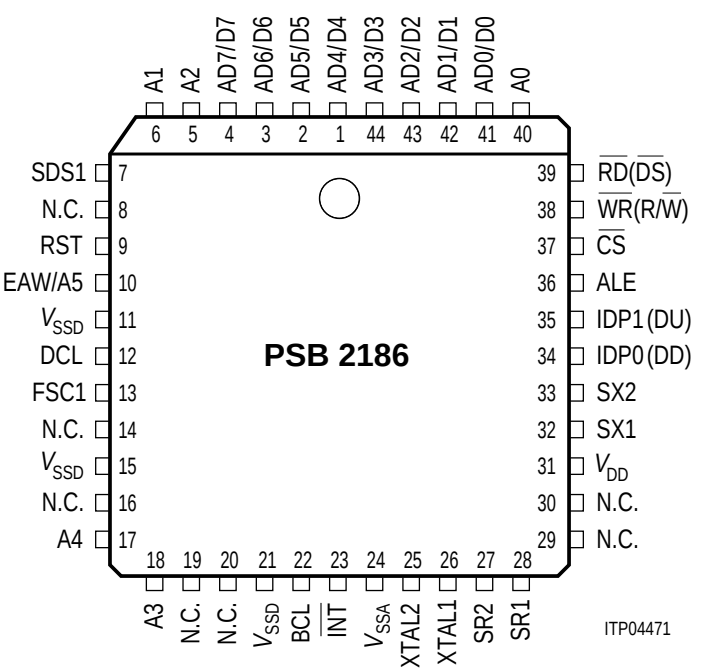
Pin Configuration
(top view)

P-MQFP-64



ITP04470

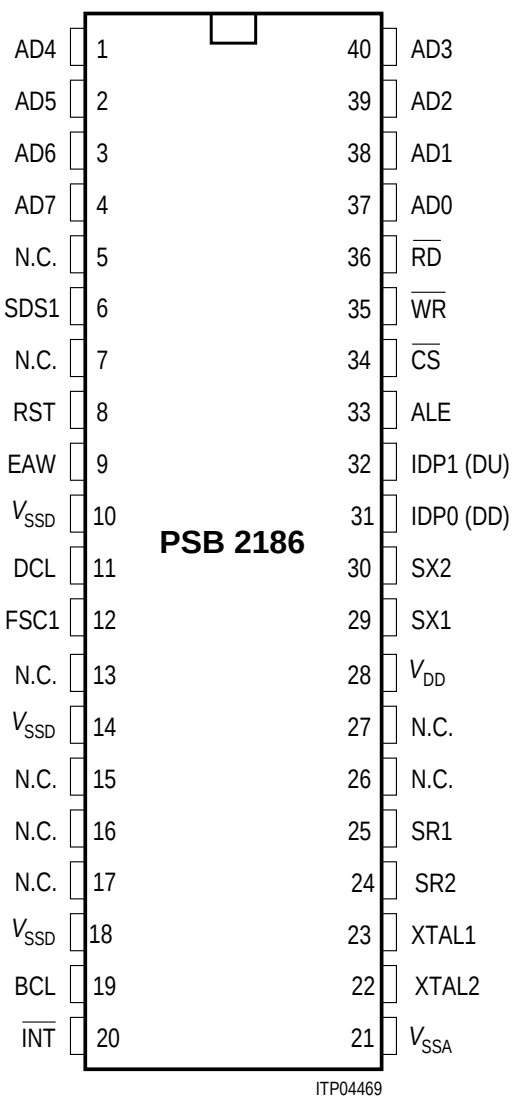
P-LCC-44



ITP04471

Pin Configuration
(top view)

P-DIP-40



1.1 Pin Definitions and Functions

Pin No. P-DIP-40	Pin No. P-MQFP-64	Pin No. P-LCC-44	Symbol	Input (I) Output (O) Open Drain (OD)	Function
37 38 39 40 1 2 3 4	37 38 39 40 41 42 43 44	41 42 43 44 1 2 3 4	AD0/D0 AD1/D1 AD2/D2 AD3/D3 AD4/D4 AD5/D5 AD6/D6 AD7/D7	I/O I/O I/O I/O I/O I/O I/O I/O	**Multiplexed Bus Mode**: Address/data bus transfers addresses from the µP system to the ISAC-S TE and data between the µP system and the ISAC-S TE. **Non-Multiplexed Bus Mode**: Data bus. Transfers data between the µP system and the ISAC-S TE.
34	27	37	$\overline{CS}$	I	**Chip Select**: A “Low“ on this line selects the ISAC-S TE for a read/write operation.
–	28	38	R/$\overline{W}$	I	**Read/Write**: When “High” identifies a valid µP access as a read operation. When “Low”, identifies a valid µP access as a write operation (Motorola bus mode).
35	28	38	$\overline{WR}$	I	**Write:** This signal indicates a write operation (Intel bus mode).
–	29	39	$\overline{DS}$	I	**Data Strobe**: The rising edge marks the end of a valid read or write operation (Motorola bus mode).
36	29	39	$\overline{RD}$	I	**Read**: This signal indicates a read operation (Intel bus mode).
20	8	23	$\overline{INT}$	OD	**Interrupt Request**: The signal is activated when the ISAC-S TE requests an interrupt. It is an open drain output.

1.1 Pin Definitions and Functions (cont'd)

Pin No. P-DIP-40	Pin No. P-MQFP-64	Pin No. P-LCC-44	Symbol	Input (I) Output (O) Open Drain (OD)	Function
33	26	36	ALE	I	**Address Latch Enable**: A high on this line indicates an address on the address/data bus (multiplexed bus type only). ALE also selects the microprocessor interface type (multiplexed or non-multiplexed) P-LCC and P-MQFP only.
8	54	9	RST	I/O	**Reset**: A “High“ on this input forces the ISAC-S TE into reset state. The minimum pulse length is four DCL-clock periods or four ms. If the terminal specific functions are enabled, the ISAC-S TE may also supply a reset signal.
12	59	13	FSC1	O (I)	**Frame Sync 1**: Frame sync output. “High“ during channel 0 on the IOM-2 interface. FSC1 becomes Input if Test Mode is programmed (ADF1).
11	58	12	DCL	O (I)	**Data Clock**: Clock of frequency equal to twice the data rate on the IOM-interface Clock output 1536-kHz IOM-2 mode DCL becomes Input if Test Mode is programmed (ADF1).
– – – – – –	30 51 50 64 63 55	40 6 5 18 17 10	A0 A1 A2 A3 A4 A5	I I I I I I	Address Bit 0 Address Bit 1 Address Bit 2(Non-multiplexed Address Bit 3bus mode) Address Bit 4 Address Bit 5

1.1 Pin Definitions and Functions (cont'd)

Pin No. P-DIP-40	Pin No. P-MQFP-64	Pin No. P-LCC-44	Symbol	Input (I) Output (O) Open Drain (OD)	Function
9	56	10	EAW	I	**External Awake** (terminal specific function). If a falling edge on this input is detected, the ISAC-S TE generates an interrupt and, if enabled, a reset pulse.
6	52	7	SDS1	O	**Serial Data Strobe 1**. A programmable strobe signal, selecting either one or two B- or IC-channels on IOM-2 interface, is supplied via this line. After reset, SDS1 takes on its function only after a write access to SPCR is made.
19	7	22	BCL	O	**Bit Clock**: Clock of frequency 768 kHz, IOM-2 mode.
10, 14, 18	57, 6, 61	11, 15, 21	V_{SSD}	–	Digital ground
21	10	24	V_{SSA}	–	Analog ground
28	13, 21	31	V_{DD}	–	Power supply (5 V ± 5 %)
23	12	26	XTAL1	I	Connection for crystal or external clock input
22	11	25	XTAL2	O	Connection for external crystal. Left unconnected if external clock is used.
24	14	27	SR2	I	**S-Bus Receiver Input**
25	16	28	SR1	I	**S-Bus Receiver Input**
29	22	32	SX1	O	**S-Bus Transmitter Output** (positive)
30	23	33	SX2	O	**S-Bus Transmitter Output** (negative)
31	24	34	IDP0(DD)	I/O	**IOM-Data Port 0 (DD)**
32	25	35	IDP1(DU)	I/O	**IOM-Data Port 1 (DU)** Open drain without internal pull-up resistor or push-pull (ADF2:ODS)

1.2 Logic Symbol

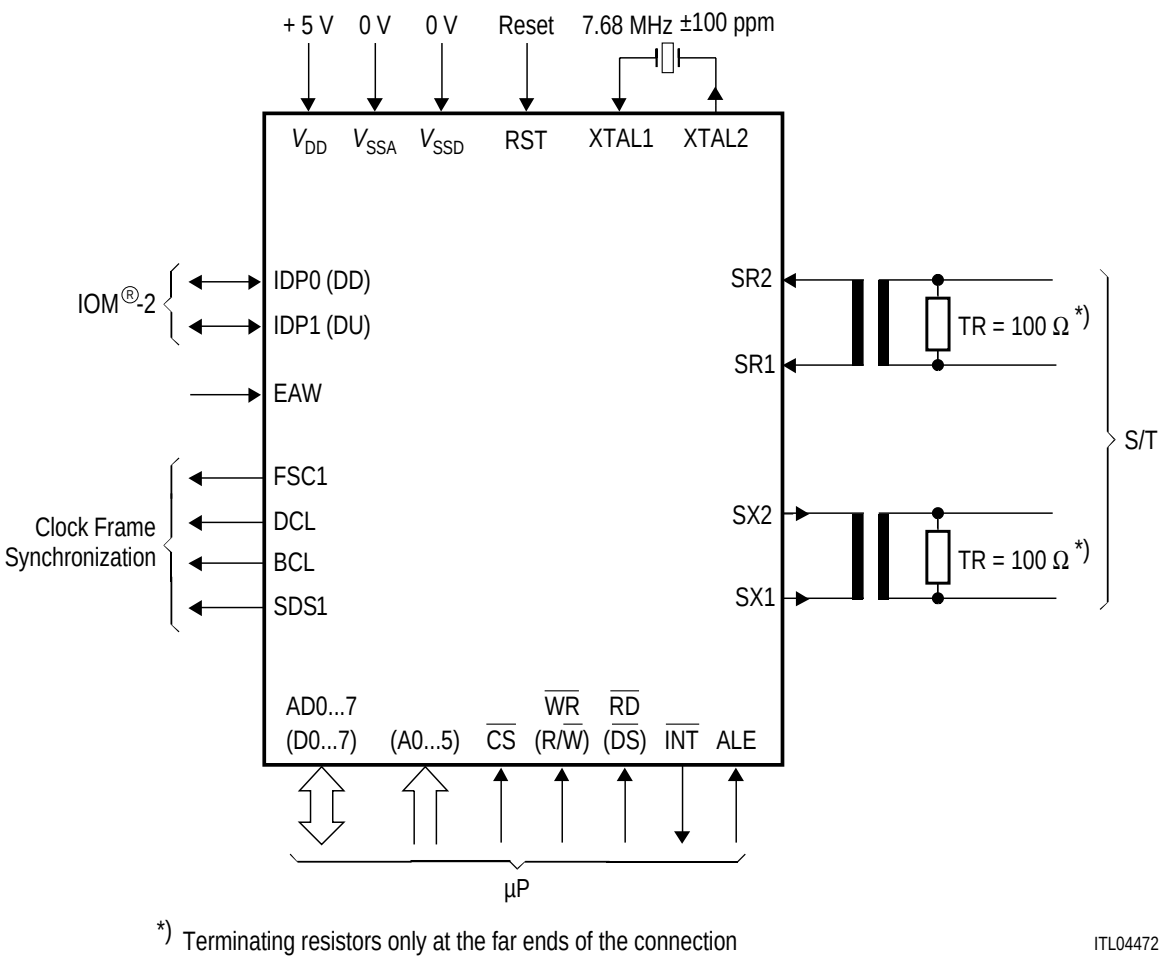


Figure 1
Logic Symbol of the ISAC®-S TE PSB 2186

1.3 Functional Block Diagram

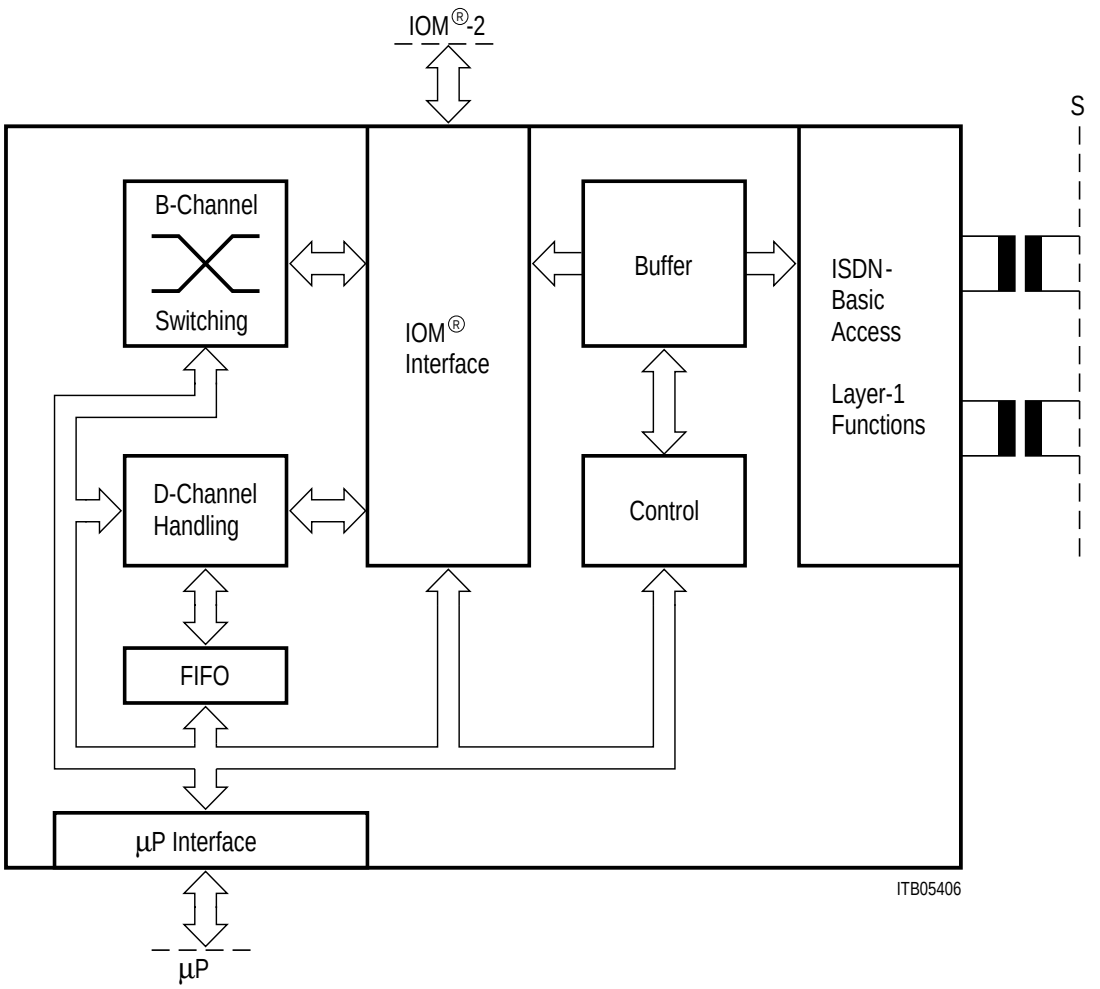


Figure 2
Block Diagram of the ISAC®-S TE

1.4 System Integration

1.4.1 ISDN Applications

The reference model for the ISDN-basic access according to CCITT I series recommendations consists of

– an exchange and trunk line termination in the central office (ET, LT)

– a remote network termination in the user area (NT)

– a two-wire loop (U interface) between NT and LT

– a four-wire link (S interface) which connects subscriber terminals and the NT in the user area as depicted in **figure 3.**

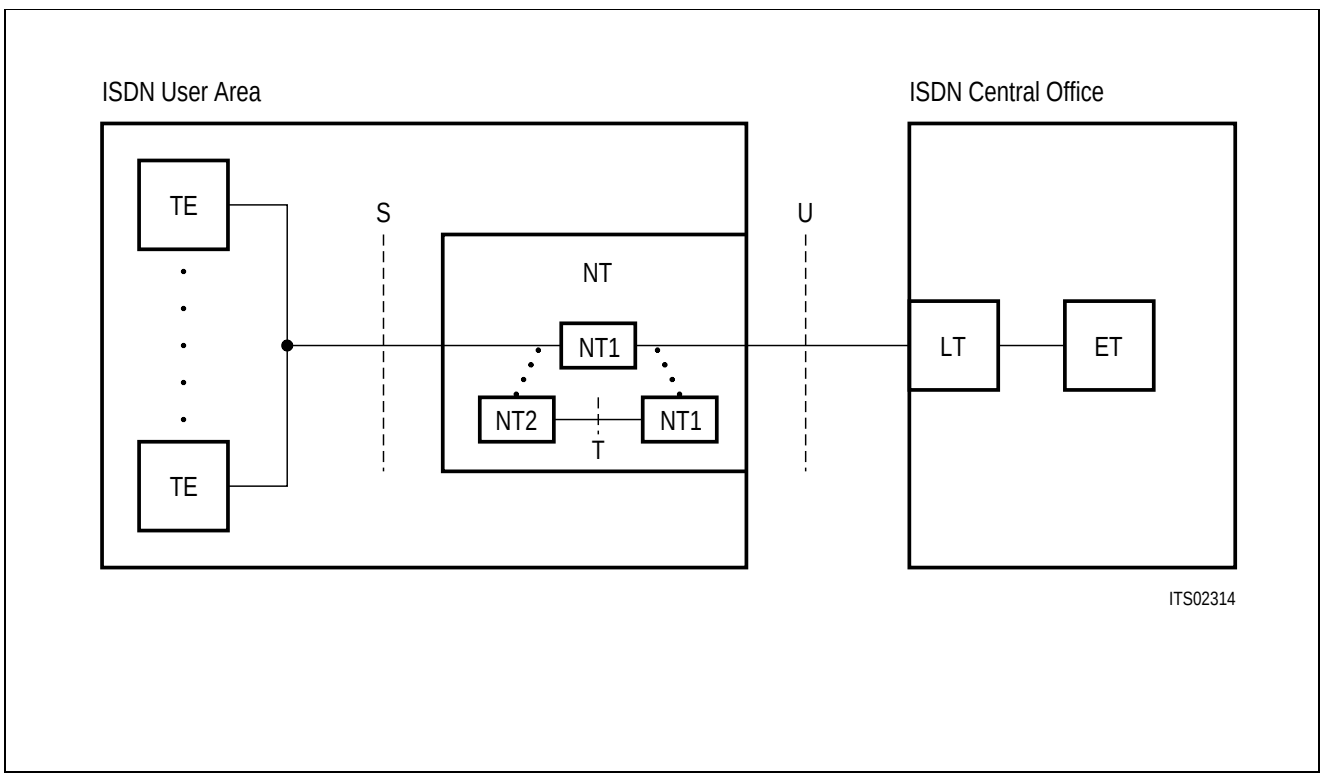


Figure 3
ISDN-Basic Subscriber Access Architecture

The NT equipment serves as a converter between the U interface at the exchange and the S interface at the user premises. The NT may consist of either an NT1 only or an NT1 together with an NT2 connected via the T interface which is physically identical to the S interface. The NT1 is a direct transformation between layer 1 of S and layer 1 of U. NT2 may include higher level functions like multiplexing and switching as in a PBX.

The ISAC-S TE is designed for the user area of the ISDN-basic access, especially for subscriber terminal equipment with S interfaces. **Figure 4** illustrates the application of the ISAC-S TE.

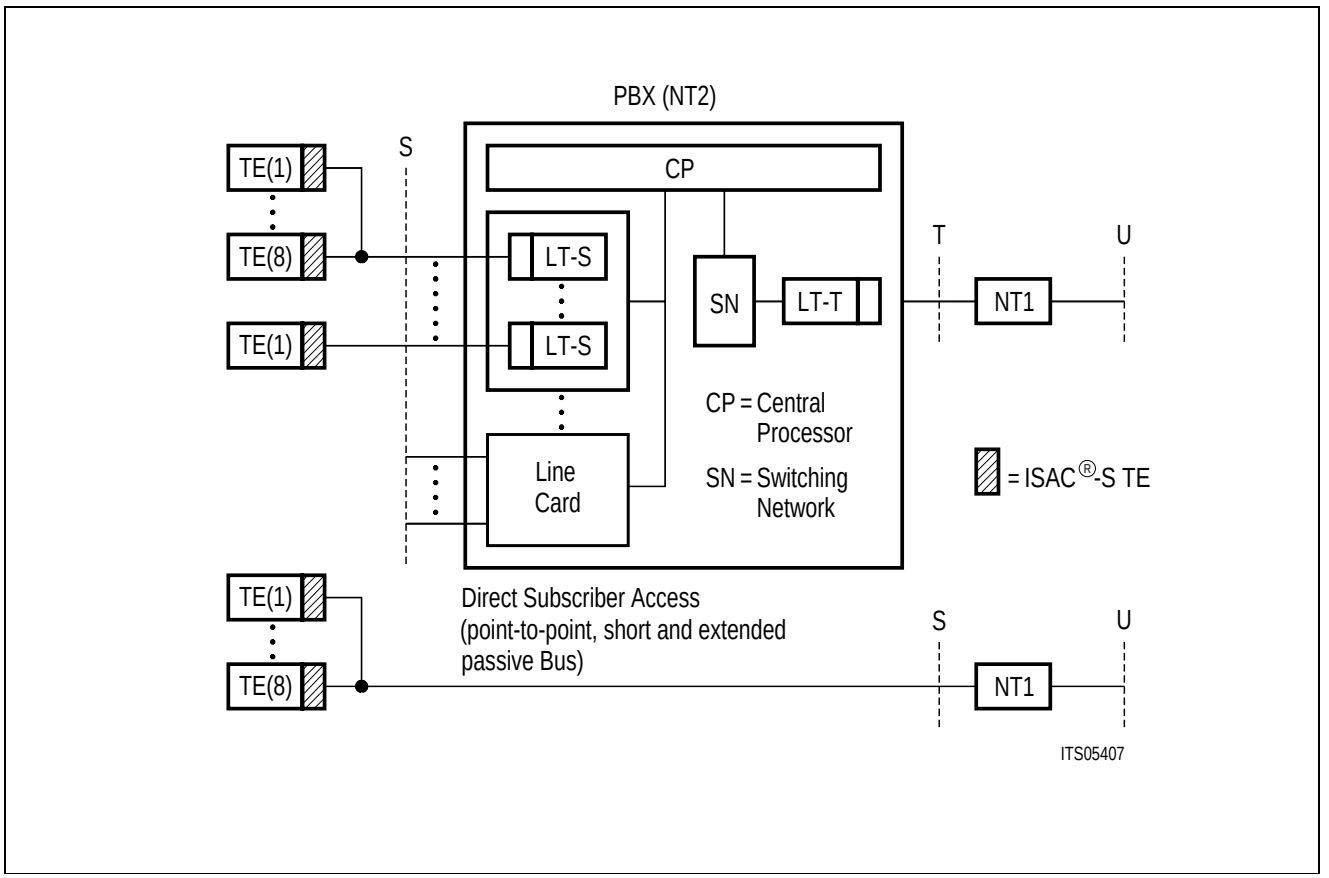


Figure 4
Applications of the ISAC®-S TE (ISDN-Basic Access)

Terminal Applications

The concept of the ISDN basic access is based on two circuit-switched 64 kbit/s B channels and a message oriented 16 kbit/s D channel for packetized data, signaling and telemetry information.

Figure 5 shows an example of an integrated **multifunctional ISDN-S** terminal using the ISAC-S TE. The ISAC-S TE provides the interface to the bus and separates the B- and D channels.

The D channel, containing signaling data and packet switched data, is processed by the LAPD controller contained in the ISAC-S TE and routed via a parallel µP interface to the terminal processor. The high level support of the LAPD protocol which is implemented by the ISAC-S TE allows the use of a low cost processor in cost sensitive applications.

The IOM-2 interface generated by the ISAC-S TE is used to connect different voice/data (V/D) application modules:

– sources/sinks for the D channel

– sources/sinks for the B1- and B2 channels.

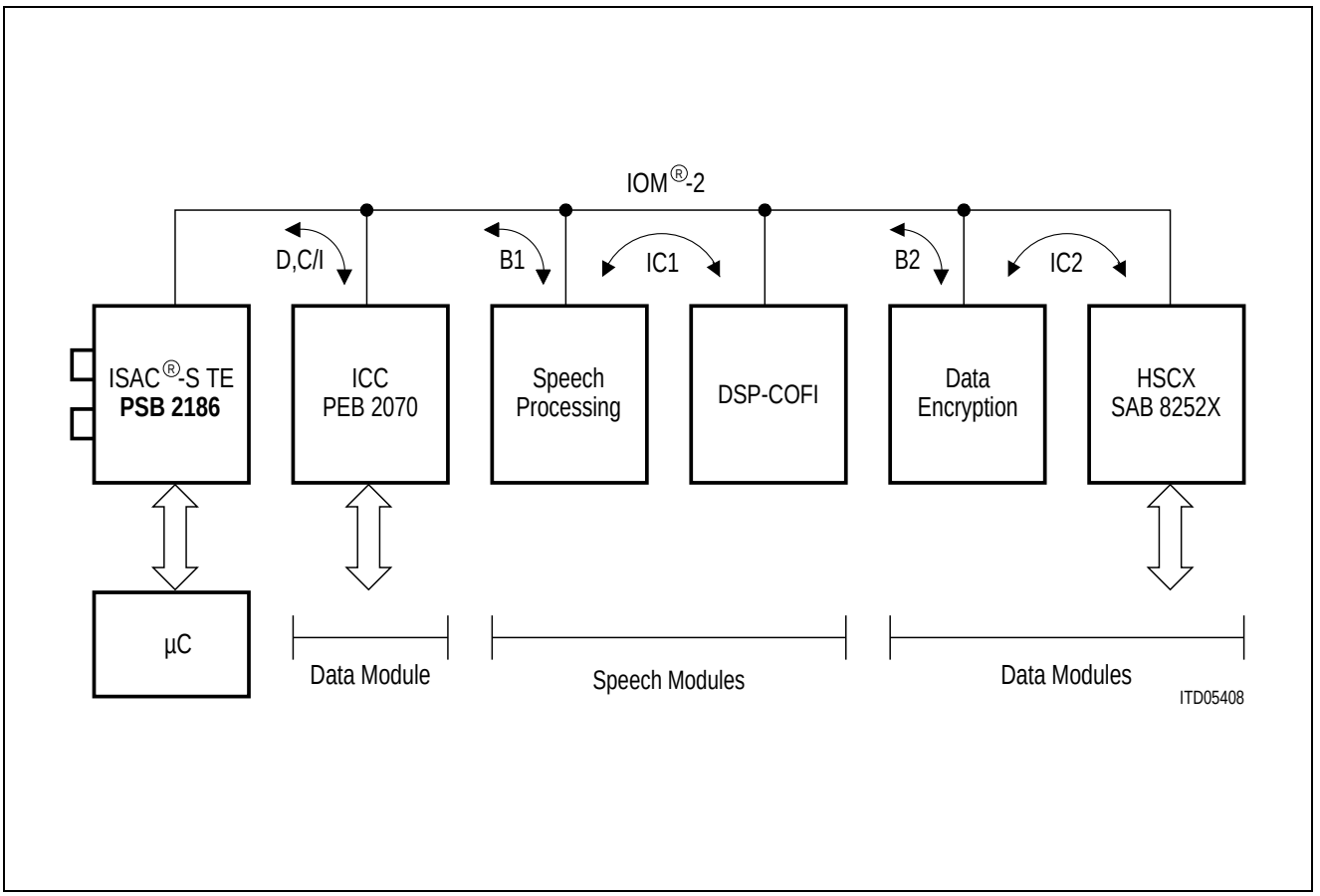


Figure 5
Example of an ISDN®-S TE Voice/Data Terminal

Up to eight D-channel components (ICC: ISDN Communication Controller PEB 2070) may be connected to the D- and C/I (Command/Indication) channels (TIC-bus). The ISAC-S TE and ICC handle contention autonomously.

Data transfers between the ISAC-S TE and the voice/data modules are done with the help of the IOM-MONITOR channel protocol. Each V/D module can be accessed by an individual address. The same protocol enables the control of IOM-terminal modules and the allocation of intercommunication channels inside the terminal. Two intercommunication channels IC1 and IC2 allow a 2×64 kbit/s transfer rate between voice/data modules.

In the example above (**figure 5**), one ICC is used for data packets in the D channel. A voice processor is connected to a programmable digital signal processing codec filter via IC1 and a data encryption module to a data device via IC2. B1 is used for voice communication, B2 for data communication.

Figure 6 shows the implementation of a ISDN feature phone using the ISAC-S TE and the Audio Ringing Codec Filter featuring speakerphone (PSB 2165, ARCOFI®-SP).

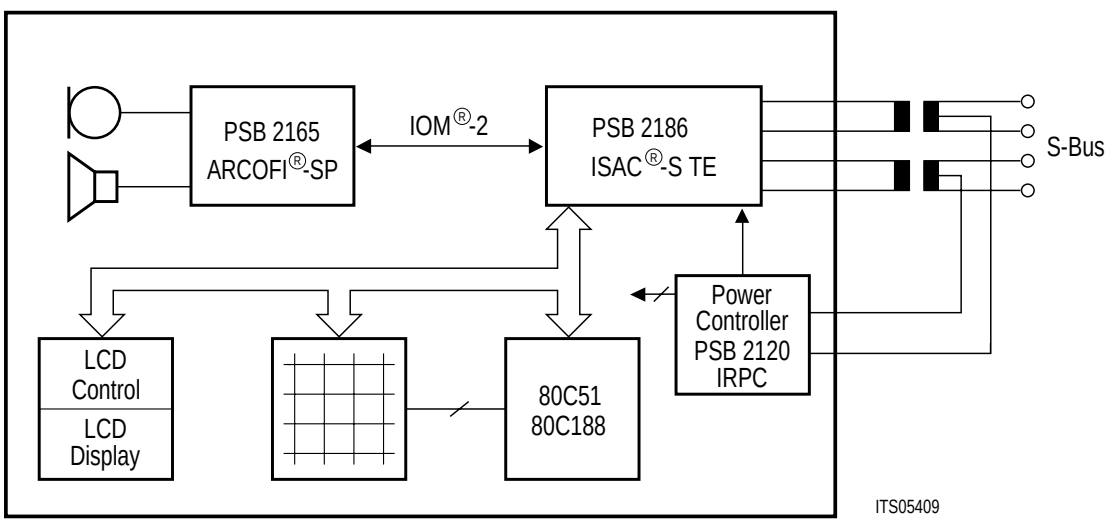


Figure 6
ISDN-Feature Telephone

1.4.2 Microprocessor Environment

The ISAC-S TE is especially suitable for cost-sensitive applications with single-chip microcontrollers (e.g. 8048, 8031, 8051). However, due to its programmable micro- processor interface and non-critical bus timing, it fits perfectly into almost any 8-bit microprocessor system environment. The microcontroller interface can be selected to be either of the Motorola type (with control signals $\overline{CS}$, R/$\overline{W}$, $\overline{DS}$) of the Siemens/Intel non-multiplexed bus type (with control signals $\overline{CS}$, $\overline{WR}$, $\overline{RD}$) or of the Siemens/Intel multiplexed address/data bus type ($\overline{CS}$, $\overline{WR}$, $\overline{RD}$, ALE).

An example how to connect the ISAC-S TE to a Siemens/Intel microcontroller is shown in **figure 7**.

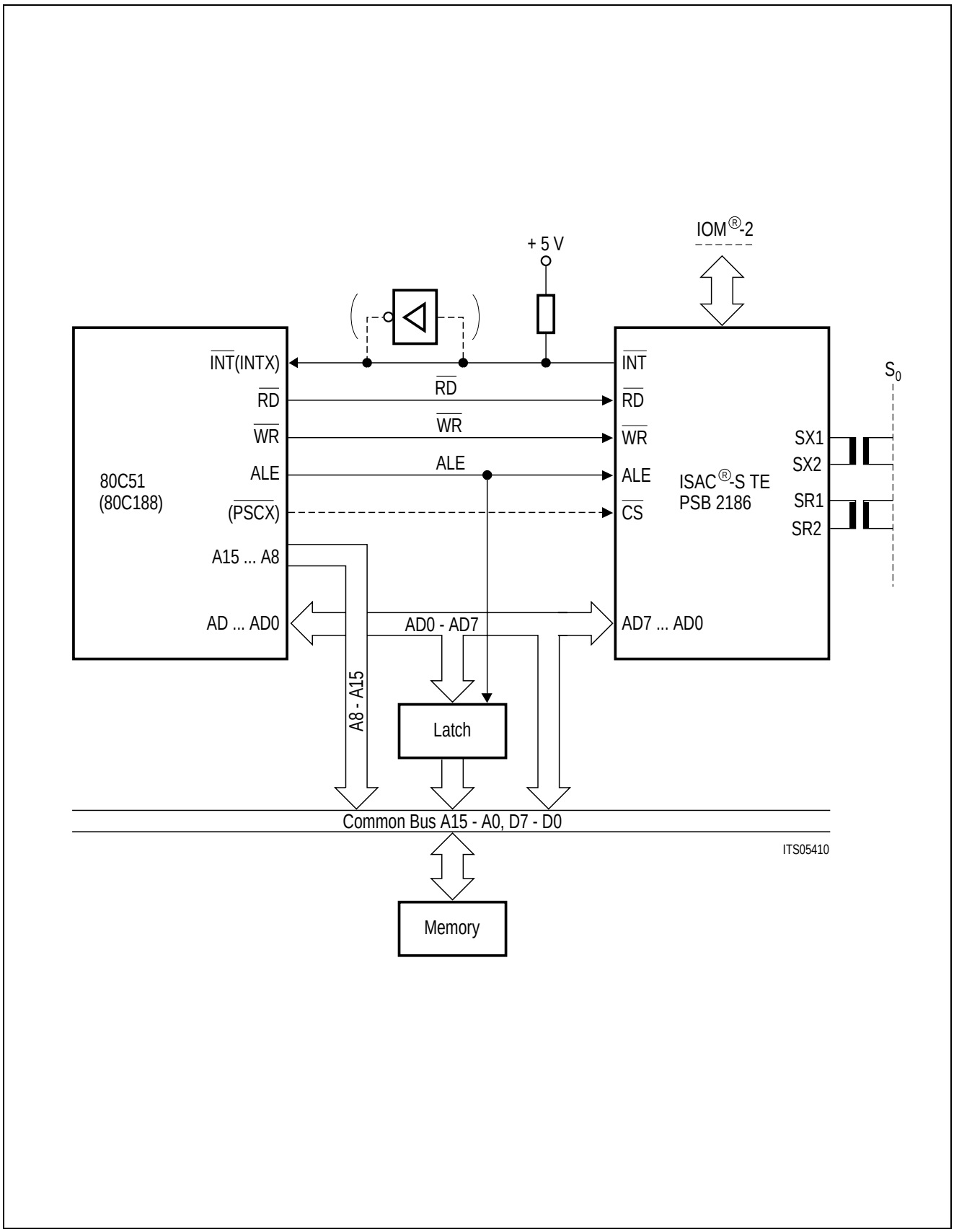


Figure 7
Connecting the ISAC®-S TE to Siemens/Intel Microcontroller

2 Functional Description

2.1 General Functions and Device Architecture

The functional block diagram of the ISAC-S TE is shown in **figure 8**.

The left-hand side of the diagram contains the layer-1 functions, according to CCITT I series recommendations:

– S-bus transmitter and receiver

– timing recovery and synchronization by means of digital PLL circuitry

– activation/deactivation

– access to S and Q channels

– handling of D channel

– test loops

– send single/continuous AMI pulses (diagnostics).

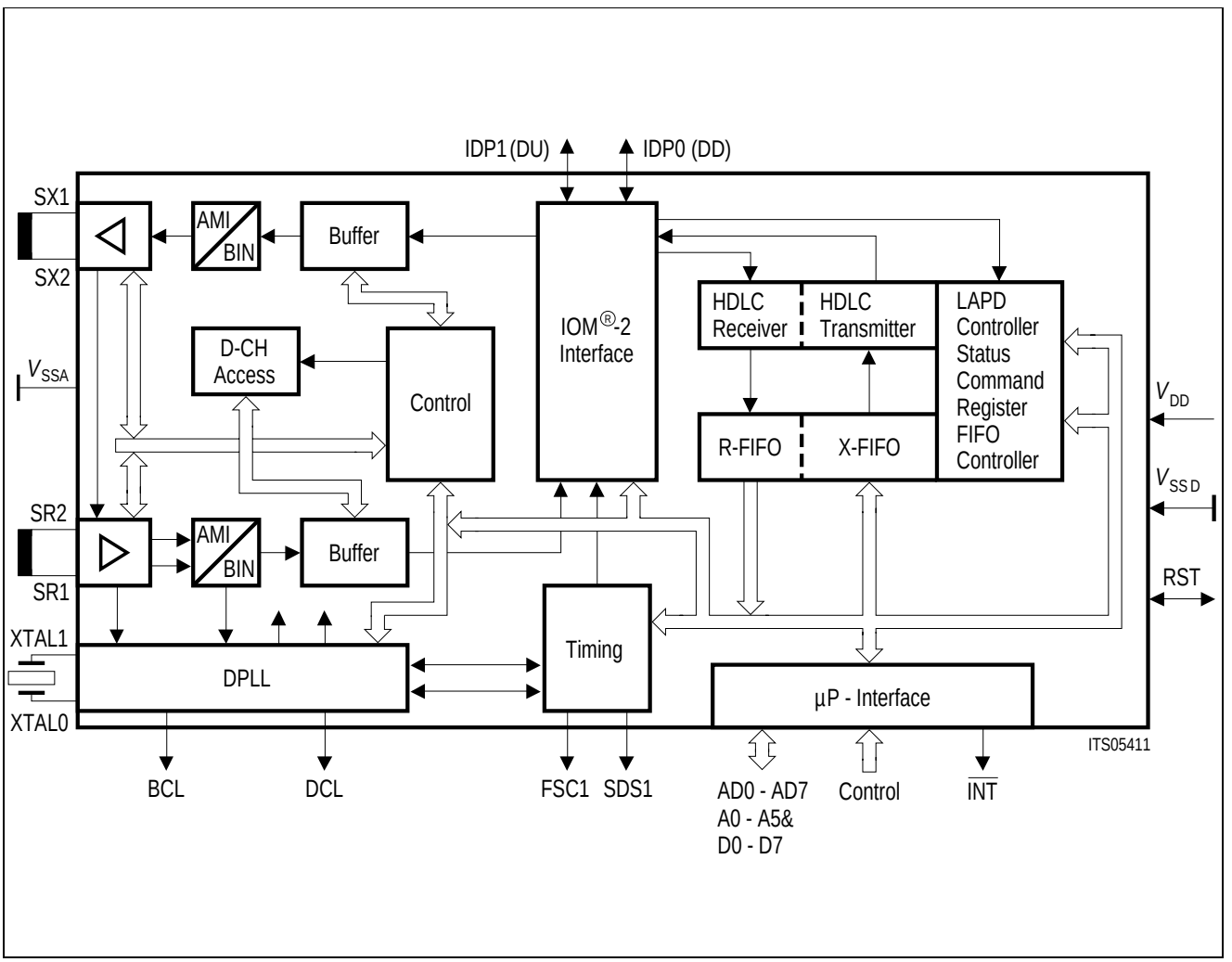


Figure 8
Architecture of the ISAC®-S TE

The right-hand side consists of:

– the serial interface logic for the IOM-2 interfaces, with B-channel switching capabilities
– the logic necessary to handle the D-channel messages (layer 2).

The latter consists of an HDLC receiver and an HDLC transmitter together with 64-byte deep FIFO's for efficient transfer of the messages to/from the user's CPU.

In a special HDLC-controller operating mode, the auto mode, the ISAC-S TE processes protocol handshakes (I- and S frames) of the LAPD (Link Access Procedure on the D channel) autonomously.

Control and monitor functions as well as data transfers between the user's CPU and the D- and B channels are performed by the 8-bit parallel µP-interface logic.

The IOM interface allows interaction between layer-1 and layer-2 functions. It implements D-channel collision resolution for connecting other layer-2 devices to the IOM interface (TIC bus), and the C/I and MONITOR channel protocols (IOM-2) to control peripheral devices.

The timing unit is responsible for the system clock and frame synchronization.

2.2 Interface and Operating Modes

The ISAC-S TE is configurable for the following application:

– ISDN terminals → TE mode

IOM®-2 Interface Mode (ADF2:IMS=1)

In this mode the IOM interface has the enhanced functionality of IOM-2. B-channel interfacing is performed directly via the IOM-2 interface.

The ISAC-S TE supports the IOM-2 terminal mode frame structure (3 channels) according to **figure 11** (**see chapter 2.3.1**).

The operating mode is shown in **table 1**.

Table 1
Operating Mode and Functions of Specific Pins of the ISAC®-S PSB 2186 in IOM®-2 Mode

Pin No. P-DIP-40-2	**11**	**12**	**19**
Pin No. P-LCC-44-1	**12**	**13**	**22**
Pin No. P-MQFP-64-1	**58**	**59**	**7**
Application	**DCL**	**FSC1**	**BCL**
TE	o:1536 kHz*	o:8 kHz*	o:768 kHz*

*) synchronized to the S/T interface o:output

The operating mode in relation to the timing recovery is illustrated in **figure 9**.

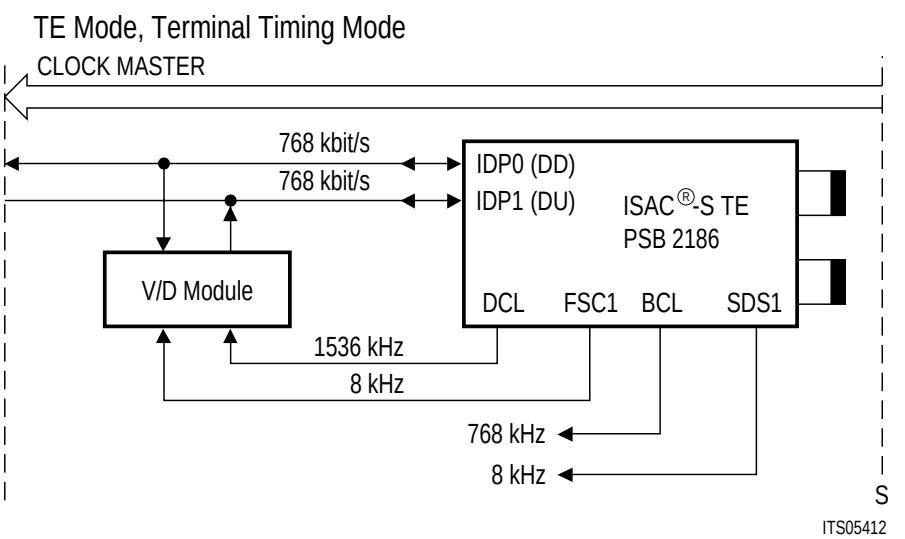


Figure 9
Operating Modes of ISAC®-S TE (IOM®-2)

2.3 IOM®-2 Mode Functions

2.3.1 Basic IOM®-2 Frame Structure

The IOM-2 is a generalization and enhancement of the IOM-1. While the basic frame structure is very similar, IOM-2 offers further capacity for the transfer of maintenance information. In terminal applications, the IOM-2 constitutes a powerful backplane bus offering intercommunication and sophisticated control capabilities for peripheral modules.

The channel structure of the IOM-2 is depicted in **figure 10**.

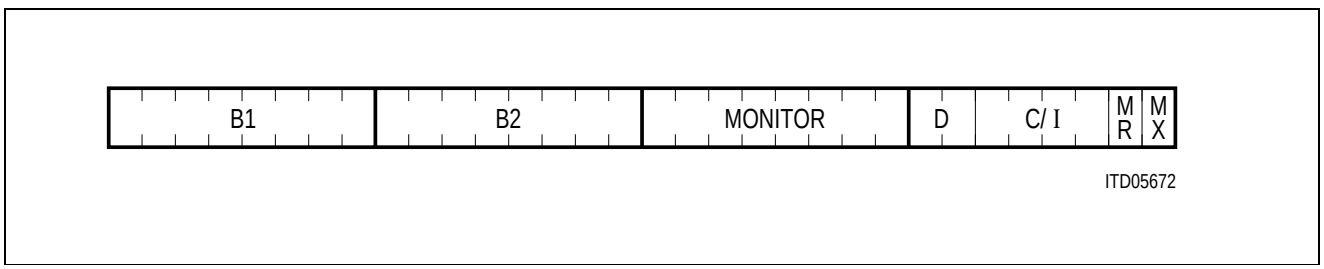


Figure 10
Channel Structure of IOM®-2

- The 64-kbit/s channels, B1 and B2, are conveyed in the first two octets.
- The third octet (monitor channel) is used for transferring maintenance information between the layer-1 functional blocks (SBCX, IECQ) and the layer-2 controller (**see chapter 2.3.4**).
- The fourth octet (control channel) contains

– two bits for the 16-kbit/s D channel

– four command/indication bits for controlling activation/deactivation and for additional control functions

– two bits MR and MX for supporting the handling of the MONITOR channel.

IOM®-2 TE Frame Structure

The frame is composed of three channels (**figure 11**):

- Channel 0 contains 144 kbit/s (for 2B+D) plus MONITOR and command/indication channels for the layer-1 device.
- Channel 1 contains two 64-kbit/s intercommunication channels plus MONITOR and command/indication channels for other IOM-2 devices.
- Channel 2 is used for IOM-bus arbitration (access to the TIC bus). Only the command/ indication bits are used in channel 2. **See section 2.3.6** for details.

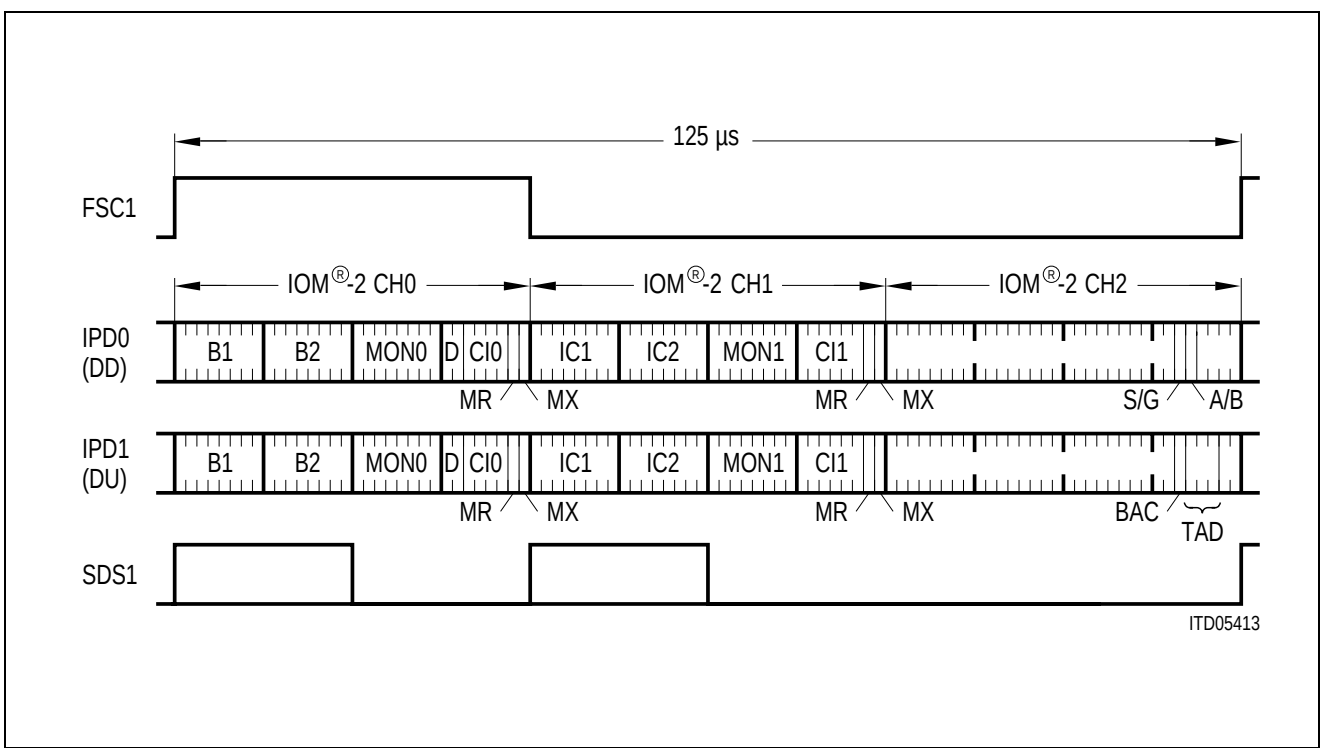


Figure 11
Definition of IOM®-2 Channels in Terminal Timing Mode

The IOM-2 signals are:

IDP0, 1 : 768 kbit/s
DCL : 1536-kHz output
FSC1 : 8-kHz output.

In addition, to support standard combos/data devices the following signals are generated as outputs:

BCL : 768-kHz bit clock
SDS1 : 8-kHz programmable data strobe signal for selecting one or both B/IC channel(s).

2.3.2 IOM®-2 Interface Connections

Output Driver Selection

The type of the IOM output is selectable via bit ODS (ADF2 register). Thus when inactive (not transmitting) IDP0, 1 are either high impedance (ODS=1) or open drain "1" (ODS=0).

Normally the IOM-2 interface is operated in the "open drain" mode (ODS=0) in order to take advantage of the bus capability. In this case pull-up resistors (1 kΩ – 5 kΩ) are required on IDP0 and IDP1.

IOM® OFF Function

In IOM-2 terminal mode (SPCR:SPM=0) the IOM interface can be switched off for external devices via IOF bit in ADF1 register. If IOF=1, the interface is switched off i.e. DCL, FSC1, IDP0/1 and BCL are high impedance.

IOM® Direction Control

For test applications, the direction of IDP0(DD) and IDP1(DU) can be reversed during certain time-slots within the IOM-2 frame. This is performed via the IDC bit in the SQXR register. For normal operation SQXR:IDC should be set to "0".

IOM® Data Ports in Terminal Mode

In this case the IOM has the 12-byte frame structure consisting of channels 0, 1 and 2 (**see figure 11**):

– IDP0 carries the 2B+D channels from the S/T interface, and the MONITOR 0 and C/I 0 channels coming from the S/T controller;

– IDP1 carries the MONITOR 0 and C/I 0 channels to the layer-1.

Channel 1 of the IOM interface is used for internal communication in terminal applications. Two cases have to be distinguished, according to whether the ISAC-S TE is operated as a master device (communication with slave devices via MONITOR 1 and C/I 1), or as a slave device (communication with one master via MONITOR 1 and C/I 1).
If IDC is set to "0" (master mode):

– IDP0 carries the MONITOR 1 and C/I 1 channels as output to peripheral (voice/data) devices;

– IDP0 carries the IC channels as output to other devices, if programmed (C×C1 – 0 = 01 in register SPCR).

If IDC is set to "1" (slave mode):

– IDP1 carries the MONITOR 1 and C/I 1 channels as output to a master device;

– IDP0 carries the IC channels as output to other devices, if programmed (C×C1 – 0 = 01 in register SPCR).

If required (cf. DIM2-0, MODE register), bit 5 of the last byte in channel 2 on IDP0 is used to indicate the S-bus state (stop/go bit) and bits 2 to 5 of the last byte are used for TIC-bus access arbitration (**see chapter 2.3.6**).

Figure 12 shows the connection in a multifunctional terminal with the ISAC-S TE as a master (**figure 12b**) and an ICC as a slave device.

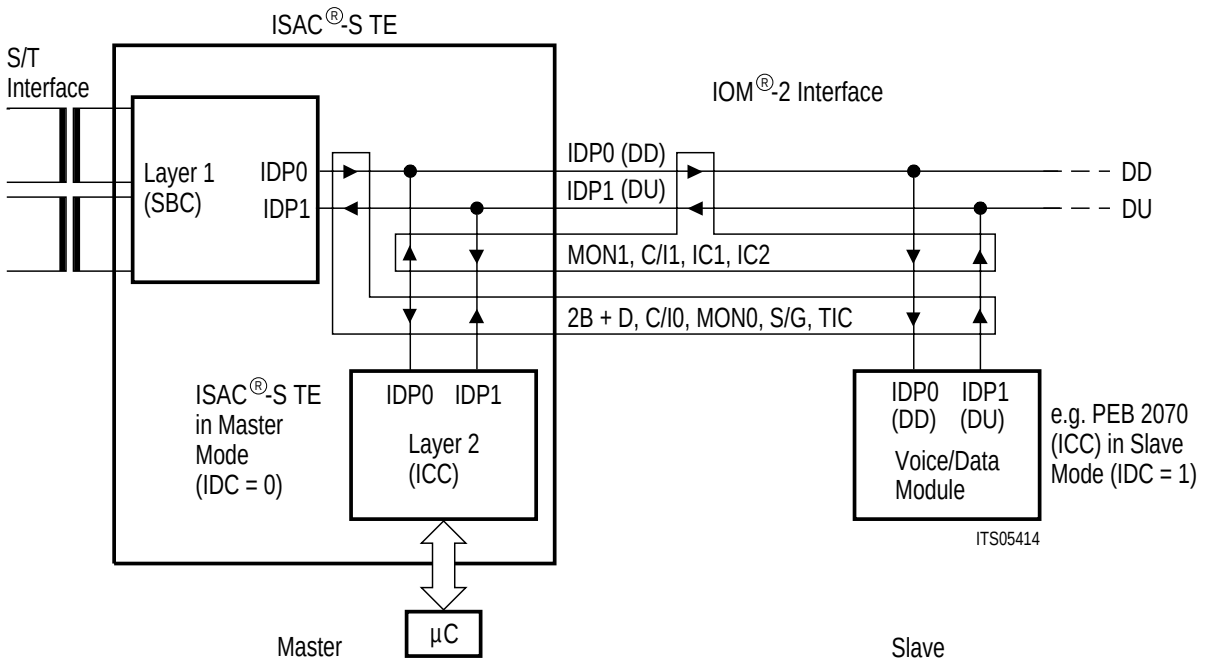
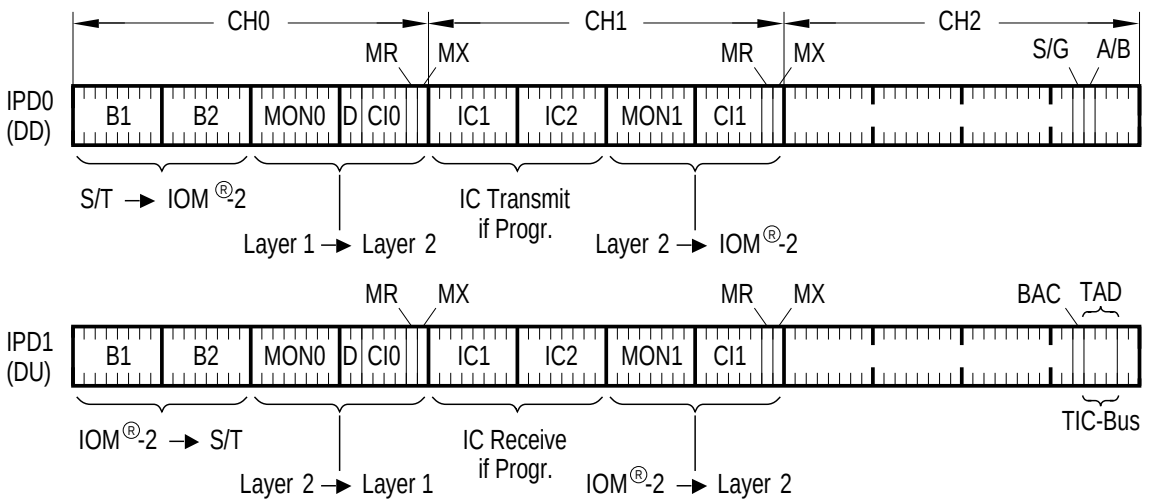


Figure 12a
IOM® Data Ports 0, 1 in Terminal Mode (SPCR:SPM=0)

(a) Master Mode (IDC = 0)



(b) Slave Mode (IDC = 1)

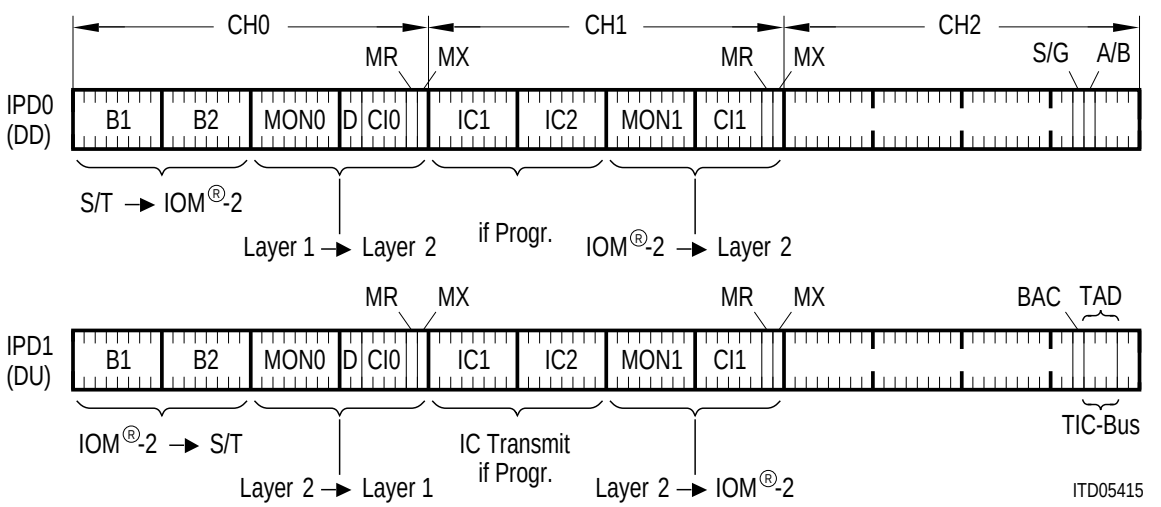


Figure 12b

2.3.3 µP Access to B and IC Channels

The microprocessor can access the B and IC (intercommunication) channels at the IOM-2 interface by reading the B1CR/B2CR or by reading and writing the C1R/C2R registers. Furthermore it is possible to loop back the B channels from/to the S/T interface or to loop back the IC channels from/to the IOM-2 interface without µP intervention.

These access and switching functions are selected with the channel connect bits (CxC1, CxC0) in the SPCR register (**table 2, figure 13**).

External B-channel sources (voice/data modules) connected to the IOM-2 interface can be disconnected with the IOM off function (ADF1:IOF) in order to not disturb the B-channel access (**see figure 13**).

If the B-channel access is used for transferring 64-kbit/s voice/data information directly from the µP port to the ISDN S/T interface, the access can be synchronized to the IOM interface by means of a synchronous transfer interrupt programmed in the STCR register.

Table 2
µP Access to B/IC Channels (IOM®-2)

C×C1	C×C0	C×R	C×R	B×CR	Output to IOM-2	Applications
		Read	Write	Read		
0	0	IC×	–	B×	–	B× monitoring, IC× monitoring
0	1	IC×	IC×	B×	IC×	B× monitoring, IC× looping from/to IOM-2
1	0	–	B×	B×	B×	B× access from/to S_0; transmission of a constant value in B× channel to S_0
1	1	B×	B×	–	B×	B× looping from S_0; transmission of a variable pattern in B× channel to S_0

Note: x=1 for channel 1 or 2 for channel 2

The general sequence of operations to access the B/IC channels is:

(set configuration register SPCR)
Program synchronous interrupt (ST0)
SIN –> Read register (B×CR, C×R)
(write register)
Acknowledge SIN (SC0)

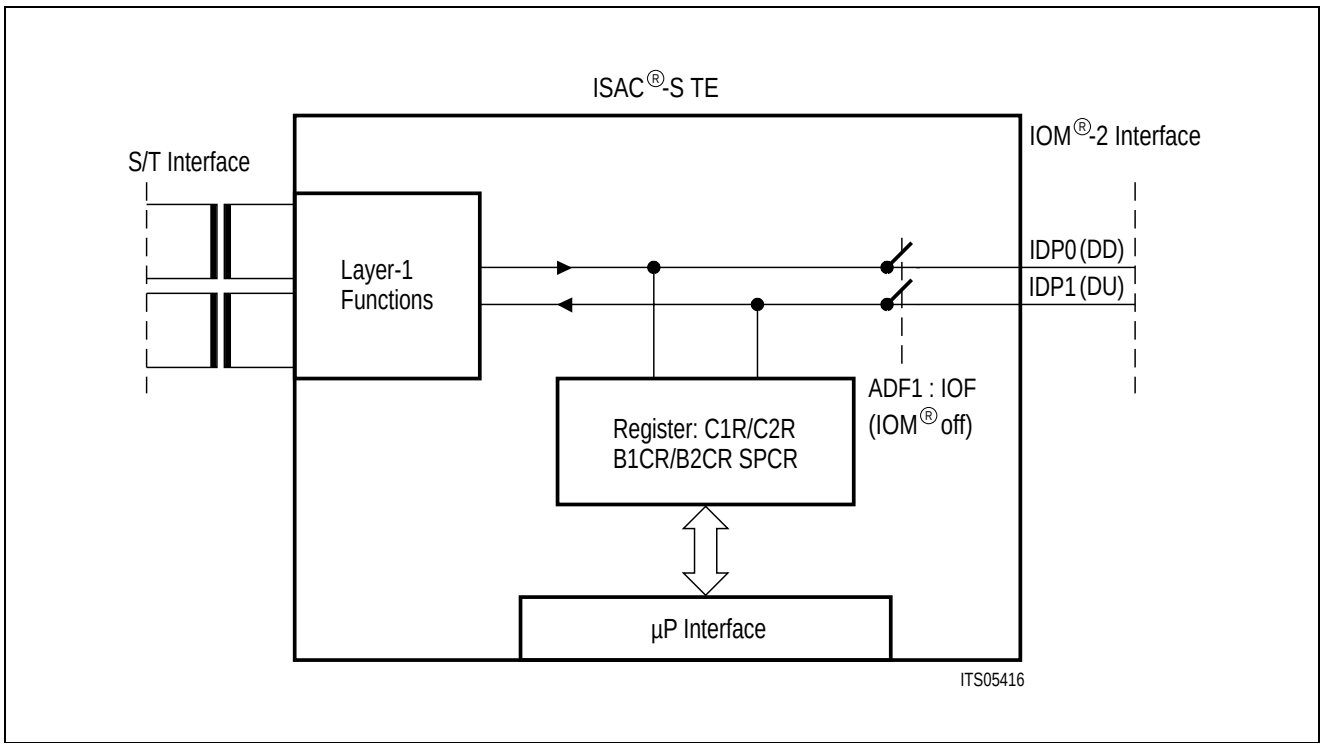


Figure 13
Principle of B/IC-Channel Access

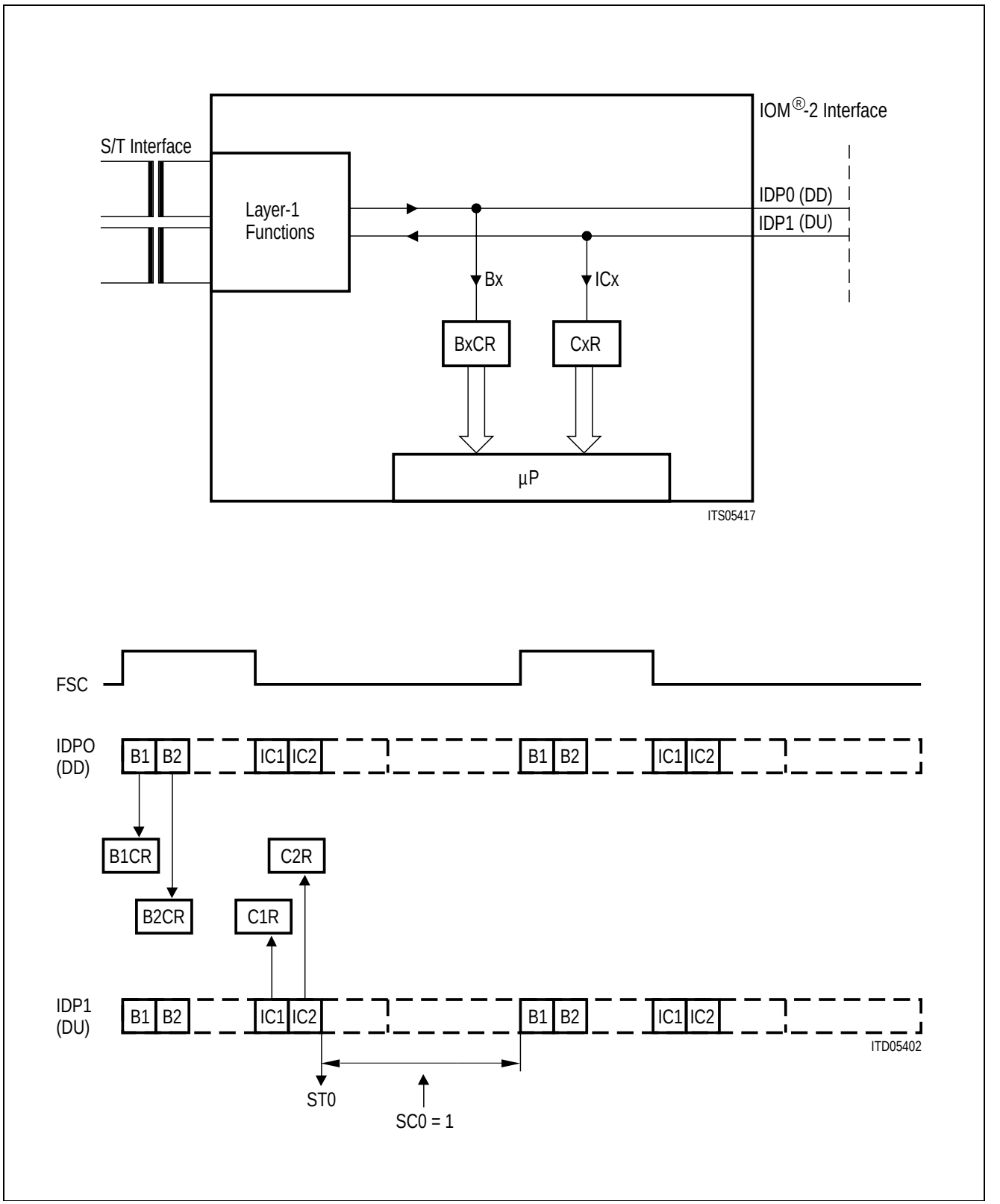
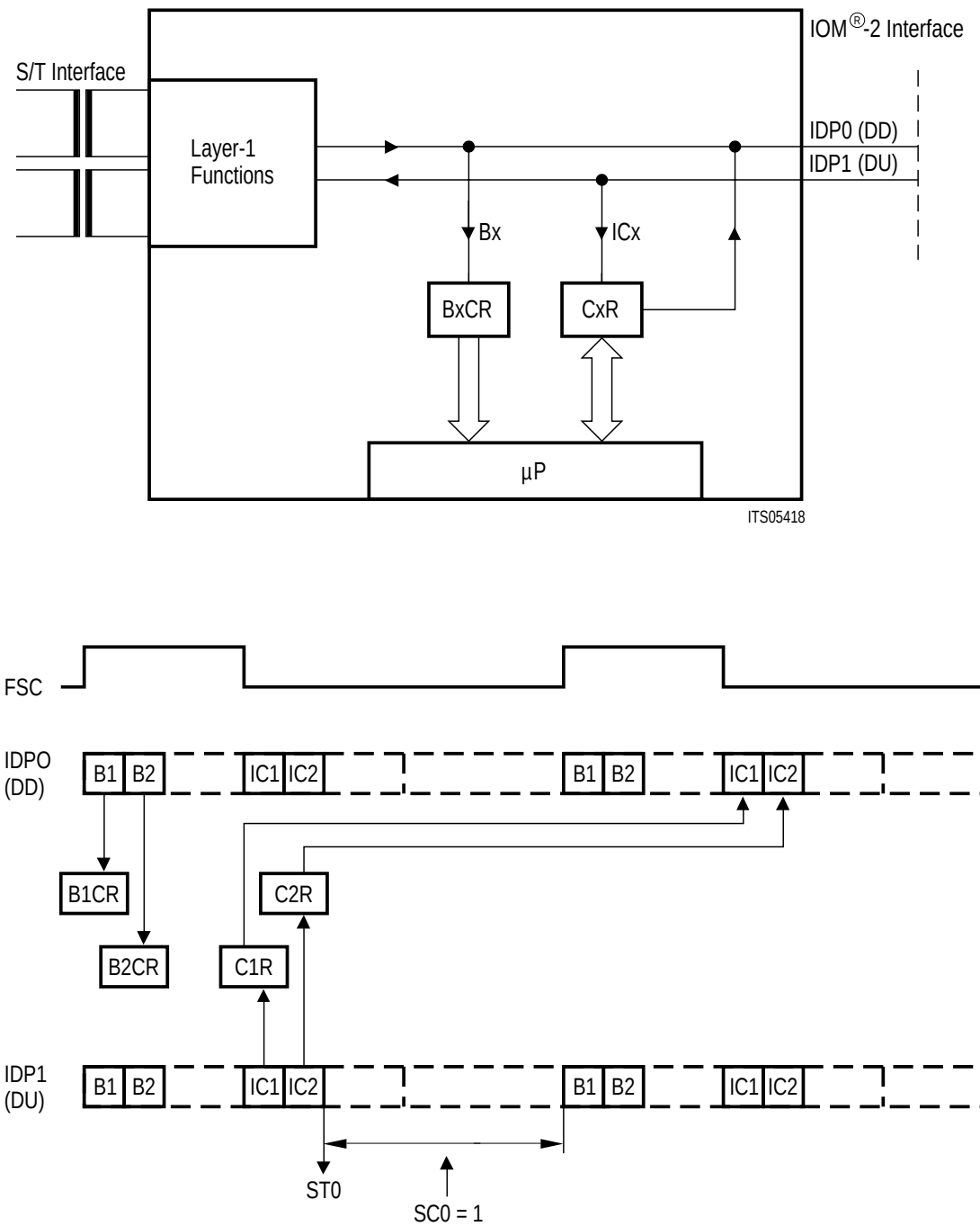
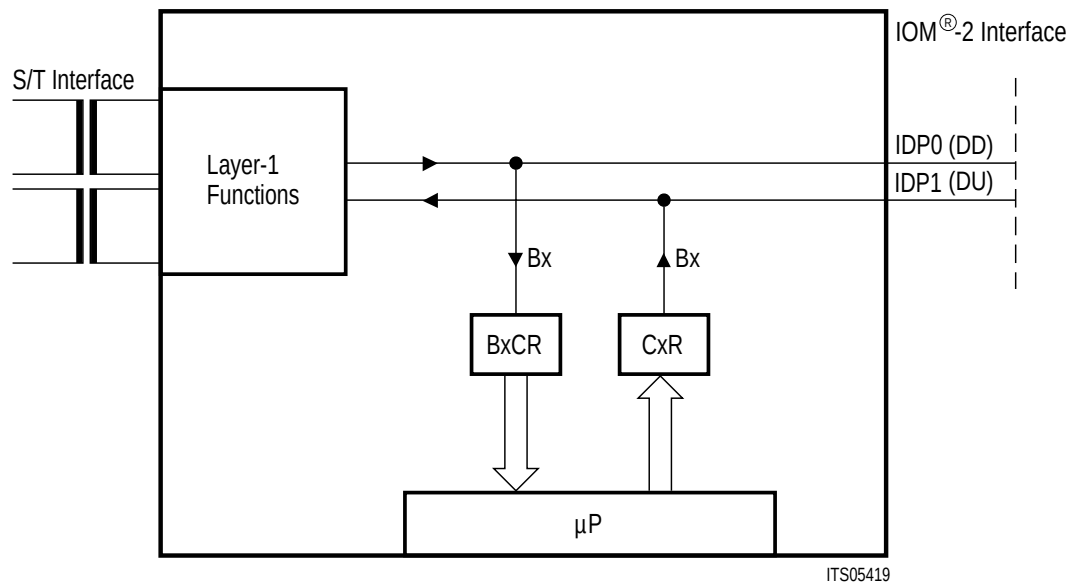


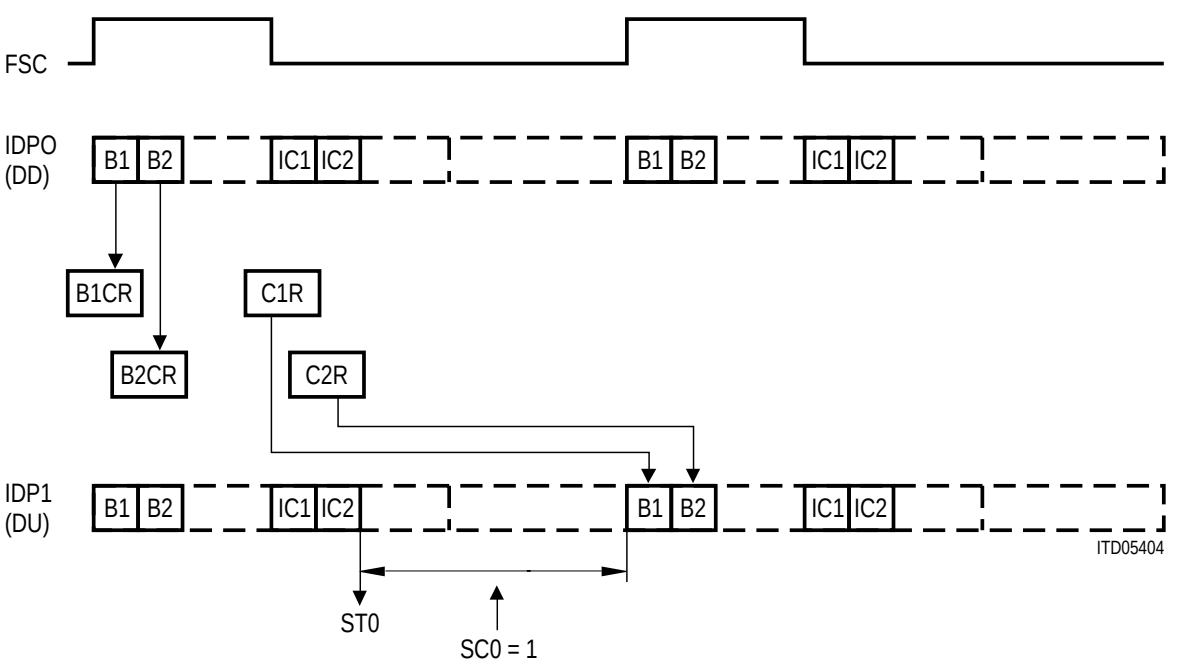
Figure 14
Access to B and IC Channels in IOM®-2 Terminal Mode

(a) SPCR:C×C1, C×C0 = 00
B× monitoring, IC× monitoring (SQXR:IDC=0)

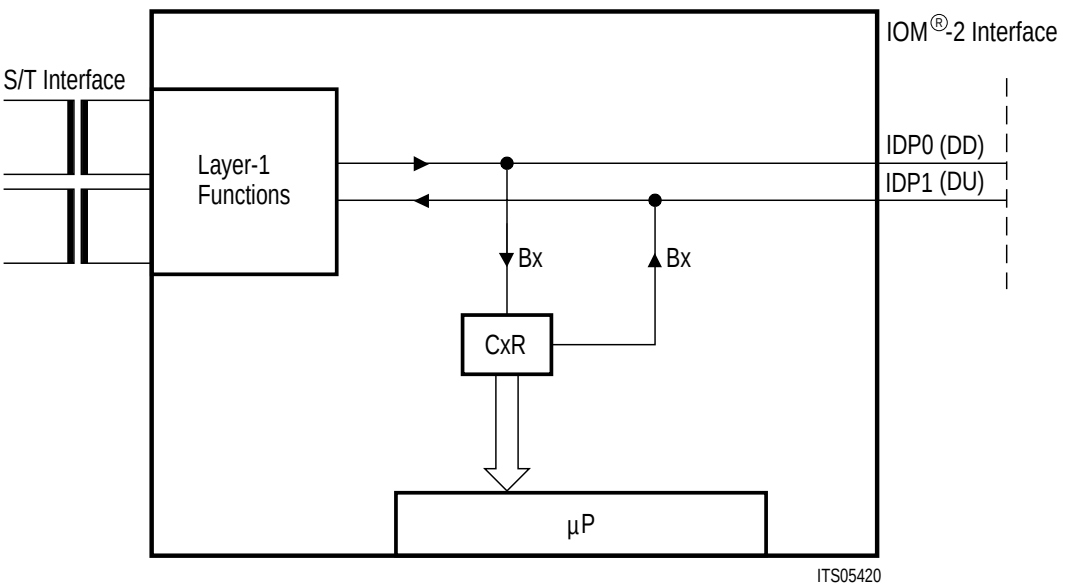


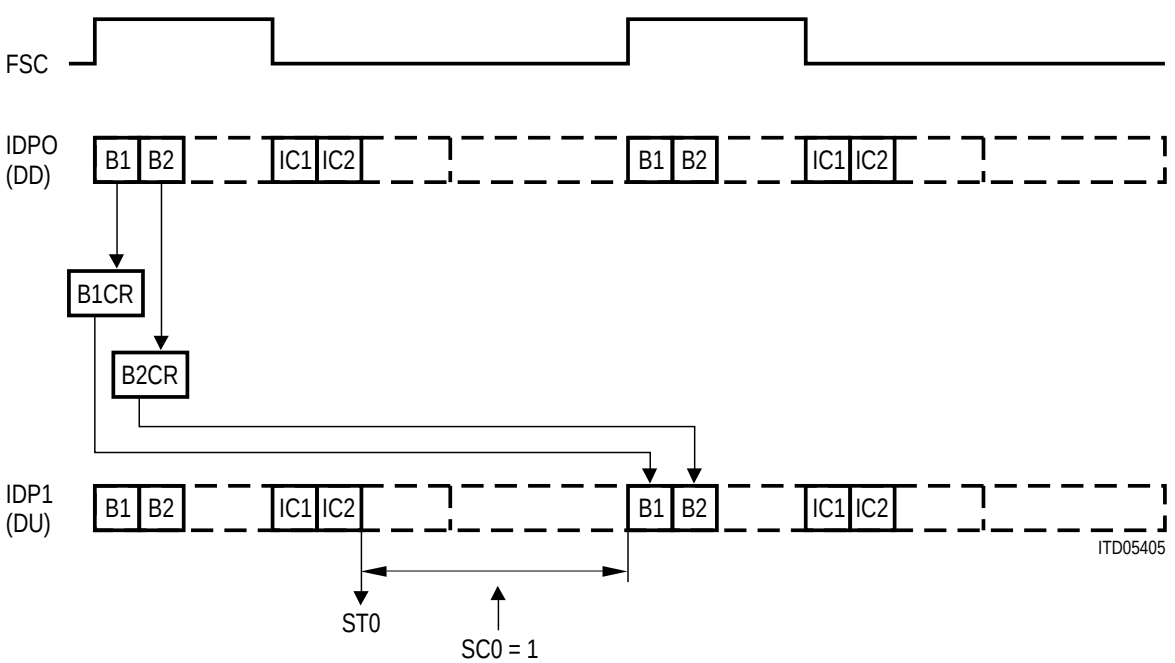
(b) SPCR:C×C1, C×C0 = 01
B× monitoring, IC× looping (SQXR:IDC=0)





(c) SPCR:C×C1, C×C0 = 10
B× access from/to S/T
transmission of constant value to S/T





(d) SPCR:C×C1, C×C0 = 11
B× looping from/to S/T
transmission of variable pattern to S/T

2.3.4 MONITOR Channel Handling

In IOM-2 mode, the MONITOR channel protocol is a handshake protocol used for high speed information exchange between the ISAC-S TE and other devices, in MONITOR channel "0" or "1" (**see figure 11**). In the non-TE mode, only one MONITOR channel is available ("MONITOR channel 0").

The MONITOR channel protocol is necessary:

- For programming and controlling devices attached to the IOM. Examples of such devices are: layer-1 transceivers (using MONITOR channel 0), and peripheral V/D modules that do not need a parallel microcontroller interface (MONITOR channel 1), such as the Audio Ringing Codec Filter PSB 2165.
- For data exchange between two microcontroller systems attached to two different devices on one IOM-2 backplane. Use of the MONITOR channel avoids the necessity of a dedicated serial communication path between the two systems. This greatly simplifies the system design of terminal equipment (**figure 17**).

Note: There is normally no necessity for monitor channel operations over "MONITOR channel 0" since the internal layer-1 part of the ISAC-S TE does not support this function. The implemented MONITOR handler can however be used with external layer-1 transceivers in case only the ICC part of the ISAC-S TE is used (ADF1:TEM).

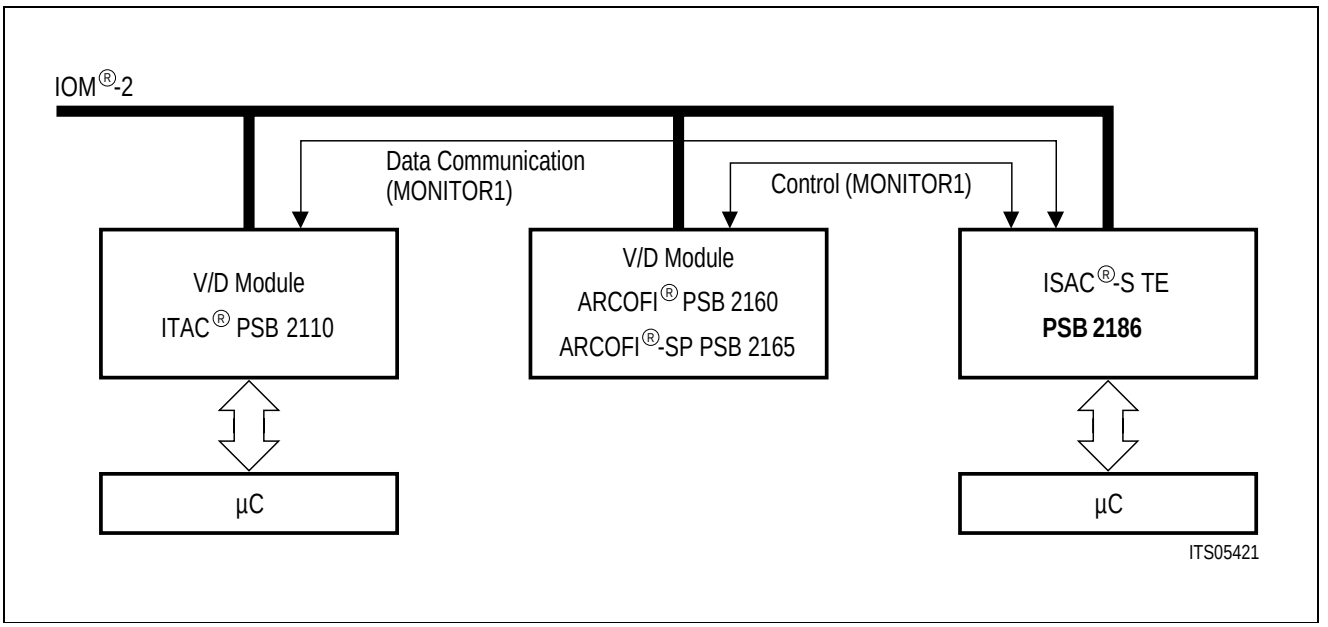


Figure 15
Examples of MONITOR Channel Applications in IOM®-2 TE Mode

The MONITOR channel operates on an asynchronous basis. While data transfers on the bus take place synchronized to frame sync, the flow of data is controlled by a handshake procedure using the MONITOR Channel Receive (MR0 or 1) and MONITOR Channel Transmit (MX0 or 1) bits. For example: data is placed onto the MONITOR channel and the MX bit is activated. This data will be transmitted repeatedly once per 8-kHz frame until the transfer is acknowledged via the MR bit.

The microprocessor may either enforce a "1" (idle) in MR, MX by setting the control bit MRC1, 0 or MXC1, 0 to "0" (MONITOR Control Register MOCR), or enable the control of these bits internally by the ISAC-S TE according to the MONITOR channel protocol. Thus, before a data exchange can begin, the control bit MRC(1, 0) or MXC(1, 0) should be set to "1" by the microprocessor.

The MONITOR channel protocol is illustrated in **figure 16**. Since the protocol is identical in MONITOR channel 0 and MONITOR channel 1 (available in TE mode only), the index 0 or 1 has been left out in the illustration.

The relevant status bits are:

- MONITOR Channel Data Received MDR (MDR0, MDR1)
- MONITOR Channel End of Reception MER (MER0, MER1)

for the **reception** of MONITOR data, and

- MONITOR Channel Data Acknowledged MDA (MDA0, MDA1)
- MONITOR Channel Data Abort MAB (MAB0, MAB1)

for the **transmission** of MONITOR data (register: MOSR)

In addition, the status bit:

MONITOR Channel Active MAC (MAC0, MAC1)
indicates whether a transmission is in progress (register: STAR).

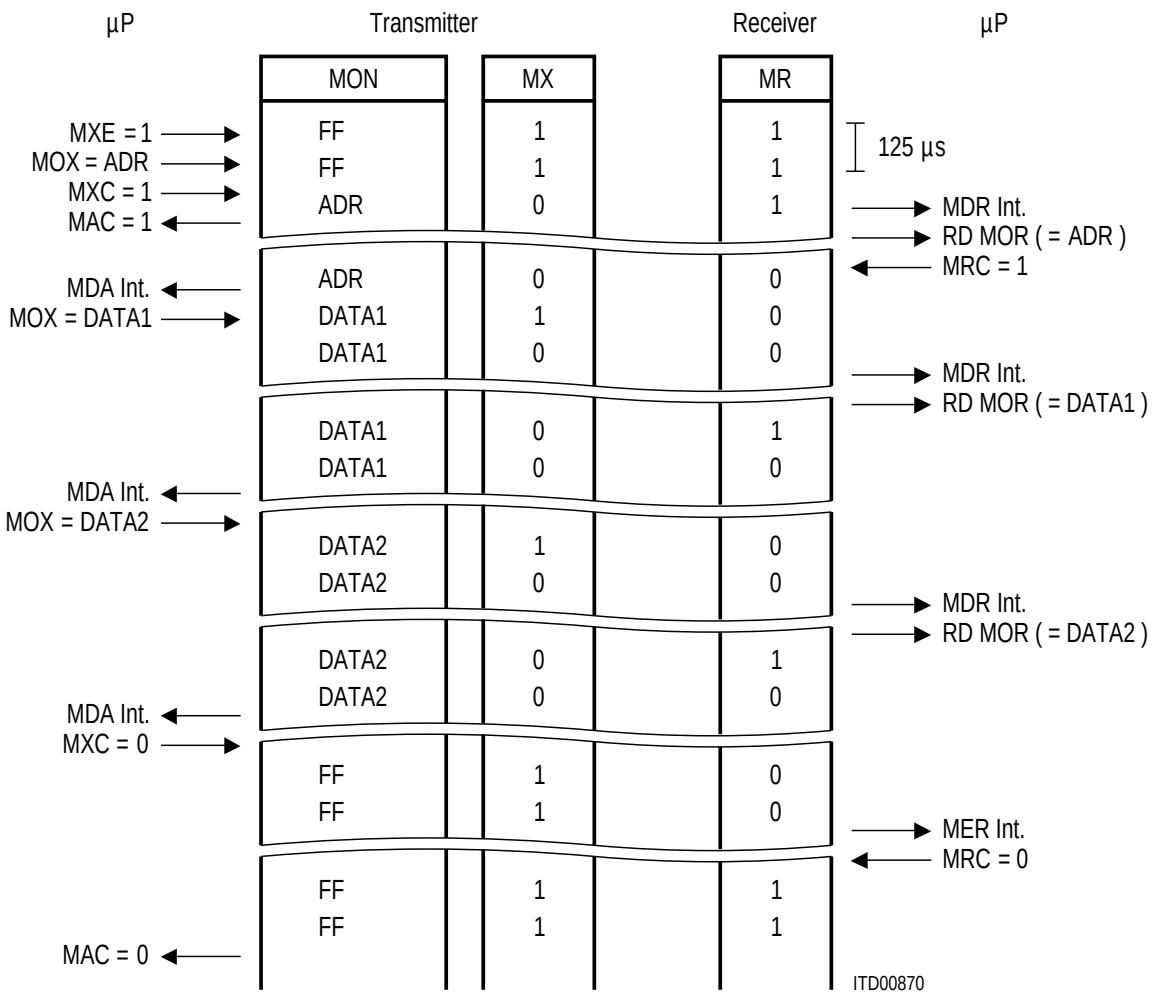


Figure 16
MONITOR Channel Protocol (IOM®-2)

Before starting a transmission, the microprocessor should verify that the transmitter is inactive, i.e. that a possible previous transmission has been terminated. This is indicated by a "0" in the MONITOR Channel Active MAC status bit.

After having written the MONITOR Data Transmit (MOX) register, the microprocessor sets the MONITOR Transmit Control bit MXC to "1". This enables the MX bit to go active (0), indicating the presence of valid MONITOR data (contents of MOX) in the corresponding frame. As a result, the receiving device stores the MONITOR byte in its MONITOR Receive MOR register and generates an MDR interrupt status.

Alerted by the MDR interrupt, the microprocessor reads the MONITOR Receive (MOR) register. When it is ready to accept data (e.g. based on the value in MOR, which in a point-to-multipoint application might be the address of the destination device), it sets the MR control bit MRC to "1" to enable the receiver to store succeeding MONITOR channel bytes and acknowledge them according to the MONITOR channel protocol. In addition, it enables other MONITOR channel interrupts by setting MONITOR Interrupt Enable to "1".

As a result, the first MONITOR byte is acknowledged by the receiving device setting the MR bit to "0". This causes a MONITOR Data Acknowledge MDA-interrupt status at the transmitter.

A new MONITOR data byte can now be written by the microprocessor in MOX. The MX bit is still in the active (0) state. The transmitter indicates a new byte in the MONITOR channel by returning the MX bit active after sending it once in the inactive state. As a result, the receiver stores the MONITOR byte in MOR and generates a new MDR-interrupt status. When the microprocessor has read the MOR register, the receiver acknowledges the data by returning the MR bit active after sending it once in the inactive state. This in turn causes the transmitter to generate an MDA-interrupt status.

This "MDA interrupt – write data – MDR interrupt – read data – MDA interrupt" handshake is repeated as long as the transmitter has data to send. Note that the MONITOR channel protocol imposes no maximum reaction times to the microprocessor.

When the last byte has been acknowledged by the receiver (MDA-interrupt status), the microprocessor sets the MONITOR Transmit Control bit MXC to "0". This enforces an inactive ("1") state in the MX bit. Two frames of MX inactive signifies the end of a message. Thus, a MONITOR Channel End of Reception MER-interrupt status is generated by the receiver when the MX bit is received in the inactive state in two consecutive frames. As a result, the microprocessor sets the MR control bit MRC to 0, which in turn enforces an inactive state in the MR bit. This marks the end of the transmission, making the MONITOR Channel Active MAC bit return to "0".

During a transmission process, it is possible for the receiver to ask a transmission to be aborted by sending an inactive MR bit value in two consecutive frames. This is effected by the microprocessor writing the MR control bit MRC to "0". An aborted transmission is indicated by a MONITOR Channel Data Abort MAB-interrupt status at the transmitter.

2.3.5 C/I-Channel Handling

The command/indication channel carries real-time status information between the ISAC-S TE and another device connected to the IOM.

1) One C/I channel (called C/I0) conveys the commands and indications between the layer-1 and the layer-2 parts of the ISAC-S TE. It can be accessed by an external layer-2 device e.g. to control the layer-1 activation/deactivation procedures. C/I0 channel access may be arbitrated via the TIC bus access protocol. In this case the arbitration is done in C/I channel 2 (**see figure 11**).

   The C/I0 channel is accessed via register CIR0 (in receive direction, layer-1 to layer-2) and register CIX0 (in transmit direction, layer-2 to layer-1). The C/I0 code is four bits long.

   A listing and explanation of the layer-1 C/I codes can be found in **chapter 3.4**.

   In the receive direction, the code from layer-1 is continuously monitored, with an interrupt being generated anytime a change occurs (ISTA:CISQ). A new code must be found in two consecutive IOM frames to be considered valid and to trigger a C/I code change interrupt status (double last look criterion).

   In the transmit direction, the code written in CIX0 is continuously transmitted in C/I0.

2) A second C/I channel (called C/I1) can be used to convey real time status information between the ISAC-S TE and various non-layer-1 peripheral devices e.g. PSB 2160 ARCOFI. The channel consists of six bits in each direction (**see figure 11**).

   The C/I1 channel is accessed via registers CIR1 and CIX1. A change in the received C/I1 code is indicated by an interrupt status without double last look criterion.

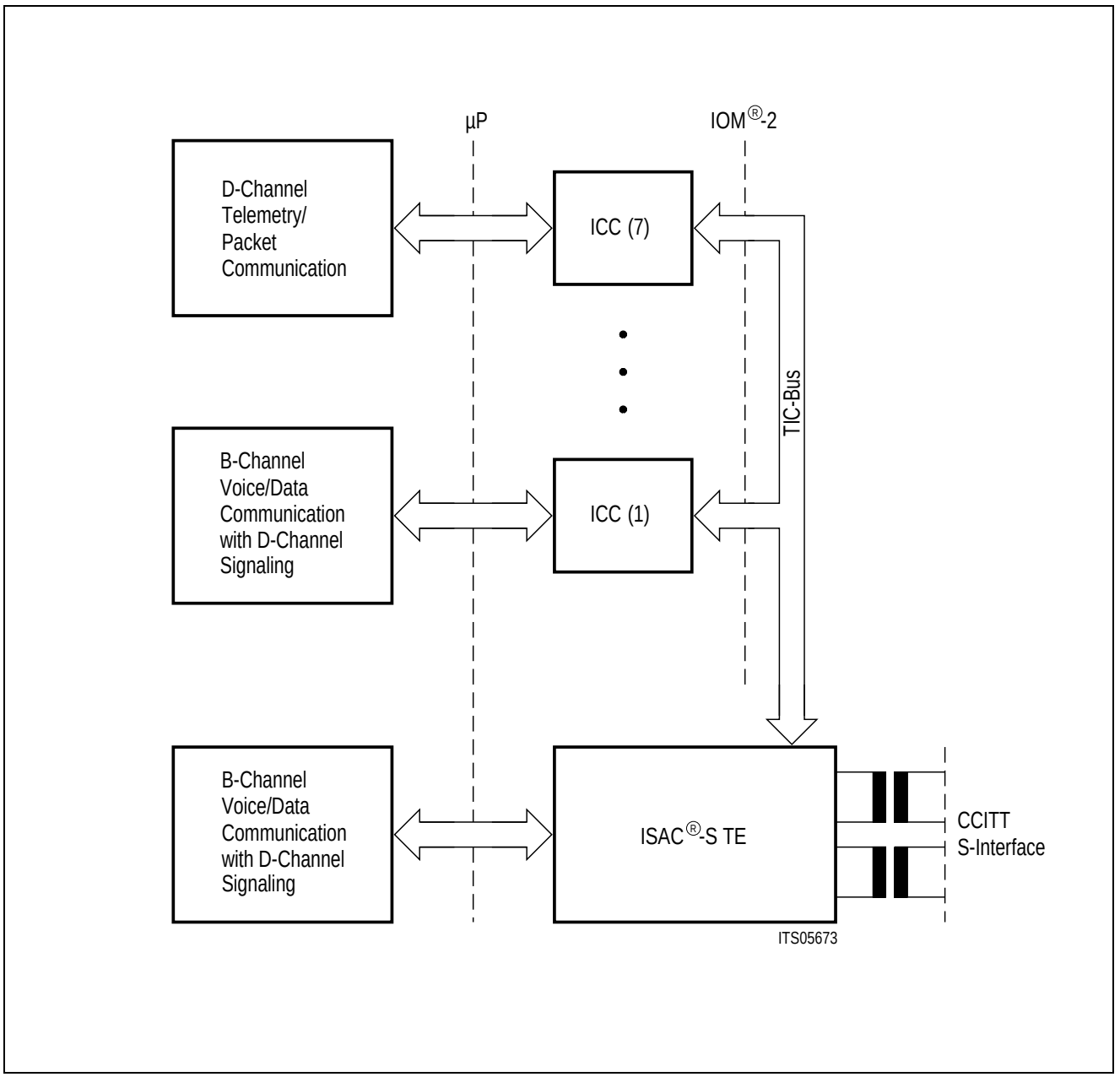


Figure 17
Applications of TIC Bus in IOM®-2 Bus Configuration

2.3.6 TIC-Bus Access

In IOM-2 interface mode the TIC-bus capability is only available in TE mode. The arbitration mechanism implemented in the last octet of IOM channel 2 of the IOM allows the access of external communication controllers (up to 7) to the layer-1 functions provided in the ISAC-S TE and to the D channel. (TIC bus; **see figure 17**). To this effect the outputs of the controllers (ICC:ISDN Communication Controller PEB 2070) are wired-or-and connected to pin IDP1. The inputs of the ICCs are connected to pin IDP0. External pull-up resistors on IDP0/1 are required. The arbitration mechanism must be activated by setting MODE:DIM2–0=001 (**see chapter 4.1.7**).
An access request to the TIC bus may either be generated by software (µP access to the C/I

channel) or by the ISAC-S TE itself (transmission of an HDLC frame). A software access request to the bus is effected by setting the BAC bit (CIX0 register) to "1".

In the case of an access request, the ISAC-S TE checks the Bus Accessed-bit (bit 5 of IDP1 last octet of CH2, **see figure 18**) for the status "bus free", which is indicated by a logical "1". If the bus is free, the ISAC-S TE transmits its individual TIC-bus address programmed in the STCR register. The TIC bus is occupied by the device which sends its address error-free. If more than one device attempt to seize the bus simultaneously, the one with the lowest address values wins.

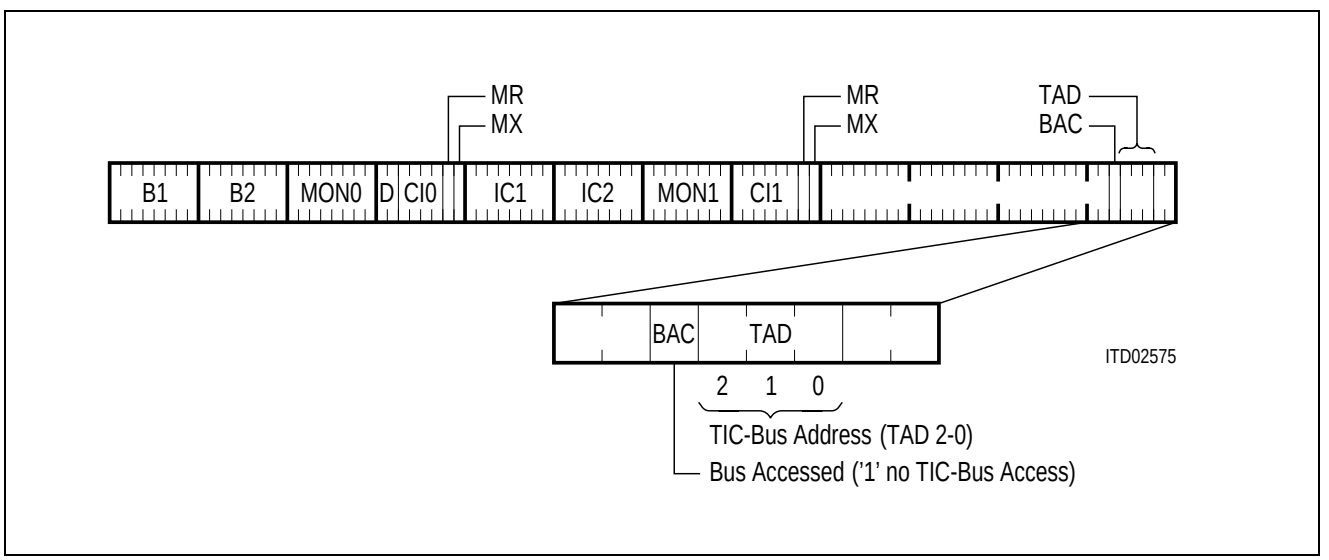


Figure 18
Structure of Last Octet of CH2 on IDP1 (DU)

When the TIC bus is seized by the ISAC-S TE, the bus is identified to other devices as occupied via the IDP1 C/I Bus Accessed-bit state "0" until the access request is withdrawn. After a successful bus access, the ISAC-S TE is automatically set into a lower priority class, that is, a new bus access cannot be performed until the status "bus free" is indicated in two successive frames.

If none of the devices connected to the IOM interface request access to the D- and C/I channels, the TIC-bus address 7 will be present. The device with this address will therefore have access, by default, to the D and C/I channels.

Note: Bit BAC (CIX0 register) should be reset by the µP when access to the C/I channels is no more requested, to grant other devices access to the D- and C/I channels.

The availability of the S/T interface D channel is indicated in bit 5 "Stop/Go" (S/G) of the IDP0 last octet of C/I channel (**figure 19**).

S/G = 1 : stop
S/G = 0 : go

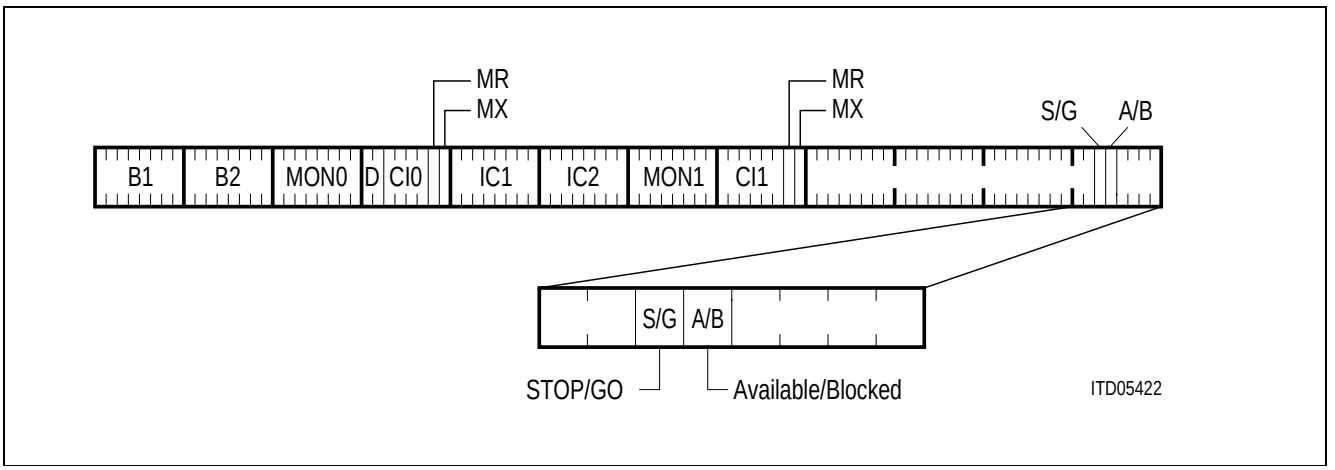


Figure 19
Structure of Last Octet of CH2 on IDP0 (DD)

The stop/go bit is available to other layer-2 devices connected to the IOM to determine if they can access the S/T bus D channel.

The available busy bit is not influenced by the ISAC-S TE.

2.4 Layer-1 Functions for the S/T Interface

– line transceiver functions for the S/T interface according to the electrical specifications of CCITT I.430;

– conversion of the frame structure between IOM and S/T interface;

– conversion from/to binary to/from pseudo-ternary code;

– level detect;

– S/T-timing generation using IOM timing synchronous to system, or vice versa;

– D-channel access control and priority handling;

– activation/deactivation procedures, triggered by primitives received over the IOM C/I channel or by INFO's received from the line;

– frame alignment;

– execution of test loops.

For a block diagram, **see figure 8**.

The wiring configurations in user premises, in which the ISAC-S TE can be used are illustrated in **figure 20**.

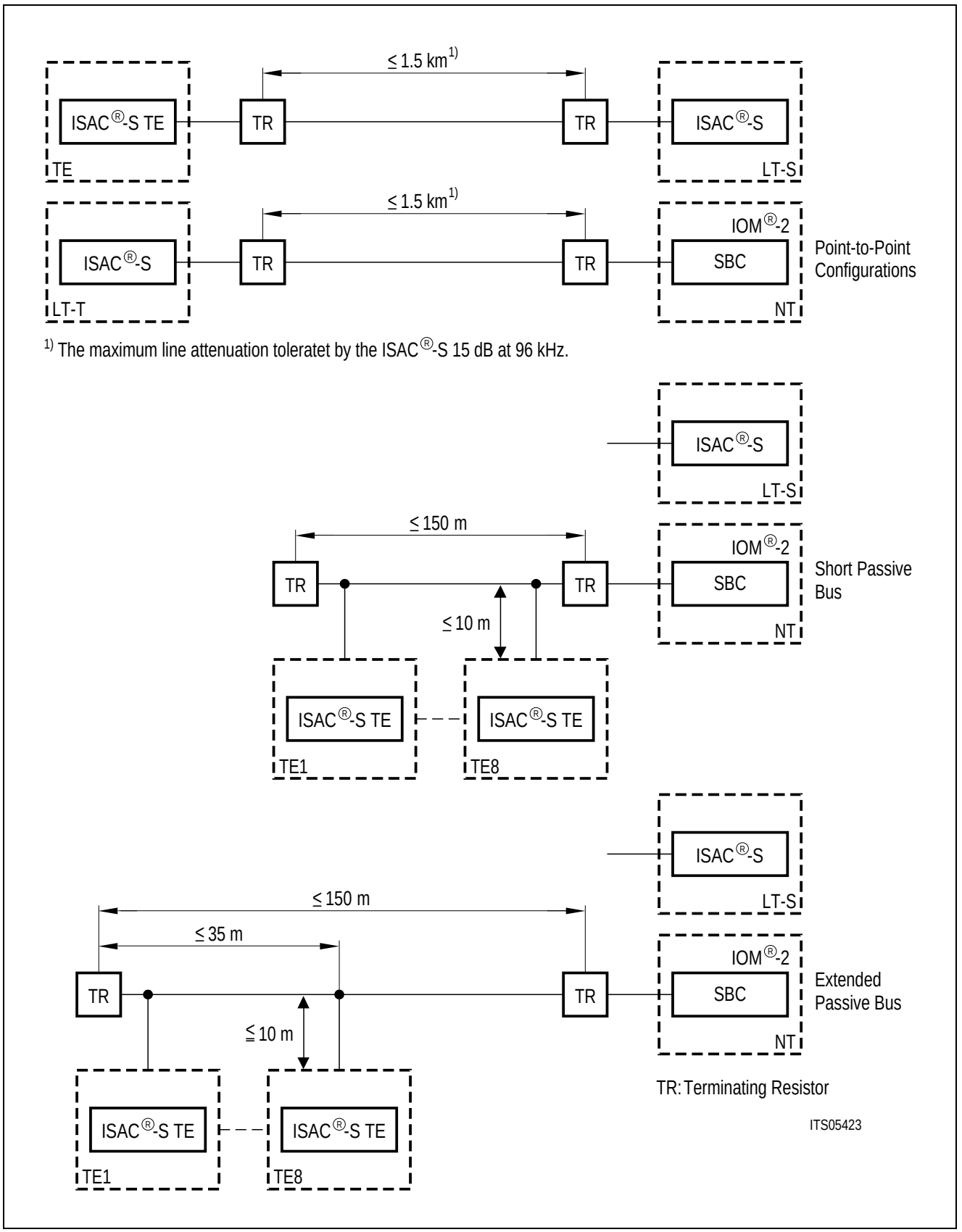


Figure 20
Wiring Configurations in User Premises

2.4.1 S/T Interface

According to CCITT recommendation I.430 pseudo-ternary encoding with 100% pulse width is used on the S/T interface. A logical "1" corresponds to a neutral level (no current), whereas logical "0" ’s are encoded as alternating positive and negative pulses. An example is shown in **figure 21**.

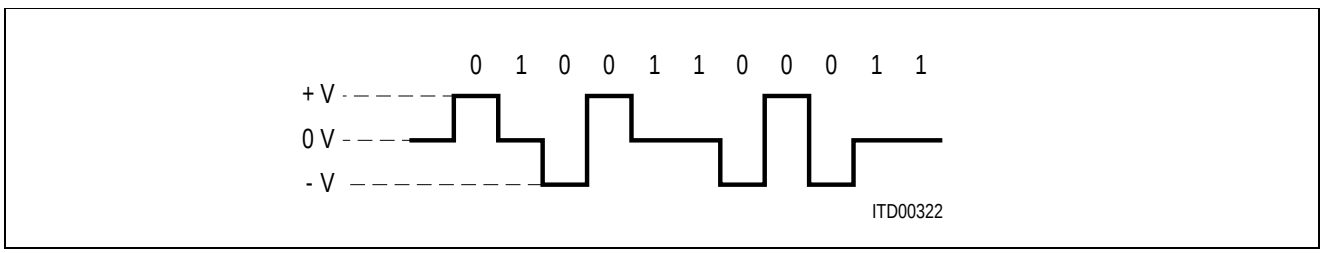


Figure 21
S/T-Interface Line Code

One frame consists of 48 bits, at a nominal bit rate of 192 kbit/s. Thus each frame carries two octets of B1, two octets of B2, and four D bits, according to the B1+B2+D structure defined for the ISDN-basic access (total useful data rate: 144 kbit/s). Frame begin is marked using a code violation (no mark inversion). The frame structures (from network to subscriber, and subscriber to network) are shown in **figure 22**.

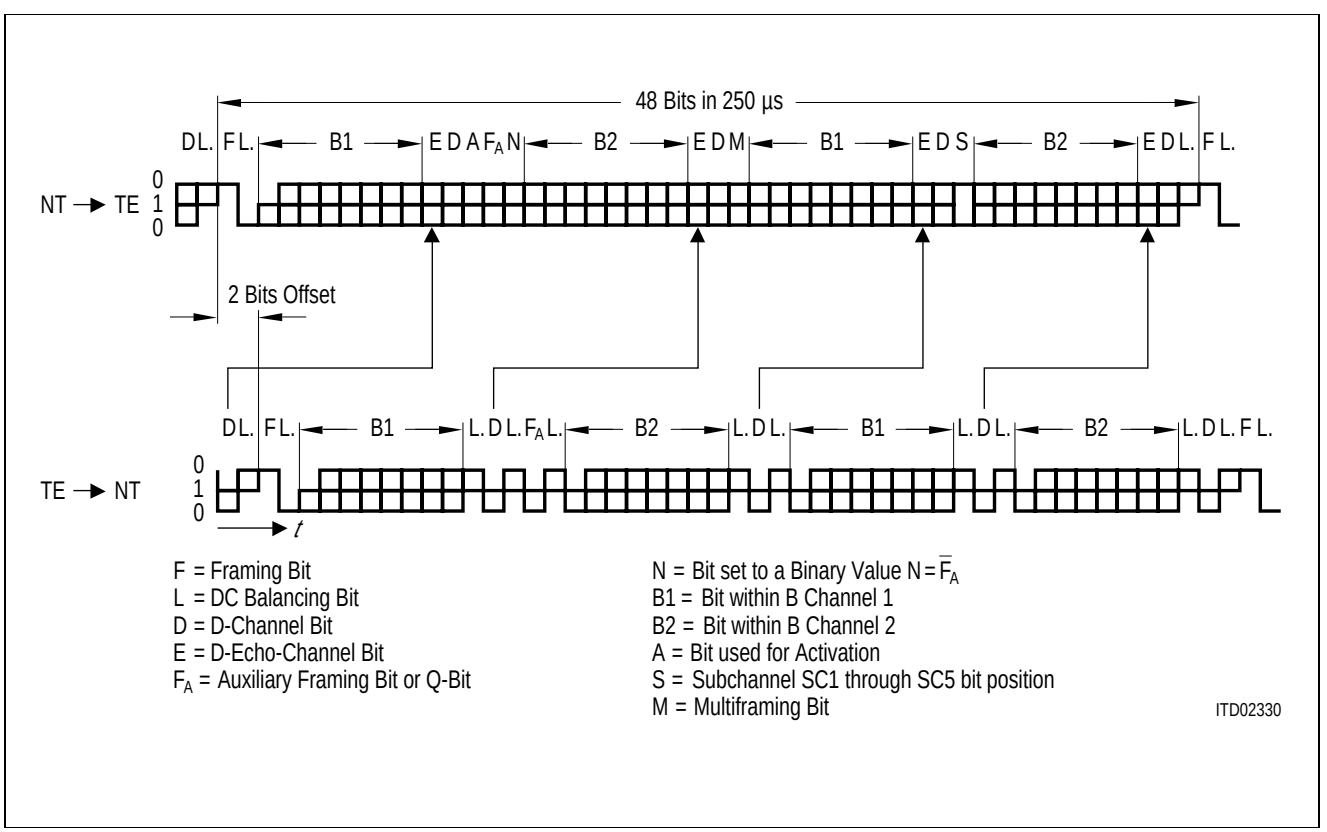


Figure 22
Frame Structure at Reference Points S and T (CCITT I.430)

Note: Dots demarcate those parts of the frame that are independently DC-balanced.

2.4.2 Analog Functions

For both receive and transmit direction, a 2:1 transformer is used to connect the ISAC-S TE transceiver to the 4 wire S/T interface. The corrections are shown in **figure 23**.

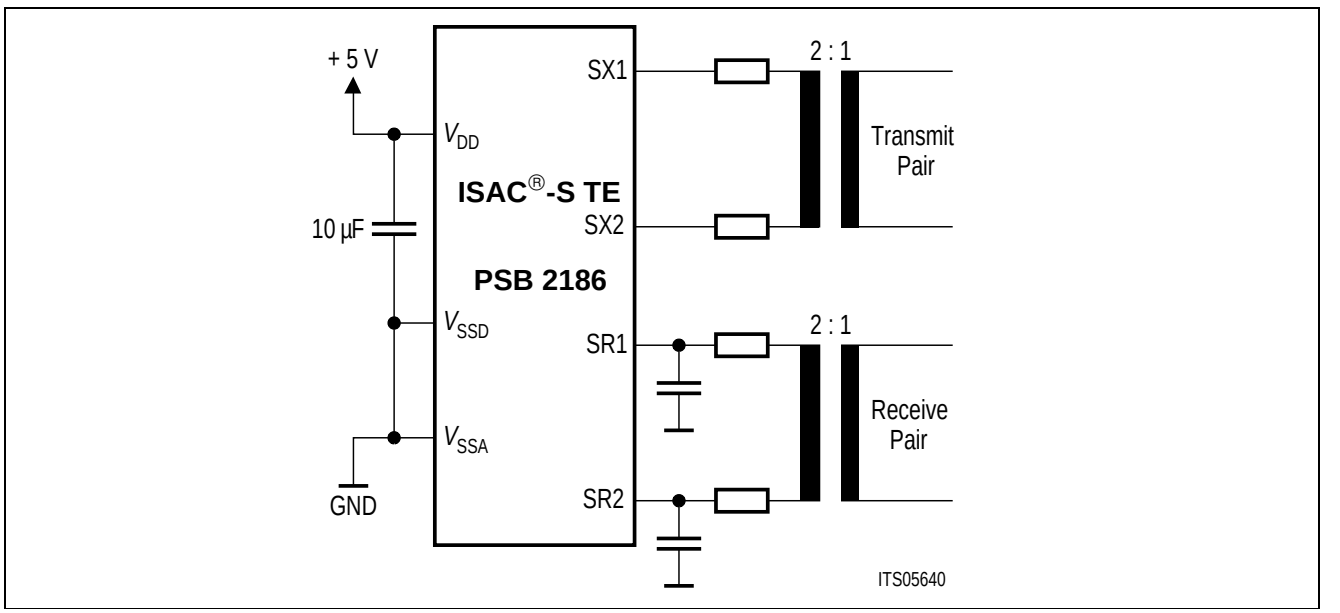


Figure 23
ISAC-S TE External S-Interface Circuitry

The full-bauded pseudo-ternary pulse shaping is achieved with the integrated transmitter which is realized as a current limited voltage source. A voltage of 2.1 V is delivered between SX1-SX2, which yields a current of 7.5 mA over 280 Ω.

The receiver is designed as a threshold detector with adaptively switched threshold levels. Pin SR1 delivers 2.5 V as an output, which is the virtual ground of the input signal on pin SR2.

The external transformer of ratio 2:1 is needed in both receive and transmit direction to provide for isolation and transform voltage levels according to CCITT recommendations.

The equivalent circuits of the integrated receiver and transmitter are shown in **figure 24**.

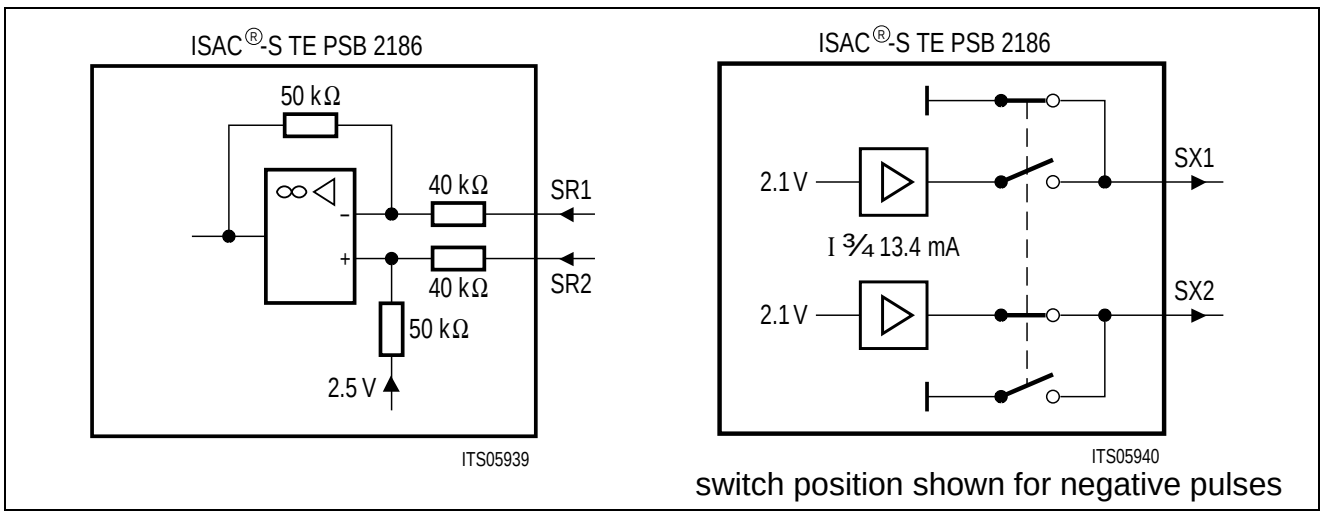


Figure 24
Equivalent Internal Circuits of Receiver and Transmitter Stages

Symmetrical S-Bus Receiver

The S-bus receiver of the PSB 2186 is a symmetrical one. This results in a simplification of the external circuitry and PCB layout to meet the I.430 receiver input impedance specification.

2.4.3 S/T-Interface Circuitry

In order to comply to the physical requirements of CCITT recommendation I.430 and considering the national requirements concerning overvoltage protection and electromagnetic compatibility (EMC), the ISAC-S TE needs some additional circuitry.
Useful hints how to design such interface circuitry are contained on the Application Note “S/T-interface circuitry using the PEB 2080 SBC or PEB 2085 ISAC-S (12/89)”.

The transmitter of the PSB 2186 ISAC-S TE is identical to that of both the PEB 2080 SBC and PEB 2085/ISAC-S, hence, the line interface circuitry should be the same (**figure 25**). The external resistors (20 … 40 Ω) are required in order to adjust the output voltage to the pulse mask (nominal 750 mV according to CCITT I.430, to be tested with the command “SSZ”) on the one hand and in order to meet the output impedance of minimum 20 Ω (transmission of a binary zero according to CCITT I.430, to be tested with the command “SCZ“) on the other hand.

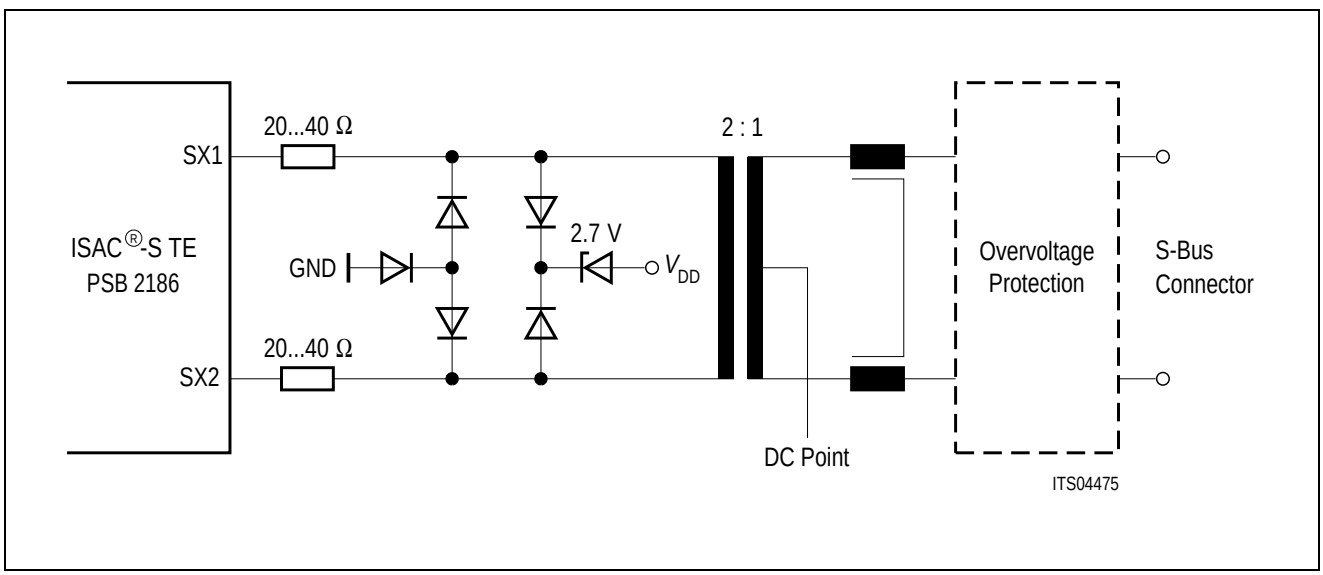


Figure 25
External Transmitter Circuitry

The receiver of the PSB 2186 ISAC-S TE is symmetrical as opposed to both PEB 2080 SBC and PEB 2085 ISAC-S. Thus two resistors of 10 kΩ must be placed in series to the receiver inputs.

In order to protect the ISAC-S inputs and comply to impedance requirements performed without power supply (96-kHz test), the 10 kΩ tester is split-up.

A 1.8 kΩ resistor protects the device inputs, while the 8.2 kΩ resistors limit the maximum current in impedance tests.

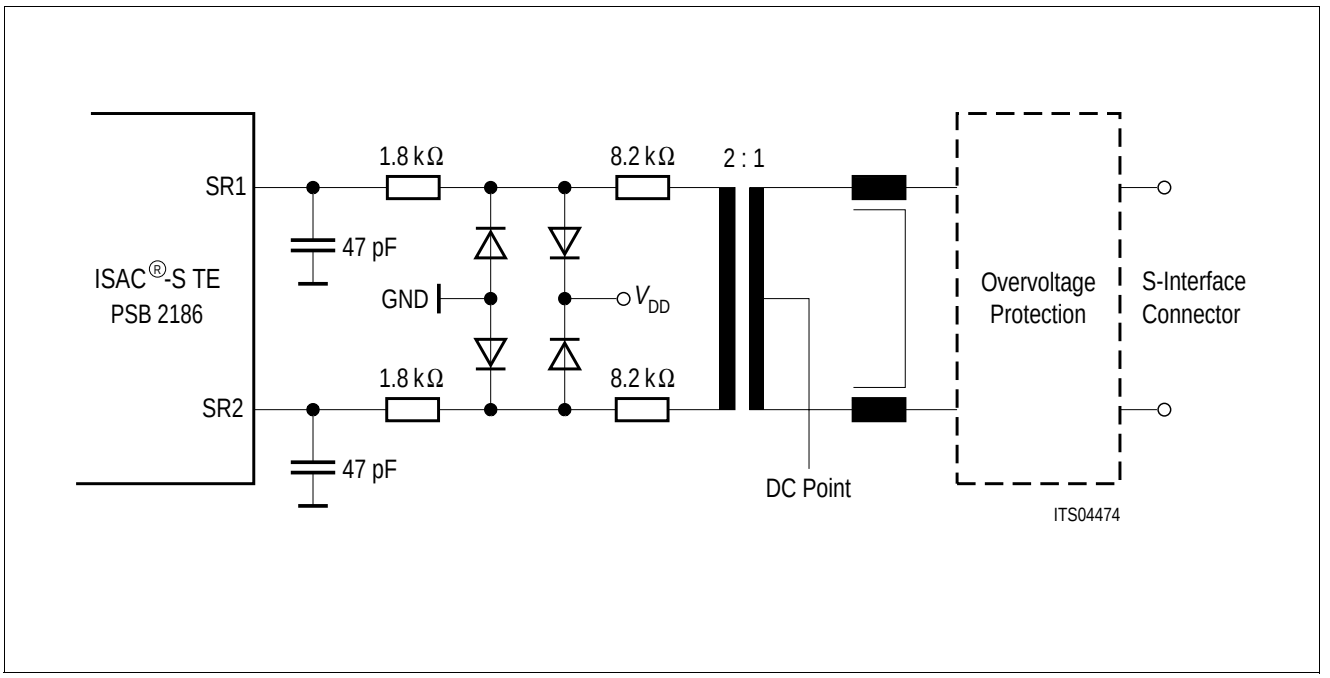


Figure 26
External Receiver Circuitry

2.4.4 S/T Interface Pre-Filter Compensation

To compensate for the extra delay introduced into the received signal by a filter, the sampling of the receive signal can be delayed by programming bits TEM and PFS in the ADF1 register as shown in **table 3**. Note that setting TEM to "1" and PFS to "0" has the effect of completely disabling layer-1 functions, for test purposes (**see section 2.6**).

Table 3
TEM/PFS Function Table

TEM	PFS	Effect
0	0	No pre-filter (0 delay)
0	1	Pre-filter delay compensation 520 ns
1	1	Pre-filter delay compensation 910 ns
1	0	Test mode (layer-1 disabled)

This delay compensation might be necessary in order to comply with the "total phase deviation input to output" requirement of CCITT recommendation I.430 which specifies a phase deviation in the range of – 7% to + 15% of a bit period.

2.4.5 Receiver Functions

2.4.5.1 Receive Signal Oversampling

In order to additionally reduce the bit error rate in severe conditions, the ISAC-S TE performs oversampling of the received signal and uses majority decision logic.

As illustrated in **figure 27**, each received bit is sampled 29 times at 7.68-MHz clock intervals inside the estimated bit window. The samples obtained are compared against a threshold VTR1 or VTR2 (**see section: Adaptive Receiver Characteristics**).

If at least 16 samples have an amplitude exceeding the selected threshold, a logical "0" is considered to be detected, otherwise a logical "1" (no signal) is considered detected.

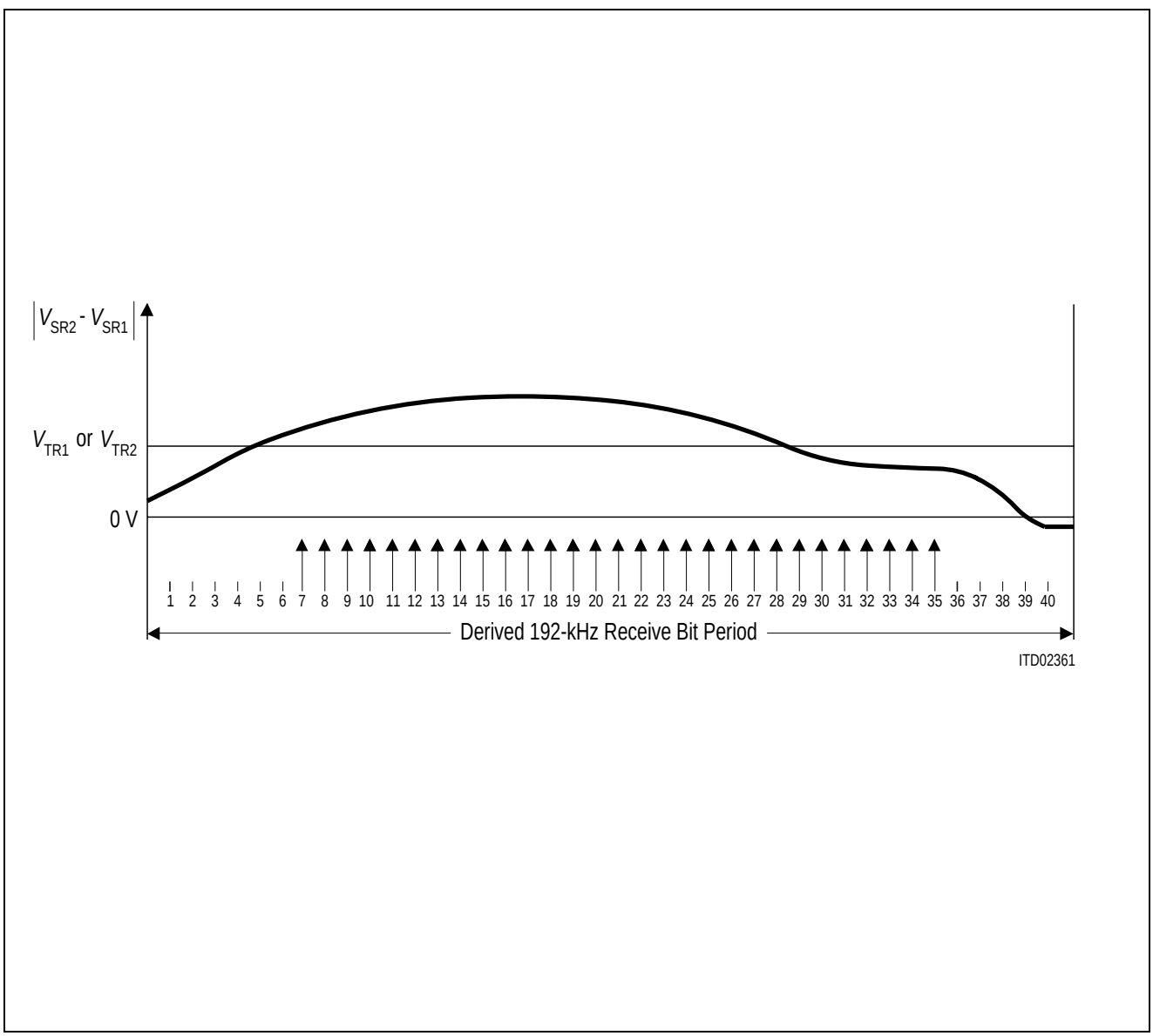


Figure 27
S/T-Receive Signal Oversampling

2.4.5.2 Adaptive Receiver Characteristics

The integrated receiver uses an adaptively switched threshold detector. The detector controls the switching of the receiver between two sensitivity levels. The hysteresis characteristics of the receiver are shown in **figure 28**.

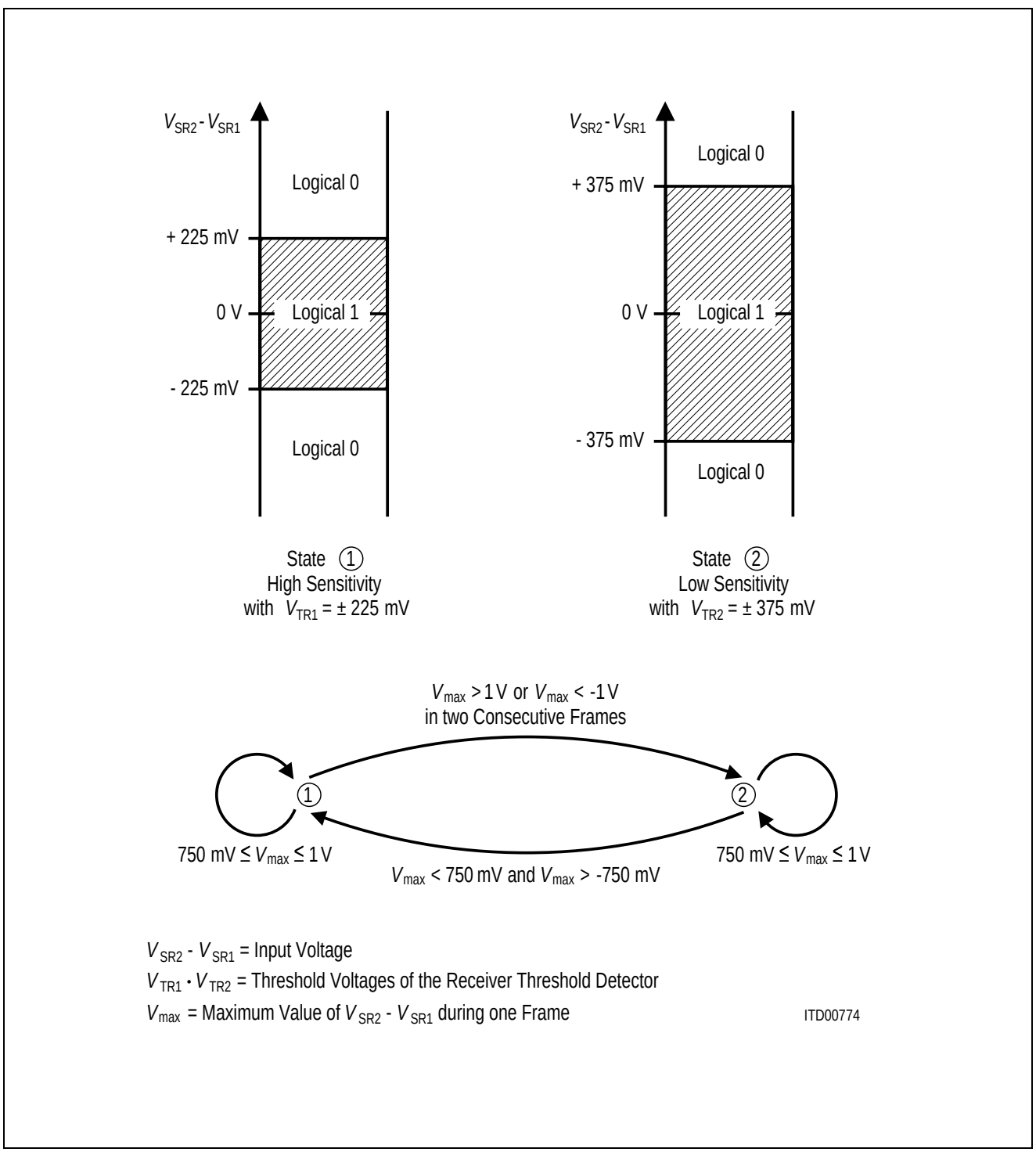


Figure 28
Switching of the Receiver between High Sensitivity and Low Sensitivity

2.4.5.3 Level Detection Power Down

In power down state, (**see chapter 3.3.1)** only an analog level detector is active. All clocks, including the IOM interface, are stopped. The data lines are "high", whereas the clocks are "low".

An activation initiated from the exchange side (Info 2 on S bus detected) will have the consequence that a clock signal is provided automatically.

From the terminal side an activation must be started by setting and resetting the SPU bit in the SPCR register (**see chapter 4**).

2.4.6 Timing Recovery

The transmit and receive bit clocks are derived, with the help of the DPLL, from the S-interface receive data stream. The received signal is sampled several times inside the derived receive clock period, and a majority logic is used to additionally reduce bit error rate in severe conditions (**see chapter 2.4.5**). The transmit frame is shifted by two bits with respect to the received frame.

The output clocks (DCL, FSC1 etc.) are synchronous to the S-interface timing.

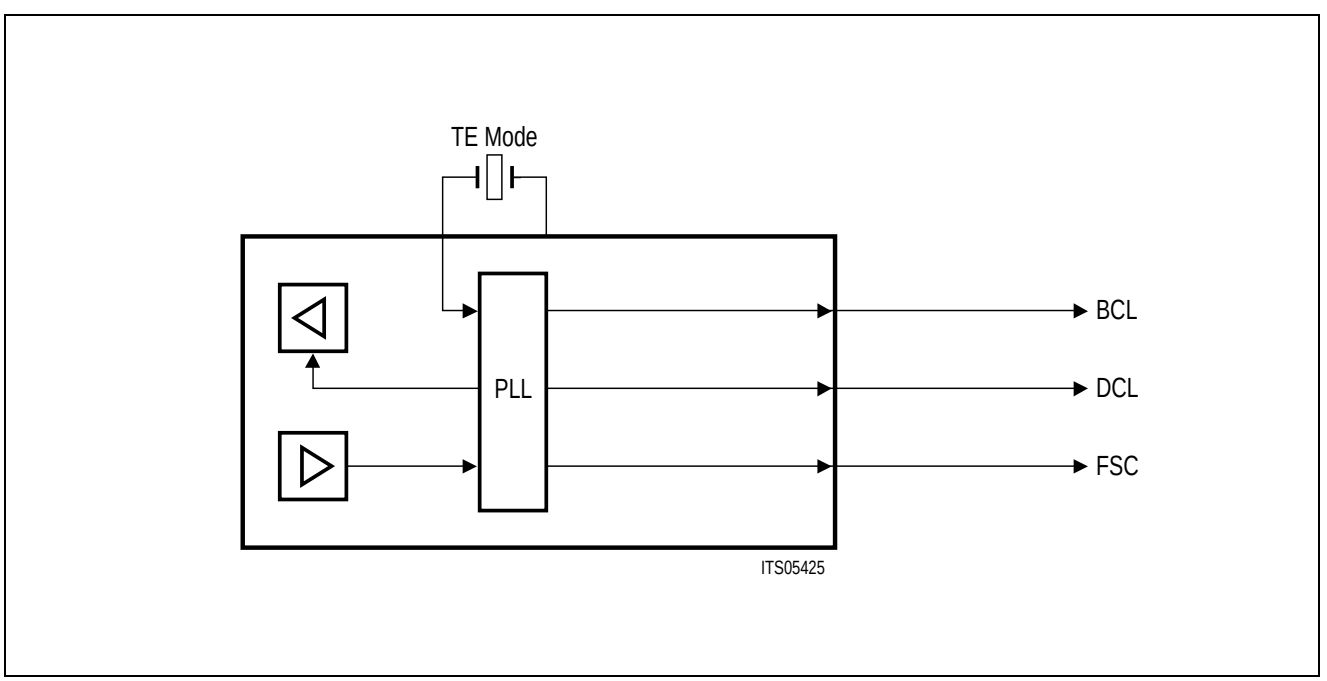


Figure 29
Clock System of the ISAC®-S TE in TE Mode

2.4.7 Activation/Deactivation

An incorporated finite state machine controls ISDN layer-1 activation/deactivation according to CCITT (**see chapter 3.4**).

Loss of Synchronization / Resynchronization

The following section describes the behaviour of the PSB 2186 in respect to the CTS test procedures for frame alignment.

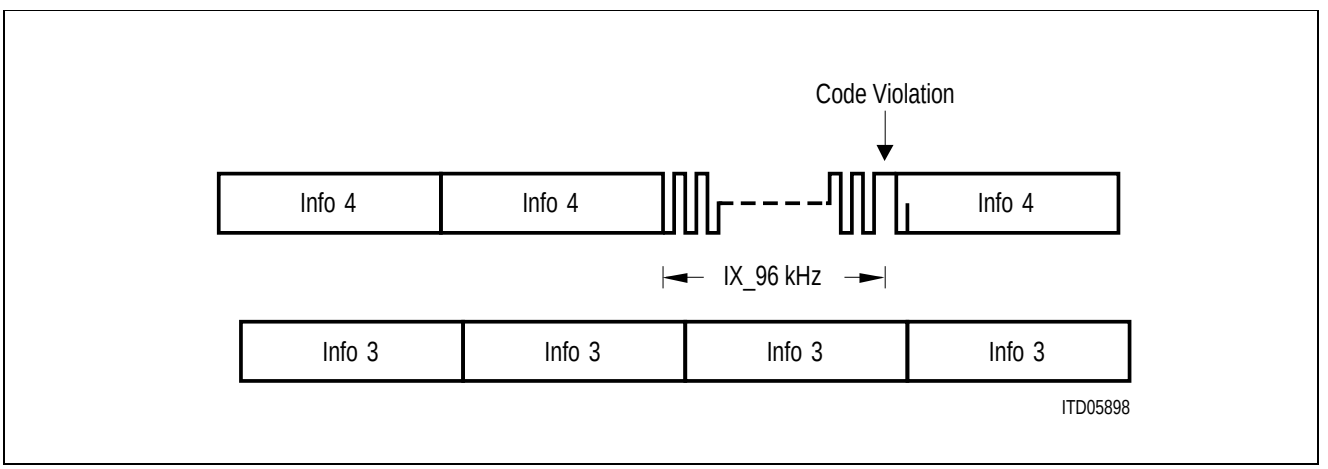
Setting of the ISAC-S TE

The ISAC-S TE needs to be programmed for multiframe operation with the Q-bits set to ’1’.

STAR2: MULT = 0
SQXR:SQX1-4 = 1111B (xFH)

2.4.7.1 FAinfA_1fr

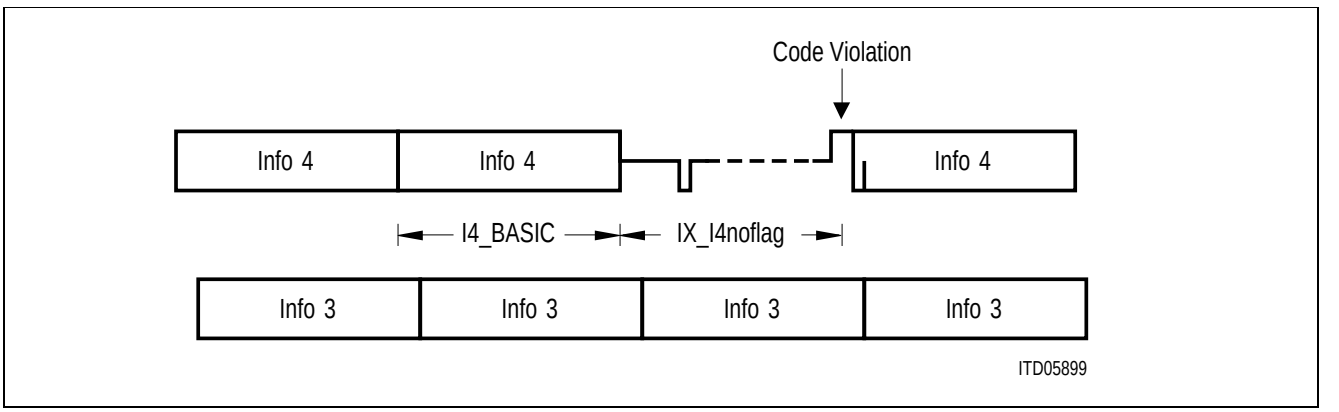
This test checks if no loss of frame alignment occurs upon a receipt of one bad frame. The pattern for the bad frame is defined as IX_96 kHz. This pattern was revised so that a code violation is generated at the begin of the next info 4 frame.



Device	Settings	Result	Comments
PSB 2186 V1.1	none	Pass	

2.4.7.2 FAinfB_1fr

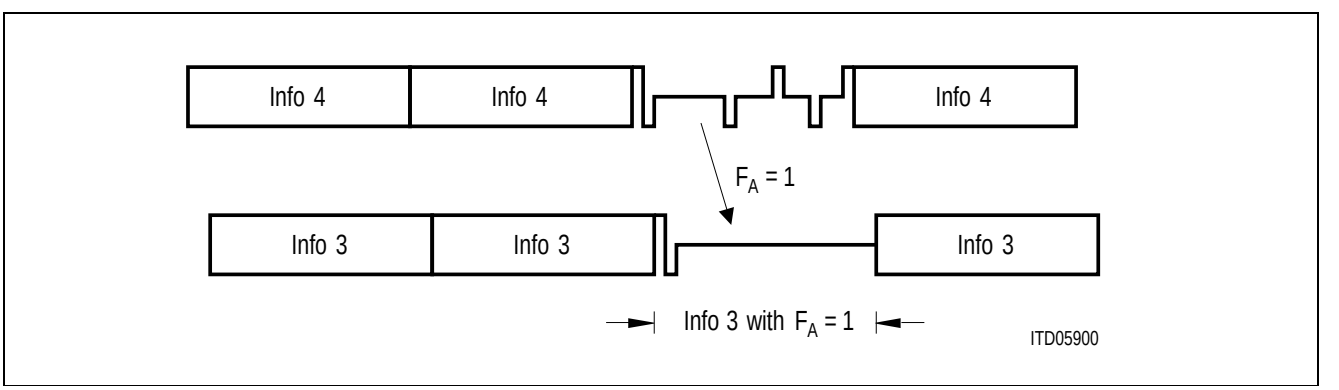
This test uses a frame which has no framing and balancing bit.



Device	Settings	Result	Comments
PSB 2186 V1.1	none	Pass	

2.4.7.3 FAinfD_1fr

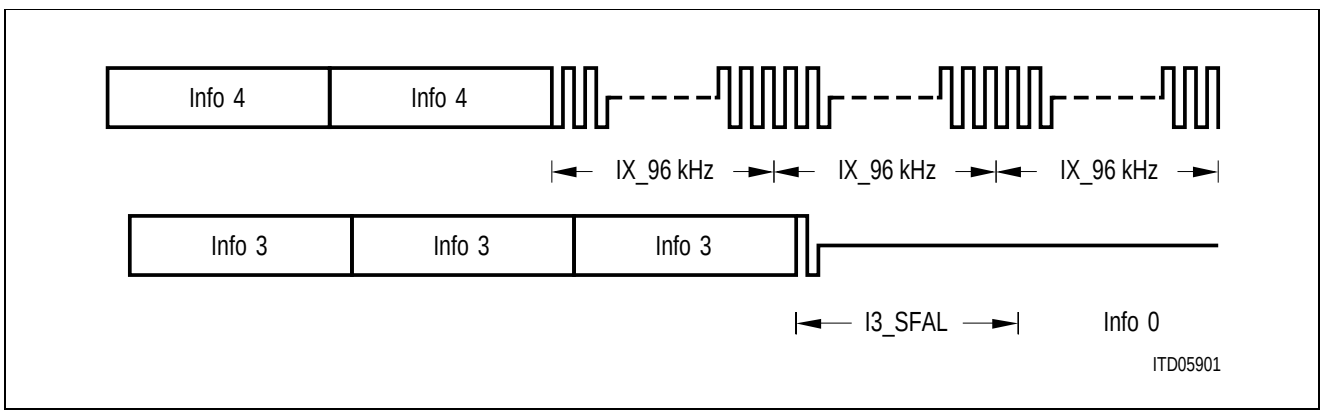
This test uses a frame which remains at binary '1' until the first code violation in bit 16. Since it is specified, that a terminal should mirror the received F_A-bit in the transmitted F_A-bit, a frame is generated by the IUT which will not generate a second code violation. The pattern for a correct i3_BASIC frame states that the F_A-bit may have any value.



Device	Settings	Result	Comments
PSB 2186 V1.1	none	Pass	

2.4.7.4 FAinfA_kfr

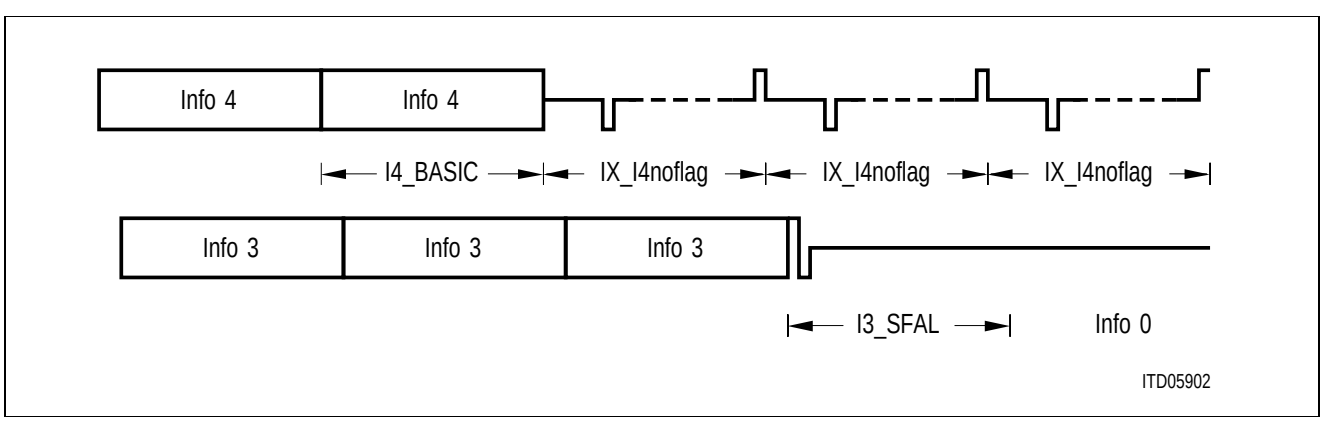
This test uses a number of IX_96 kHz frames to check the loss of synchronization.



Device	Settings	Result	Comments
PSB 2186 V1.1	n = 2	Pass	

2.4.7.5 FAinfB_kfr

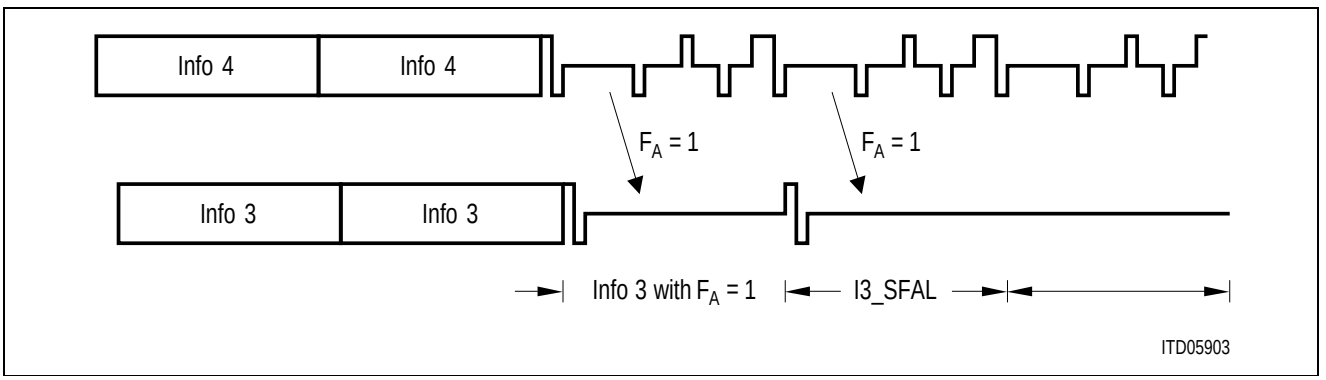
This test uses a number of IX_I4noflag frames to check the loss of synchronization.



Device	Settings	Result	Comments
PSB 2186 V1.1	n = 2	Pass	

2.4.7.6 FAinfD_kfr

This test uses a number of IX_I4voil16 frames to check the loss of synchronization. The first Info 3 frame with the F_A-bit set to one looks like a i3_SFAL frame but it is a correct info 3 frame since the receiver stays synchronous (**see FAinfD_1fr**).

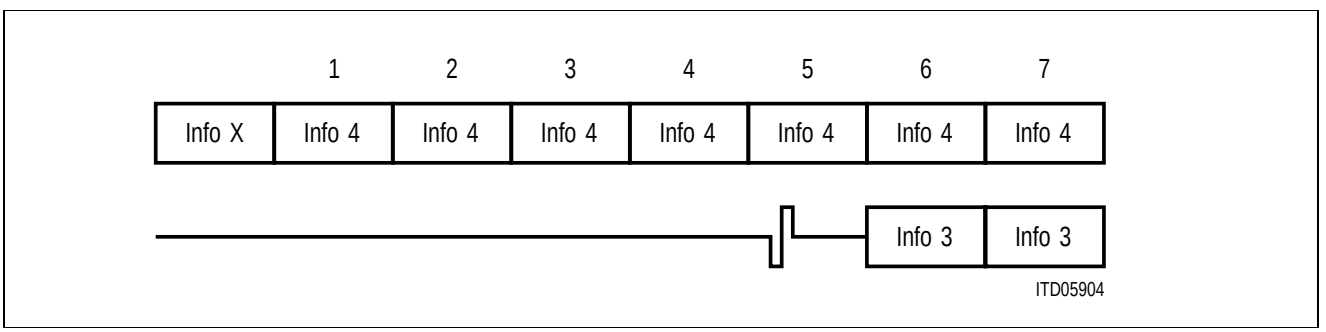


Device	Settings	Result	Comments
PSB 2186 V1.1	n = 2	Pass	

2.4.7.7 FAregain

This test uses I4_BASIC frames to regain the frame alignment. The protocol tester evaluates the difference between sending the first info 4 frame until a complete info 3 frame has been received. This period is considered as 'm+1'. 'm' must be specified before the test is started.

The PSB 2186 achieves synchronization after 5 or 6 frames. The actual value depends on internal timing conditions which can not be influenced from extern. This is a result of changes that were made to handle the iXvoil16 test case correctly. The info 4 pattern generates the second code violation at the position of the F_A-bit. Around that bit position, the state machine changes its states. As a result of that overlap, the info 3 frame is transmitted after 5 frames or one frame later.



Device	Settings	Result	Comments
PSB 2186 V1.1	m = 5 or 6	Pass	

2.4.8 D-Channel Access

The D channel is submitted to the D-channel access procedure according to CCITT recommendation I.430.

The D-channel access procedure according to CCITT I.430 including priority management is fully implemented in the ISAC-S:

If collision detection is programmed (MODE:DIM2-0), a collision is detected if either an echo bit of "0" is recognized and a D bit of "1" was generated, or an echo bit of "1" is recognized and a D bit of "0" was generated. When this occurs, D-channel transmission is immediately stopped, and the echo channel is monitored to enable a subsequent D-channel access to be attempted.

Stop/Go Bit

As the collision resolution is performed by the layer-1 part of the device, an information about the D-channel status ("ready" or "busy") must be sent back to the layer-2 part to control HDLC transmission. For this goal a Stop/Go (S/G) bit is transmitted over the IOM interface to the layer-2 device.

The S/G bit is transmitted in bit 90 of an IOM-2 frame (12-byte structure) (**see figure 19**).

A logical "1" of the S/G bit indicates a collision on the S bus. By sending the S/G bit a logical "0" to the layer-2 controller in anticipation of the S bus D channel "ready"-state, the first valid 0 bits will emerge from the layer-1 part at exactly that moment an access is becoming possible.

Selection of D-Channel Access Mode

For proper operation of the D-channel access procedure, the ISAC-S TE must be programmed via the MODE (**see chapter 4.1.7**) register to evaluate the stop/go bit. This is achieved by setting MODE:DIM2-0 to 001 or 011.

Selection of the Priority Class

The priority class (priority 8 or priority 10) is selected by transferring the appropriate activation command via the Command/Indicate (C/I) channel of the IOM interface to the layer-1 controller. If the activation of the S interface is initiated by a TE, the priority class is selected implicitly by the choice of the activation command. If the S-Interface is activated from the NT, an activation command selecting the desired priority class should be programmed at the TE on reception of the activation indication (AI8). In the activated state, the priority class may be changed whenever required simply by programming the respective activation request command (AR8 or AR10). The following table summarizes the C/I codes used for setting the priority classes:

Table 4
Priority Commands/Indications

Command (upstream)	Abbr.	Code	Remarks
Activate request, set priority 8	AR8	1000	Activation command: Set D-channel priority to 8
Activate request, set priority 10	AR10	1001	Activation command: Set D-channel priority to 10
Indication (downstream)	**Abbr.**	**Code**	**Remarks**
Activate indication with priority class 8	AI8	1100	Info 4 received: D-channel priority is 8 or 9
Activate indication with priority class 10	AI10	1101	Info 4 received: D-channel priority is 10 or 11

2.4.9 S- and Q-Channel Access

Access to the received/transmitted S- or Q channel is provided via registers. As specified by CCITT I.430, the Q bit is transmitted from TE to NT in the position normally occupied by the auxiliary framing bit (F_A) in one frame out of 5, whereas the S bit is transmitted from NT to TE in a spare bit, **see figure 22**.

The functions provided by the ISAC-S are:

– Synchronization to the received 20 frame multiframe by means of the received M bit pattern. Synchronism is achieved when the M bit has been correctly received during 20 consecutive frames starting from frame number 1 (**table 5**).

– When synchronism is achieved, the four received S bits in frames 1, 6, 11 and 16 are stored as SQR1 to SQR4 in the SQRR register if the complete M bit multiframe pattern was correctly received in the corresponding multiframe. A change in any of the received four bits (SQR1, 2, 3 or 4) is indicated by an interrupt (CISQ in ISTA and SQC in CIR0).

– When an M bit is observed to have a value different from that expected, the synchronism is considered lost. The SQR bits are not updated until synchronism is regained. The synchronization state is constantly indicated by the SYN bit in the SQRR register.

– When synchronism with the received multiframe is achieved, the four bits SQX1 to SQX4 stored in the SQXR register are transmitted as the four Q bits (F_A-bit position) in frames 1, 6, 11 and 16 respectively (starting from frame number one). Otherwise the bit transmitted is a mirror of the received F_A-bit. At loss of synchronism (mismatch in M bit) the mirroring is resumed starting with the next F_A-bit.

– The S/T multiframe synchronization can be disabled in the STAR2 register (MULT bit).

Table 5
S- and Q-Bit Position Identification and Multiframe Structure

S- and Q-Channel Structure

Frame Number	NT-to-TE F_A-Bit Position	NT-to-TE M Bit	NT-to-TE S Bit	TE-to-NT F_A-Bit Position
1	ONE	ONE	S1	Q1
2	ZERO	ZERO	ZERO	ZERO
3	ZERO	ZERO	ZERO	ZERO
4	ZERO	ZERO	ZERO	ZERO
5	ZERO	ZERO	ZERO	ZERO
6	ONE	ZERO	S2	Q2
7	ZERO	ZERO	ZERO	ZERO
8	ZERO	ZERO	ZERO	ZERO
9	ZERO	ZERO	ZERO	ZERO
10	ZERO	ZERO	ZERO	ZERO
11	ONE	ZERO	S3	Q3
12	ZERO	ZERO	ZERO	ZERO
13	ZERO	ZERO	ZERO	ZERO
14	ZERO	ZERO	ZERO	ZERO
15	ZERO	ZERO	ZERO	ZERO
16	ONE	ZERO	S4	Q4
17	ZERO	ZERO	ZERO	ZERO
18	ZERO	ZERO	ZERO	ZERO
19	ZERO	ZERO	ZERO	ZERO
20	ZERO	ZERO	ZERO	ZERO
1	ONE	ONE	S1	Q1
2	ZERO	ZERO	ZERO	ZERO
etc.				

2.5 Terminal Specific Functions

Watchdog and External Awake

In addition to the ISAC-S TE standard functions supporting the ISDN-basic access, the ISAC-S TE contains optional functions, useful in various terminal configurations.

The terminal specific functions are enabled by setting bit TSF (STCR register) to "1". This has two effects:

The EAW line is defined as an external awake input;

Second, the interrupts SAW and WOV (EXIR register) are enabled:

- SAW (Subscriber Awake) generated by a falling edge on the EAW line
- WOV (Watchdog timer OVerflow) generated by the watchdog timer. This occurs when the processor fails to write two consecutive bit patterns in ADF1:

ADF1	WTC1	WTC2	

Watchdog Timer Control 1, 0.

The WTC1 and WTC2 bits have to be successively written in the following manner within 128 ms:

	WTC1	**WTC2**
1. 2.	1 0	0 1

As a result the watchdog timer is reset and restarted. Otherwise a WOV is generated.

Deactivating the terminal specific functions is only possible with a hardware reset.

Having enabled the terminal specific functions via TSF=1, the user can make the ISAC-S TE generate a reset signal by programming the Reset Source Select **RSS** bit (CIXR/CIX0 register), as follows:

0 →A reset signal is generated as a result of

– a falling edge on the EAW line (subscriber awake)

– a C/I code change (exchange awake)

A falling edge on the EAW line also forces the IDP1 line of the IOM interface to zero. The consequence of this is that the IOM interface and the ISAC-S TE leaves the power-down state.

A corresponding interrupt status (CISQ or SAW) is also generated.

1 →A reset signal is generated as a result of the expiration of the watchdog timer (indicated by the WOV interrupt status).

Note that the watchdog timer is not running when the ISAC-S TE is in the power-down state (IOM not clocked).

Note: Bit RSS has a significance only if terminal specific functions are activated (TSF=1).

The RSS bit should be set to "1" by the user when the ISAC-S TE is in power-up to prevent an edge on the EAW line or a change in the C/I code from generating a reset pulse.

Switching RSS from 0 to 1 or from 1 to 0 resets the watchdog timer.

The reset pulse generated by the ISAC-S TE (output via RST pin) has a pulse width of:

– 125 µs when generated by the watchdog timer
– 16 ms when generated by EAW line or C/I-code change.

2.6 Test Functions

The ISAC-S TE provides several test and diagnostic functions which can be grouped as follows:

- digital loop via TLP (Test Loop, SPCR register) command bit: IDP1 is internally connected with IDP0, output from layer 1 (S/T) on IDP0 is ignored; this is used for testing ISAC-S TE functionality excluding layer 1;
- test of layer-2 functions while disabling all layer-1 functions and pins associated with them (including clocking, in TE mode), via bit TEM (Test Mode in ADF1 register); the ISAC-S TE is then fully compatible to the ICC (PEB 2070) seen at the IOM interface.
- loop at the analog end of the S interface;

Test loop 3 is activated with the C/I-channel command Activate Request Loop (ARL). An S interface is not required since INFO3 is looped back to the receiver. When the receiver has synchronized itself to this signal, the message "Test Indication" (or "Awake Test Indication") is delivered in the C/I channel. No signal is transmitted over the S interface.

In the test loop mode the S-Interface awake detector is enabled i.e. if a level is detected (e.g. Info 2/Info 4) this will be reported by the Awake Test Indication (ATI). The loop function is not effected by this condition and the internally generated 192-kHz line clock does not depend on the signal received at the S interface.

2.7 Layer-2 Functions for the ISDN-Basic Access

LAPD, layer 2 of the D-channel protocol (CCITT I.441) includes functions for:

– Provision of one or more data link connections on a D channel (multiple LAP). Discrimination between the data link connections is performed by means of a data link connection identifier (DLCI = SAPI + TEI)

– HDLC framing

– Application of a balanced class of procedure in point-multipoint configuration.

The simplified block diagram in **figure 30** shows the functional blocks of the ISAC-S TE which support the LAPD protocol.

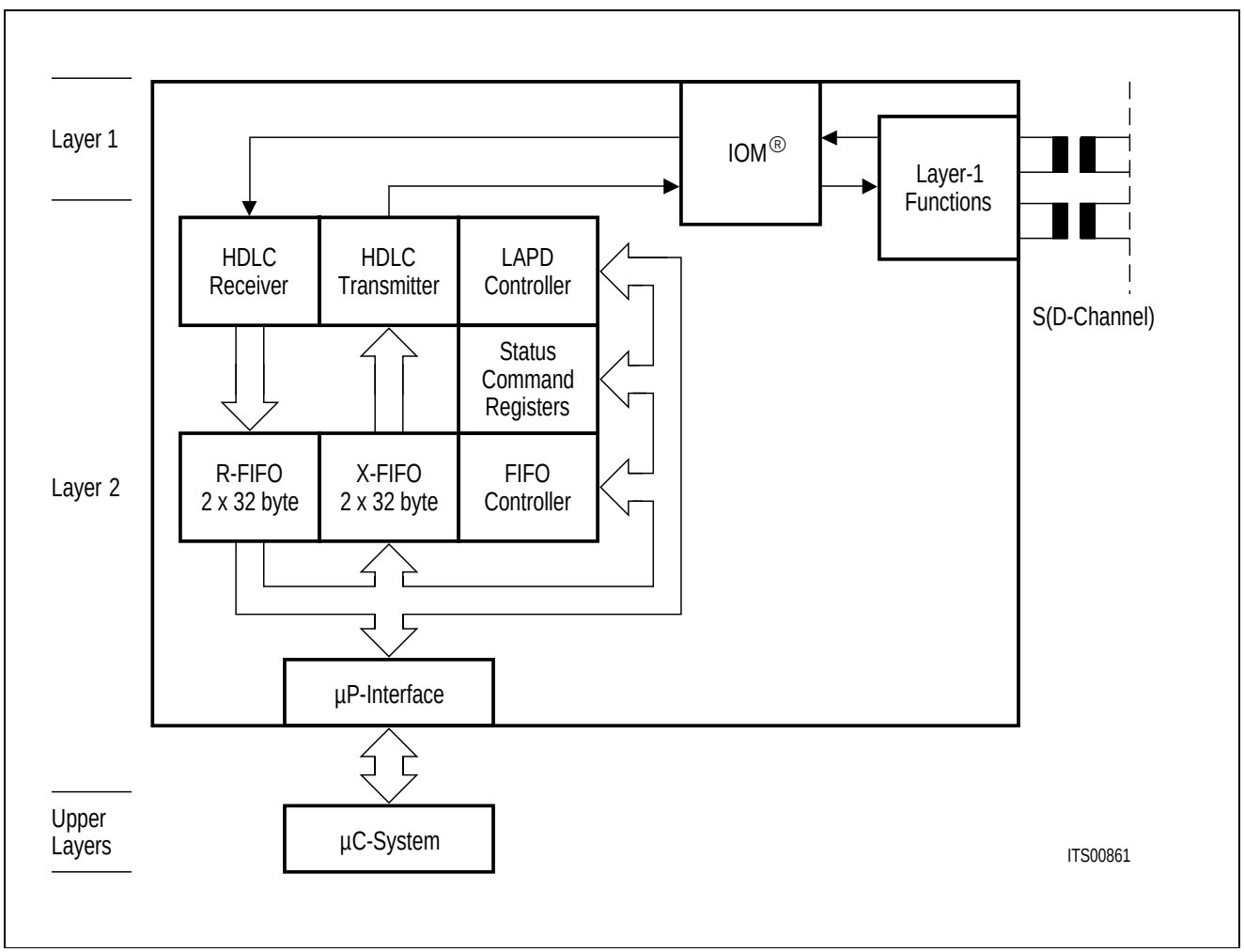


Figure 30
D-Channel Processing of the ISAC®-S TE

For the support of LAPD the ISAC-S TE contains an HDLC transceiver which is responsible for flag generation/recognition, bit stuffing mechanism, CRC check and address recognition.

A powerful FIFO structure with two 64-byte pools for transmit and receive directions and an intelligent FIFO controller permit flexible transfer of protocol data units to and from the µC system.

2.7.1 Message Transfer Modes

The HDLC controller can be programmed to operate in various modes, which are different in the treatment of the HDLC frame in the receive direction. Thus, the receive data flow and the address recognition features can be programmed in a flexible way, which satisfies different system requirements.

In the auto mode the ISAC-S TE handles elements of procedure of the LAPD (S and I frames) according to CCITT I.441 fully autonomously.

For the address recognition the ISAC-S TE contains four programmable registers for individual SAPI and TEI values SAP1-2 and TEI1-2, plus two fixed values for "group" SAPI and TEI, SAPG and TEIG.

There are 5 different operating modes which can be set via the MODE register (addr. 22_H):

Auto-mode (MDS2, MDS1 = 00)

Characteristics:

– Full address recognition (1 or 2 bytes).

– Normal (mod 8) or extended (mod 128) control field format

– Automatic processing of numbered frames of an HDLC procedure (**see 2.7.5**)

If a 2-byte address field is selected, the high address byte is compared with the fixed value FE_H or FC_H (group address) as well as with two individually programmable values in SAP1 and SAP2 registers. According to the ISDN LAPD protocol, bit 1 of the high byte address will be interpreted as command/response bit (C/R) dependent on the setting of the CRI bit in SAP1, and will be excluded from the address comparison.

Similarly, the low address byte is compared with the fixed value FF_H (group TEI) and two compare values programmed in special registers (TEI1, TEI2). A valid address will be recognized in case the high and low byte of the address field match one of the compare values. The ISAC-S TE can be called (addressed) with the following address combinations:

– SAP1/TEI1

– SAP1/FF_H

– SAP2/TEI2

– SAP2/FF_H

– FE_H (FC_H)/TEI1

– FE_H (FC_H)/TEI2

– FE_H (FC_H)/FF_H

Only the logical connection identified through the address combination SAP1, TEI1 will be processed in the auto mode, all others are handled as in the non-auto mode. The logical connection handled in the auto-mode must have a window size 1 between transmitted and acknowledged frames. HDLC frames with address fields that do not match with any of the address combinations, are ignored by the ISAC-S TE.

In case of a 1-byte address, TEI1 and TEI2 will be used as compare registers. According to the X.25 LAPB protocol, the value in TEI1 will be interpreted as command and the value in TEI2 as response.

The control field is stored in the RHCR register and the I field in the RFIFO. Additional information is available in the RSTA.

Non-auto mode (MDS2, MDS1 = 01)

Characteristics:Full address recognition (1 or 2 bytes)
Arbitrary window sizes

All frames with valid addresses (address recognition identical to auto mode) are accepted and the bytes following the address are transferred to the µP via RHCR and RFIFO. Additional information is available in the RSTA.

Transparent mode 1 (MDS2, MDS1, MDS0 = 101)

Characteristics: TEI recognition

A comparison is performed only on the second byte after the opening flag, with TEI1, TEI2 and group TEI (FF_H). In the case of a match, the first address byte is stored in SAPR, the (first byte of the) control field in RHCR, and the rest of the frame in the RFIFO. Additional information is available in the RSTA.

Transparent mode 2 (MDS2, MDS1, MDS0 = 110)

Characteristics: no address recognition

Every received frame is stored in the RFIFO (first byte after opening flag to CRC field). Additional information can be read from the RSTA.

Transparent mode 3 (MDS2, MDS1, MDS0 = 111)

Characteristics: SAPI recognition

A comparison is performed on the first byte after the opening flag with SAP1, SAP2 and group SAPI (FE/FC_H). In the case of a match, all the following bytes are stored in RFIFO. Additional information can be read from the RSTA.

2.7.2 Protocol Operations (auto-mode)

In addition to address recognition all S and I frames are processed in hardware in the auto-mode. The following functions are performed:

– update of transmit and receive counter
– evaluation of transmit and receive counter
– processing of S commands
– flow control with RR/RNR
– response generation
– recognition of protocol errors
– transmission of S commands, if an acknowledgement is not received
– continuous status query of remote station after RNR has been received
– programmable timer/repeater functions.

The processing of frames in auto-mode is described in detail in **chapter 2.7.5**: **Documentation of the Auto-Mode.**

2.7.3 Reception of Frames

A 2 × 32 byte FIFO buffer (receive pools) is provided in the receive direction.

The control of the data transfer between the CPU and the ISAC-S TE is handled via interrupts.

There are two different interrupt indications concerned with the reception of data:

– RPF (Receive Pool Full) interrupt, indicating that a 32-byte block of data can be read from the RFIFO and the received message is not yet complete.

– RME (Receive Message End) interrupt, indicating that the reception of one message is completed, i.e. either

- one message ≤ 32 bytes, or
- the last part of a message > 32 bytes

is stored in the RFIFO.

Depending on the message transfer mode the address and control fields of received frames are processed and stored in the Receive FIFO or in special registers as depicted in **figure 32**.

The organization of the RFIFO is such that up to two short (≤ 32 bytes), successive messages, with all additional information can be stored. The contents of the RFIFO would be, for example, as shown in **figure 31.**

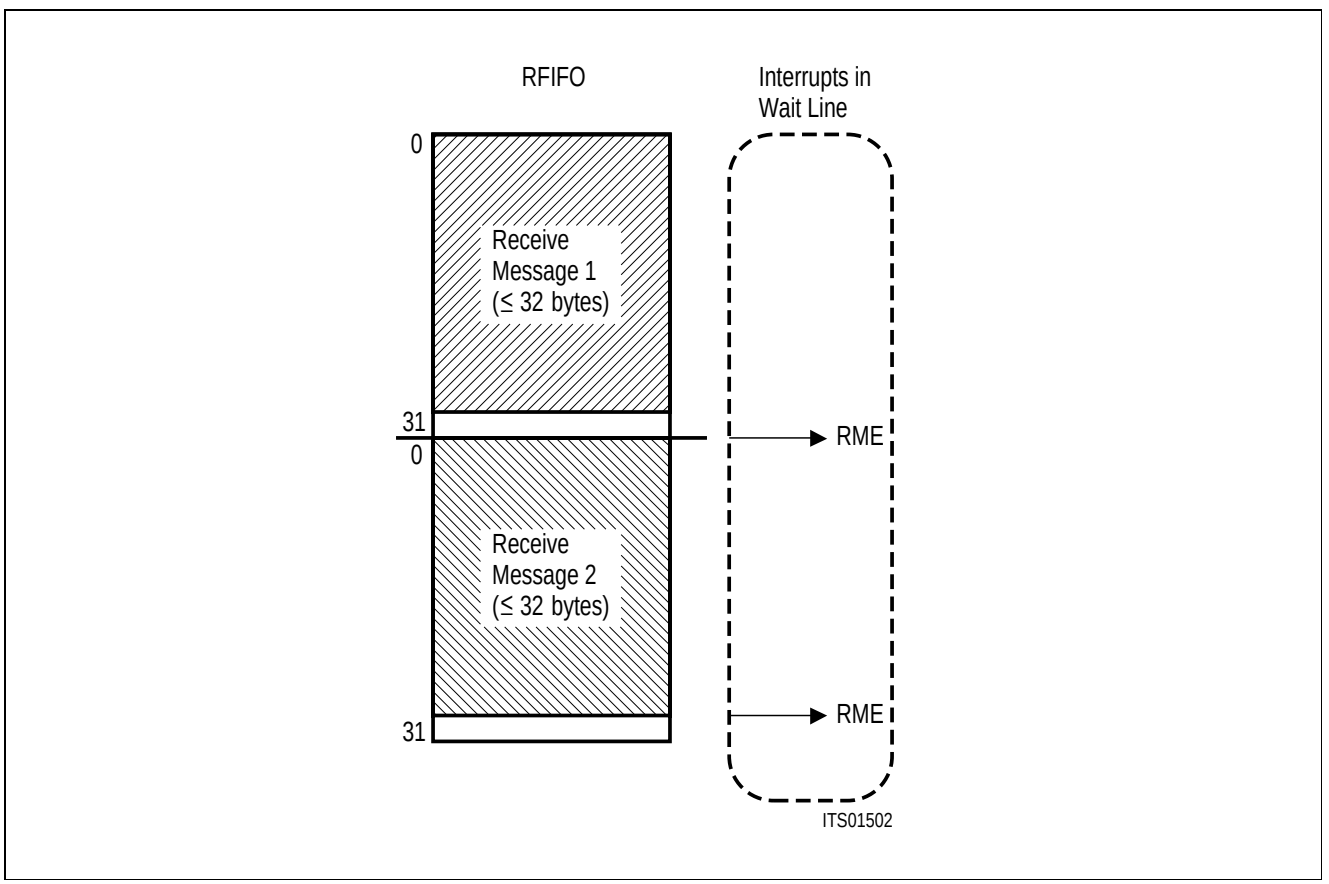


Figure 31
Contents of RFIFO (short message)

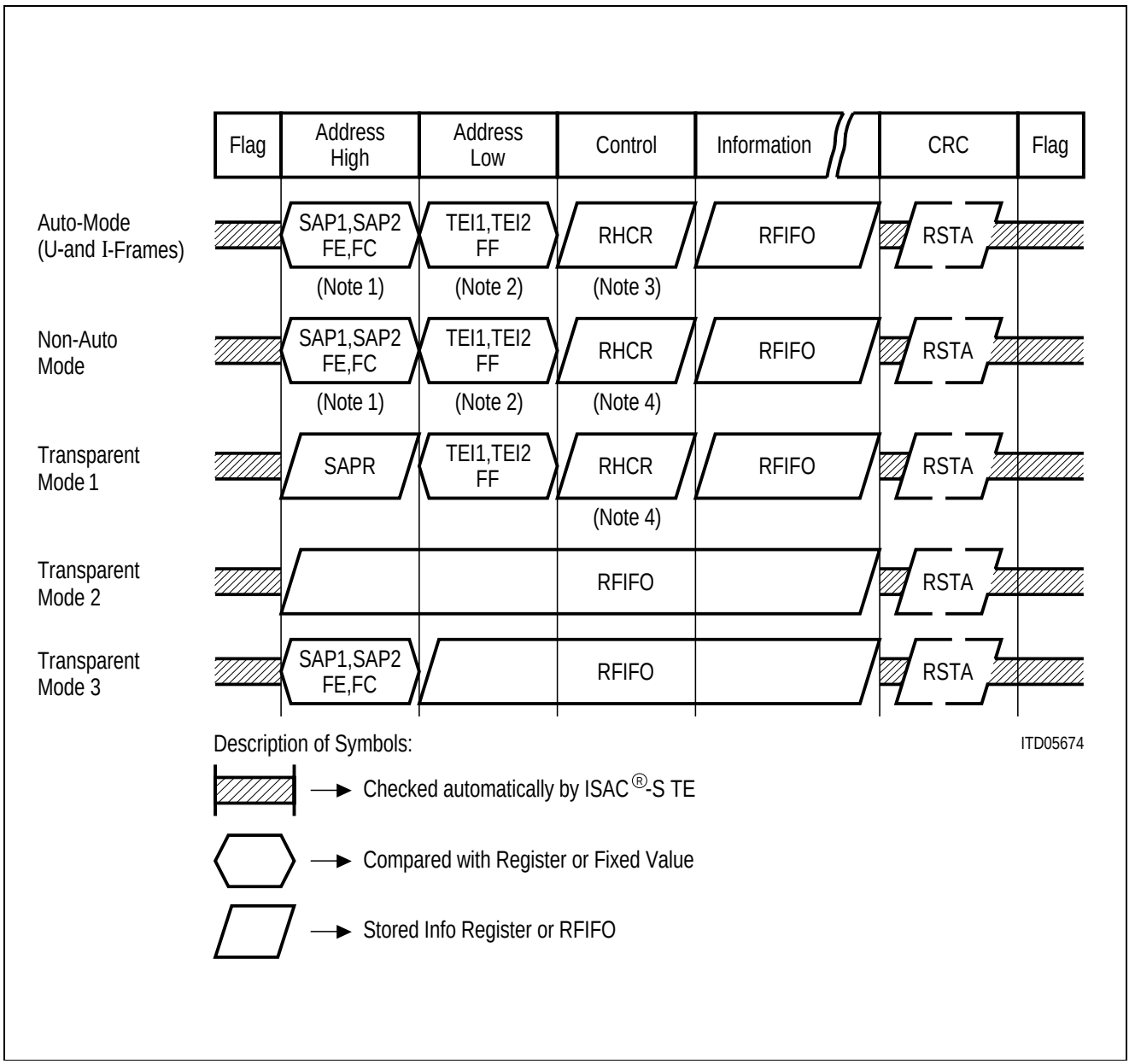


Figure 32
Receive Data Flow

Note 1 Only if a 2-byte address field is defined (MDS0 = 1 in MODE register).

Note 2 Comparison with Group TEI (FF_H) is only made if a 2-byte address field is defined (MDS0 = 1 in MODE register).

Note 3 In the case of an extended, modulo 128 control field format (MCS = 1 in SAP2 register) the control field is stored in the RHCR in compressed form (I frames).

Note 4 In the case of extended control field, only the first byte is stored in the RHCR, the second in the RFIFO.

When 32 bytes of a message longer than that are stored in the RFIFO, the CPU is prompted to read out the data by an RPF interrupt. The CPU must handle this interrupt before more than 32 additional bytes are received, which would cause a "data overflow".This corresponds to a maximum CPU reaction time of 16 ms (data rate 16 kbit/s).

After a remaining block of less than or equal to 16 bytes has been stored, it is possible to store the first 16 bytes of a new message (**see figure 33**).

The internal memory is now full. The arrival of additional bytes will result in "data overflow" (RSTA:RDO) and a third new message in "frame overflow" (EXIR:RFO).
The generated interrupts are inserted together with all additional information into a queue to be individually passed to the CPU.

After an RPF or RME interrupt has been processed, i.e. the received data has been read from the RFIFO, this must be explicitly acknowledged by the CPU issuing an RMC (Receive Message Complete) command.

The ISAC-S TE can then release the associated FIFO pool for new data. If there is an additional interrupt in the queue it will be generated after the RMC acknowledgement.

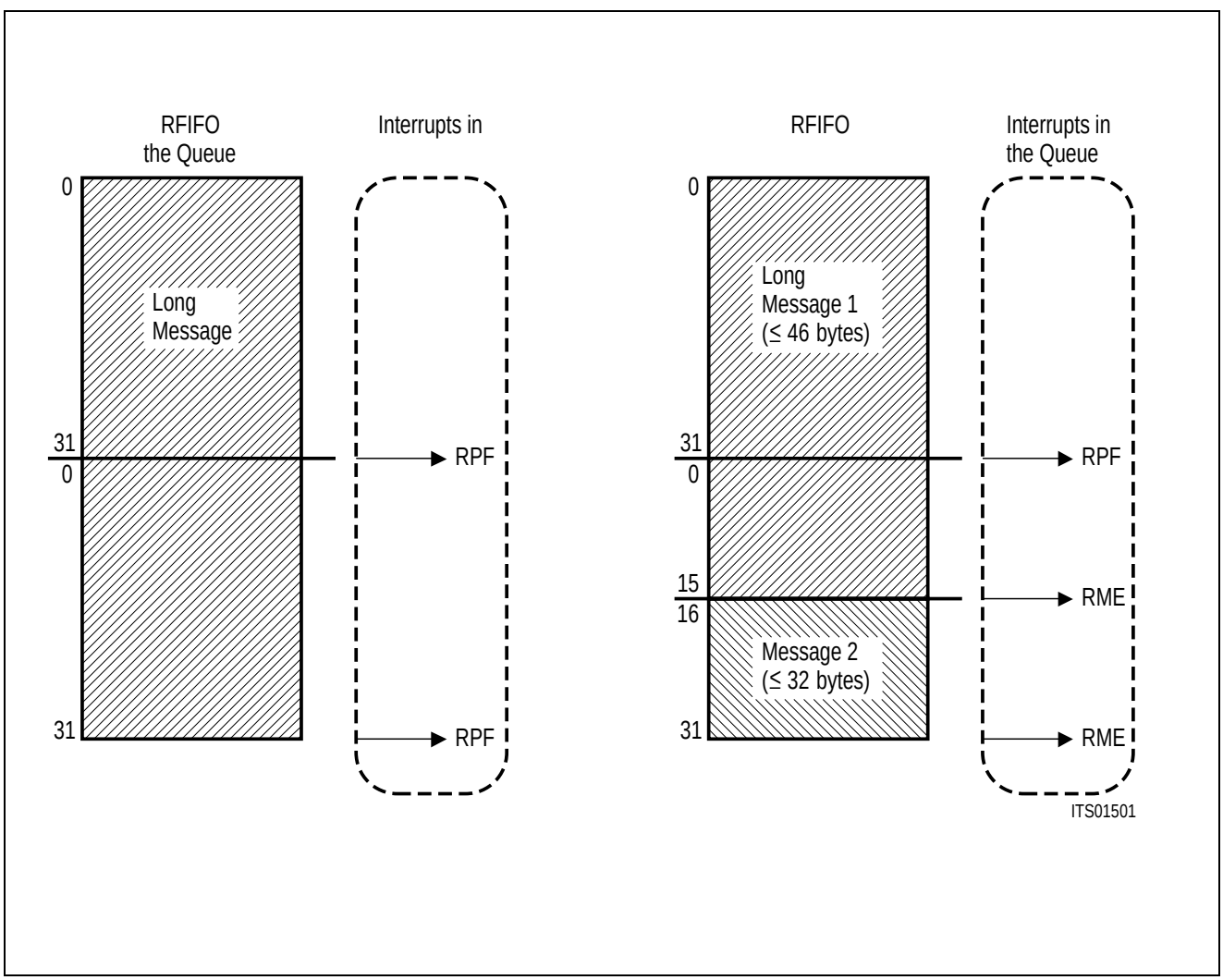


Figure 33
Contents of the RFIFO (long messages)

Information about the received frame is available for the µP when a RME interrupt is generated, as shown in **table 6.**

Table 6
Receive Information at RME Interrupt

Information	Register (adr. hex)	Bit	Mode
First byte after flag (SAPI of LAPD address field)	SAPR (26)	–	Transparent mode 1
Control field	RHCR (29)	–	Auto-mode, I (modulo 8) and U frames
Compressed control field	RHCR (29)	–	Auto-mode, I frames (modulo 128)
2nd byte after flag	RHCR (29)	–	Non-auto mode, 1-byte address field
3rd byte after flag	RHCR (29)	–	Non-auto mode, 2-byte address field Transparent mode 1
Type of frame (Command/Response)	RSTA (27)	C/R	Auto-mode, 2 byte address field Non-auto mode, 2-byte address field Transparent mode 3
Recognition of SAPI	RSTA (27)	SA1-0	Auto-mode, 2 byte address field Non-auto mode, 2-byte address field Transparent mode 3
Recognition of TEI	RSTA (27)	TA	All except transparent modes 2, 3
Result of CRC check (correct/incorrect)	RSTA (27)	CRC	ALL
Data available in RFIFO (yes/no)	RSTA (27)	RDA	ALL
Abort condition detected (yes/no)	RSTA (27)	RAB	ALL
Data overflow during reception of a frame (yes/no)	RSTA (27)	RDO	ALL
Number of bytes received in RFIFO	RBCL (25)	RBC4-0	ALL
Message length	RBCL (25) RBCH (2A)	RBC11-0 OV	ALL

2.7.4 Transmission of Frames

A 2 × 32 byte FIFO buffer (transmit pools) is provided in the transmit direction.

If the transmit pool is ready (which is true after an XPR interrupt or if the XFW bit in STAR is set), the CPU can write a data block of up to 32 bytes to the transmit FIFO. After this, data transmission can be initiated by command.

Two different frames types can be transmitted:

– Transparent frame (command: XTF), or

– I frames (command: XIF)

as shown in **figure 34**.

For transparent frames, the whole frame including address and control field must be written to the XFIFO.

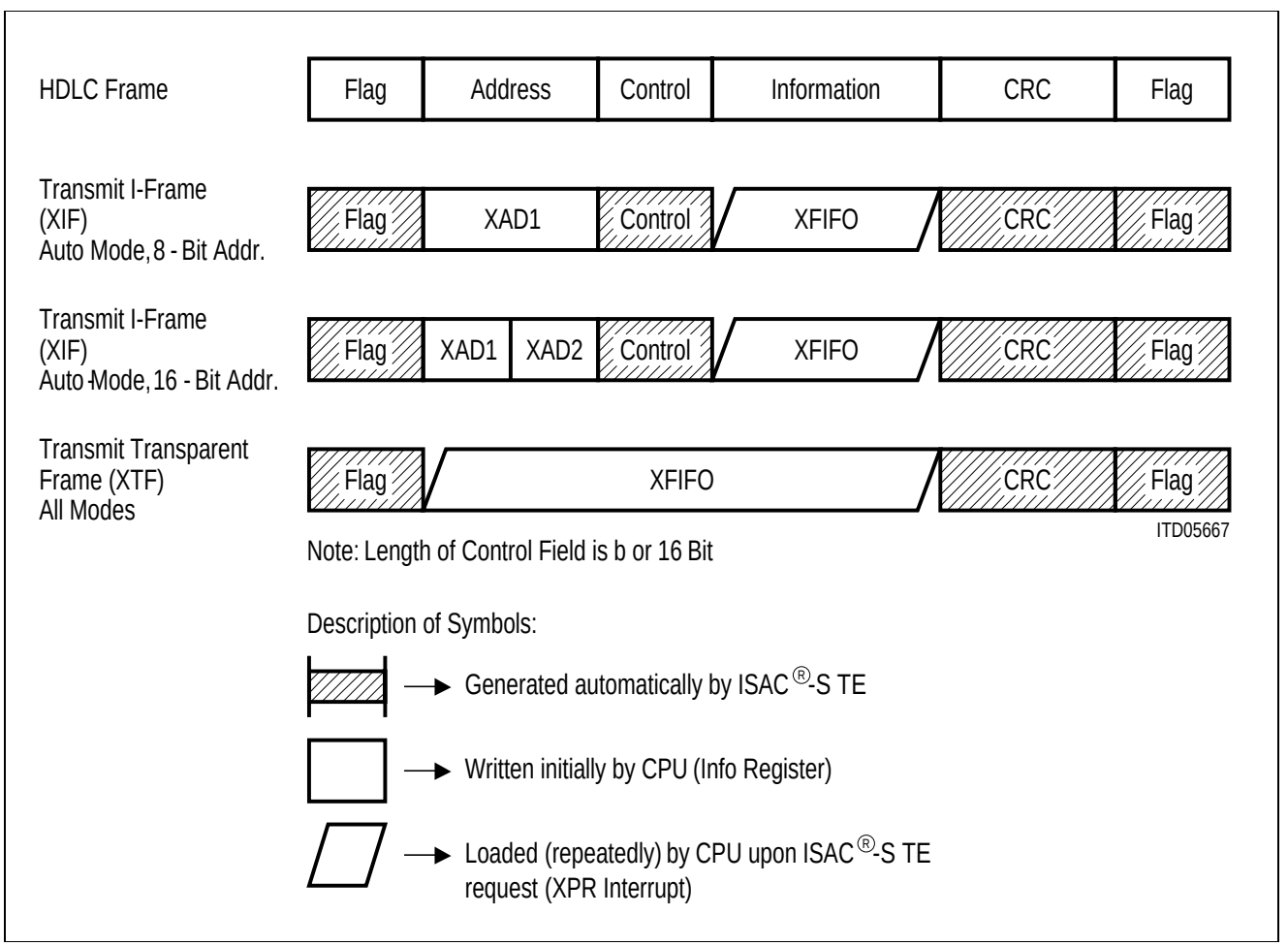


Figure 34
Transmitter Data Flow

The transmission of I frames is possible only if the ISAC-S TE is operating in the auto-mode. The address and control field is autonomously generated by the ISAC-S TE and appended to the frame, only the data in the information field must be written to the XFIFO.

If a 2-byte address field has been selected, the ISAC-S TE takes the contents of the XAD 1 register to build the high byte of the address field, and the contents of the XAD 2 register to build the low byte of the address field.

Additionally the C/R bit (bit 1 of the high byte address, as defined by LAPD protocol) is set to "1" or "0" dependent on whether the frame is a command or a response.

In the case of a 1 byte address, the ISAC-S TE takes either the XAD 1 or XAD 2 register to differentiate between command or response frame (as defined by X.25 LAPB).

The control field is also generated by the ISAC-S TE including the receive and send sequence number and the poll/final (P/F) bit. For this purpose, the ISAC-S TE internally manages send and receive sequence number counters.

In the auto-mode, S frames are sent autonomously by the ISAC-S TE. The transmission of U frames, however, must be done by the CPU. U frames must be sent as transparent frames (CMDR:XTF), i.e. address and control field must be defined by the CPU.

Once the data transmission has been initiated by command (CMDR:XTF or XIF), the data transfer between CPU and the ISAC-S TE is controlled by interrupts.

The ISAC-S TE repeatedly requests another data packet or block by means of an ISTA:XPR interrupt, every time no more than 32 bytes are stored in the XFIFO.

The processor can then write further data to the XFIFO and enable the continuation of frame transmission by issuing an XIF/XTF command.

If the data block which has been written last to the XFIFO completes the current frame, this must be indicated additionally by setting the XME (Transmit Message End) command bit. The ISAC-S TE then terminates the frame properly by appending the CRC and closing flag.

If the CPU fails to respond to an XPR interrupt within the given reaction time, a data underrun condition occurs (XFIFO holds no further valid data). In this case, the ISAC-S TE automatically aborts the current frame by sending seven consecutive "ones" (ABORT sequence).

The CPU is informed about this via an XDU (Transmit Data Underrun) interrupt.

It is also possible to abort a message by software by issuing a CMDR:XRES (Transmitter RESet) command, which causes an XPR interrupt.

After an end of message indication from the CPU (CMDR:XME command), the termination of the transmission operation is indicated differently, depending on the selected message transfer mode and the transmitted frame type.

If the ISAC-S TE is operating in the **auto mode**, the window size (= number of outstanding unacknowledged frames) is limited to "1"; therefore an acknowledgement is expected for every I frame sent with an XIF command. The acknowledgement may be provided either by a received S or I frame with corresponding receive sequence number.

If no acknowledgement is received within a certain time (programmable), the ISAC-S TE requests an acknowledgement by sending an S frame with the poll bit set (P = 1) (RR or RNR). If no response is received again, this process is repeated in total N2 times (retry count, programmable via TIMR register).

The termination of the transmission operation may be indicated either with:

– XPR interrupt, if a positive acknowledgement has been received,

– XMR interrupt, if a negative acknowledgement has been received, i.e. the transmitted message must be repeated (XMR = Transmit Message Repeat),

– TIN interrupt, if no acknowledgement has been received at all after N2 times the expiration of the time period t_1 (TIN = Timer INterrupt, XPR interrupt is issued additionally).

Note: Prerequisite for sending I frames in the auto-mode (XIF) is that the **internal** operational mode of the timer has been selected in the MODE register (TMD bit = 1).

The transparent transmission of frames (XTF command) is possible in all message transfer modes. The successful termination of a transparent transmission is indicated by an XPR interrupt.

In all cases, collisions which occur on the S-Bus (D channel) before the first XFIFO pool has been completely transmitted and released are treated without µP interaction. The ISAC-S TE will retransmit the frame automatically.

If a collision is detected after the first pool has been released, the ISAC-S TE aborts the frame and requests the processor to repeat the frame with an XMR interrupt.

2.7.5 Documentation of the Auto Mode

The auto mode of the ICC and ISAC-S TE is only applicable for the states 7 and 8 of the LAPD protocol. All other states (1 to 6) have to be performed in Non-Auto Mode (NAM). Therefore this documentation gives an overview of how the device reacts in the states 7 and 8, which reactions require software programming and which are done by the hardware itself, when interrupts and status register contents are set or change. The necessary software actions are also detailed in terms of command or mode register access.

The description is based on the SDL diagrams of the ETSI TS 46-20 dated 1989.

The diagrams are only annotated by documentary signs or texts (mostly register descriptions) and can therefore easily be interpreted by anyone familiar with the SDL description of LAPD. All deviations that occur are specially marked and the impossible actions, paths etc. are crossed out.

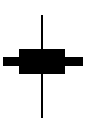
To get acquainted with this documentation, first read through the legend-description and the additional general considerations, then start with the diagrams, referring to the legend and the register description in the Technical Manual if necessary.

We hope you will profit from this documentation and use our software-saving auto-mode.

2.7.5.1 Legend of the Auto-Mode Documentation

a.Symbols within a path

There are 3 symbols within a path

a.1. 

In the auto-mode the device processes all subsequent state transitions branchings etc. up to the next symbol.

a.2. 

In the auto-mode the device does not process the state transitions, branchings etc. Within the path appropriate directions are given with which the software can accomplish the required action.

a.3. 

A path cannot be implemented and no software or hardware action can change this. These paths are either optional or only applicable for window-size > 1.

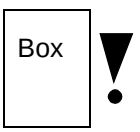
b.Symbols at a path

There is 1 symbol at a path

b.1.  marks the beginning of a path, for which a.3 applies.

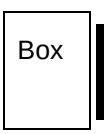
c.Symbols at an internal or external message box.

There are 2 symbols at a message box.

c.1. 

This symbol means, that the action described in the box is not possible. Either the action specified is not done at all or an additional action is taken (written into the box).

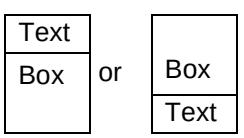
Note: The impossibility to perform the optional T203 timer-procedure is not explicitly mentioned; the corresponding actions are only crossed out.

c.2. 

This symbol means, that within a software-path, by taking the prescribed register actions the contents of the box will be done automatically.

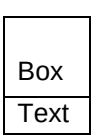
d.Text within boxes

Text within boxes can be grouped in one of two classes.

d.1. 

The text denotes an interrupt which is always associated with the event. (But can also be associated with other events). (See ISTA- and EXIR-register description in the Technical Manual for an interrupt description).

d.2. The text describes a register access

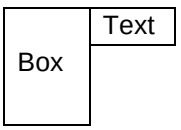


either a register read access to discriminate this state from others or to reach a branching condition

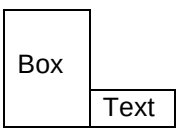
or a register write access to give a command.

The text is placed in the box that describes the functions for which the register access is needed.

e.Text attached at the side of boxes

e.1. 

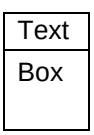
The text describes an interrupt associated with the contents of the box. The interrupt is always associated with the box contents, if the interrupt name is not followed by a "/", it is associated only under appropriate conditions if a "/" is behind it.

e.2. 

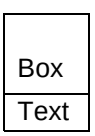
The text describes a possible or mandatory change of a bit in a status-register associated with the contents of the box.

(The attached texts can also be placed on the left side.)

f.Text above and below boxes

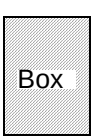
f.1. 

Text describes a mandatory action to be performed on the contents of the box.

f.2. 

Text describes a mandatory action to be taken as a result of the contents of the box.
Action here means register access.

g.Shade boxes



The box describes an impossible state or action for the device.

2.7.5.2 Additional General Considerations when Using the Auto Mode

a)Switching from auto-mode to non-auto mode.

As mentioned in the introduction the auto mode is only applicable in the states 7 and 8 of the LAPD. Therefore whenever these states have to be left (which is indicated by a "Mode:NAM" text) there are several actions to be taken that could not all be detailed in the SDL diagrams:

a.1)write non-auto mode and TMD = 0 into the mode register.

a.2)write the timer register with an arbitrary value to stop it. The timer T200 as specified in the LAPD protocol is implemented in the hardware only in the states 7 and 8; in all other states this or any other timer-procedure has to be done by the software with the possible use of the timer in external timer mode.

a.3)read the WFA bit of the STAR2 register and store it in a software variable. The information in this bit may be necessary for later decisions. When switching from auto mode to non-auto mode XPR interrupts may be lost.

a.4)In the non-auto mode the software has to decode I-, U- and S frames because I and S frames are only handled autonomously in the auto-mode.

a.5)The RSC and PCE interrupts, the contents of the STAR2 register and the RRNR bit in the STAR register are only meaningful within the auto-mode.

a.6)leave some time before RHR or XRES is written to reset the counters, as a currently sent frame may not be finished yet.

b)What has to be written to the XFIFO?

In the legend description when the software has to write contents of a frame to the XFIFO only "XFIFO" is shown in the corresponding box. We shall give here a general rule of what has to be written to the XFIFO:

a) For sending an I frame with CMDR:XIF, only the information field content, i.e. no SAPI, TEI, Control field should be written to the XFIFO.

b) For sending a U frame or any other frame with CMDR:XTF, the SAPI, TEI and the control field has to be written to the XFIFO.

c)The interrupts XPR and XMR.

The occurence of an XPR interrupt in auto-mode after an XIF command indicates that the I frame sent was acknowledged and the next I frame can be sent, if STAR2:TREC indicates state 7 and STAR:RRNR indicates Peer Rec not busy. If Peer Rec is busy after an XPR, the software should wait for the next RSC interrupt before sending the next I frame. If the XPR happens to be in the timer recovery state, the software has to poll the STAR2 register until the state multiple frame established is reached or a TIN interrupt is issued which requires auto mode to be left (One of these two conditions will occur before the time T200 $\times$ N200). In non-auto mode or after an XTF command the XPR just indicates, that the frame was sent successfully.

The occurence of an XMR interrupt in auto-mode after an XIF command indicates that the I frame sent was either rejected by the Peer Entity or that a collision occured on the S interface. In both cases the I frame has to be retransmitted (after an eventual waiting for the RSC interrupt if the Peer Rec was busy; after an XMR the device will always be in the state 7). In non-auto mode or after an XTF command the XMR indicates that a collision occured on the S interface and the frame has to be retransmitted.

d)The resetting of the RC variable:

The RC variable is reset in the ICC and ISAC-S TE when leaving the state timer recovery. The SDL diagrams indicate a reset in the state multiple frame established when T200 expires. There is no difference to the outside world between these implementations however our implementation is clearer.

e)The timer T203 procedure:

We do not fully support the optional timer T203 procedure, but we can still find out whether or not S frames are sent on the link in the auto-mode. By polling the STAR2:SDET bit and (re)starting a software timer whenever a one is read we can build a quasi T203 procedure which handles approximately the same task. When T203 expires one is supposed to go into the timer recovery state with RC = 0. This is possible for the ICC and ISAC-S TE by just writing the STI bit in the CMDR register (auto-mode and internal timer mode assumed).

f)The congestion procedure as defined in the 1 TR 6 of the "Deutsche Bundespost":

In the 1 TR 6 a variable N2 $\times$ 4 is defined for the maximum number of Peer Busy requests. The 1 TR 6 is in this respect not compatible with the Q921 of CCITT or the ETSI 46-20 but it is, nevertheless, sensible to avoid getting into a hangup situation. With the ICC and ISAC-S TE this procedure can be implemented:

After receiving an RSC interrupt with RRNR set one starts a software-timer. The timer is reset and stopped if one either receives another RSC interrupt with a reset RRNR, if one receives a TIN interrupt or if other conditions occur that result in a reestablishment of the link. The timer expires after N2 $\times$ 4 $\times$ T200 and in this case the 1 TR 6 recommends a reestablishment of the link.

2.7.5.3 Dealing With Error Conditions in Auto Mode

In the Recommendation Q.921 of CCITT (Blue Book) several error conditions are described. We shall deal with them as far as they touch the auto mode of the ISAC-S (which only applies for states 7,8 of Q.921).

Throughout the following document in subsections 1 we shall give the original Q.921-Text. For better discrimination against comments the original text is printed in italic characters. Please note that **Q.921/table 5** has been corrected according to Corrigendum No. 1 10/1989. Subsections 2 document how the ISAC-S react in all cases, and subsections 3 will give hints how your software should respond to these reactions.

Invalid Frames and Frame Abortion

During data transmission invalid frames and frame abortion generally lead to error conditions.

Q921: Invalid Frames and Frame Abortion

Paragraphs 2.9 and 2.10 of the Q.921 deal with Invalid Frames and Frame Abortion. In the following the original text is given.

Q.921 § 2.9: Invalid Frames

An invalid frame is a frame which:

a) is not properly bounded by two flags, or
b) has fewer than 6 octets between flags or frames that contain sequence numbers, or
c) does not consist of an integral number of octets prior to zero bit insertion or following zero bit extraction, or
d) contains a frame check sequence error, or
e) contains a single octet address field, or
*f) contains a service access point identifier (**see § 3.3.3**) which is not supported by the receiver.*

Invalid frames shall be discarded without notification to the sender. No action is taken as the result of that frame.

Q.921 § 2.10: Frame Abort

Receipt of seven or more contiguous 1 bits shall be interpreted as an abort and the data link layer shall ignore the frame currently being received.

Reaction of the ISAC-S TE

a) A frame which does not start with a flag is discarded in the ISAC-S TE. A frame which does not end with a flag is one, that is aborted, i.e. if § 2.9b does not apply then the ISAC-S TE
– discards the frame, if it was an S-frame
or, if it was an I or U-frame
– generates an ISTA: RME (or RPFs and a RME) and
– puts RSTA: RAB = 1 after the RME-Interrupt RAB = 1.

A frame is supposed to be unbounded according to § 5.8.5 if the byte counter RBCH, RBCL after RPF or RME exceeds 528.

b) The frame is discarded by the ISAC-S TE if
with **U-frames or undefined frames**it contains less or equal to 4 octets or
with **I-frames**it contains less or equal to 5 octets
with **S-frames** it contains less than 6 octets.

For U-frames with a content between 4 and 5 octets exclusively or for I-frames between 5 and 6 octets exclusively an ISTA: RME interrupt is generated and afterwards the RSTA: CRC is set to 0.

c) An S-frame is discarded. In the own-receiver-busy state I-frames are discarded.
For an I-frame in the normal state and U frames, after several possible RPF interrupts and the final RME interrupt, the bit RSTA: CRC is set to 0 in this case.

d) In case of an -S frame, the frame is discarded
-U and I-framesRSTA: CRC is set to "0" in this case.

e) the frame is discarded
f) the frame is discarded

The reaction to § 2.10 has been already discussed under a)

Necessary Software Actions

The software should read the Register RSTA after a RME-interrupt. After having read RAB = 1 or CRC = 0, all frame contents read from the FIFO should be discarded and a CMDR: RMC should be written. After each RPF or RBCH, RBCL should be read and if it exceeds 528, CMDR: RRES should be written. In this way all invalid frames are discarded by the software.

Data Overflow

In case of a data overflow, which is only possible while receiving an I-frame or an U-frame with a non-empty information field, the ISAC-S TE interrupt with ISTA: RME and sets RSTA: RDO to 1. A RSTA: RDO and an ISTA: RFO are a hint that the dynamic reaction time of your software to the RPF, RME interrupt is too slow, so you should change your software. During the development phase you may set CMDR: RNR after an RDO, RFO-condition to protect against further errors, but the final solution can only be to exclude RDO, RFO conditions by an improved software design.

Frame Rejection Condition

Q.921 § 5.8.5: Frame Rejection Condition

A frame rejection condition results from one of the following conditions:

*a) the receipt of an undefined frame (**see § 3.6.1**, third paragraph)*
b) the receipt of a supervisory or unnumbered frame with incorrect length
c) the receipt of an invalid N(R), or
d) the receipt of a frame with an information field which exceeds the maximum established length.

Upon occurrence of a frame rejection condition whilst in the multiple frame operation, the data link layer entity shall:

– issue a MDL-ERROR-INDICATION primitive, and
*– initiate re-establishment (**see § 5.7.2**).*

Upon occurrence of a frame rejection condition during establishment of or release from multiple frame operation, or whilst a data link is not established, the data link layer entity shall:

– issue a MDL-ERROR-INDICATION primitive, and
– discard the frame.

Note: *For satisfactory operation it is essential that a receiver is able to discriminate between invalid frames, as defined in § 2.9, and frames with an information field which exceeds the maximum established length (**see § 3.6.11 item d**). An unbounded frame may be assumed, and thus discarded, if two times the longest permissible frame plus two octets are received without a flag detection.*

For a better understanding we insert the text of § 3.6.1, which is referred to in § 5.8.5 and which reads:

§ 3.6.1 Commands and responses

The following commands and responses are used by either the user or the network data link layer entities and are represented in **Q.921/table 5**. Each data link connection shall support the full set of commands and responses for each application implemented. The frame types associated with each of the two applications are identified in **Q.921/table 5**.

Frame types associated with an application not implemented shall be discarded and no action shall be taken as a result of that frame.

For purposes of the LAPD procedures in each application, those frame types not identified in **Q.921/table 5** are identified as undefined command and/or response control field. The actions to be taken are specified in § 5.8.5.

We include the original **table 5** which is mentioned in § 3.6.1:

Table 7
Q.921 (Table 5)

Application	Format	Commands	Responses	Encoding 8 7 6 5 4 3 2 1	Octet
Unacknowledged and Multiple-Frame acknowledged Information Transfer	Information Transfer	I(nformation)		N(S) 0	4
				N(R) P	5
	Supervisory	RR (receive ready)	RR (receive ready)	0 0 0 0 0 0 0 1	4
				N(R) P/F	5
		RNR (receive not ready)	RNR (receive not ready)	0 0 0 0 0 1 0 1	4
				N(R) P/F	5
		REJ (reject)	REJ (reject)	0 0 0 0 1 0 0 1	4
				N(R) P/F	5
	Unnumbered	SABME (set async. balanced Mode extd).		0 1 1 P 1 1 1 1	4
			DM (disconnected mode)	0 0 0 F 1 1 1 1	4
		UI (Unnumbered Information)		0 0 0 P 0 0 1 1	4
		DISC (disconnect)		0 1 0 P 0 0 1 1	4
			UA (unnumbered Acknowledgement)	0 1 1 F 0 0 1 1	4
			FRMR (frame reject)	1 0 0 F 0 1 1 1	4
Connection Management		XID* (Exch. Ident)	XID* (Exch. Ident)	1 0 1 P/F 1 1 1 1	4

***Note:** Use of the XID frame other than for parameter negotiation procedures (**see § 5.4**) is for further study. The commands and responses in **Q.921/table 5** are defined in § 3.6.2 to § 3.6.12

Reaction of the ISAC-S TE

In the following various possible actions to be taken according to § 5.8.5 parts a) through c) are discussed separately.

a) There are different types of undefined frames:

1) I-frame which is not a command an ISTA: PCE-interrupt is generated

2) S-frame with bits 8-5 in Octet 4 = 0 an ISTA: PCE-interrupt is generated

3) A frame with bits 4-1 in octet 4 equal to "1101" (selective reject) — an ISTA: PCE is generated

4) Frame with bits 2-1 in octet 4 equal to "11" but control field not contained in ISTA: RME interrupt; — the control field can be read afterwards in RHCR (after having checked for invalid frame condition).

5) SABME, UI, DISC, not a command, DM, UA, FRMR not a response ISTA: RME interrupt; — the control field can be read afterwards in RHCR, the C/R-bit in the SAPR-register (after having checked for invalid frame condition).

b) If the length of the frame is too small 1.1.1b) applies and the frame is invalid. Therefore incorrect length can only mean:

1) S-frame with more than 6 octets an ISTA:PCE-interrupt is generated; the contents of the additional octets is discarded.

2) Undefined frames with 5 octets, bits 2-1 in octet 4 not being equal to "11" (e.g. modulo 8 S-frame) — an ISTA:PCE-interrupt is generated

3) SABME, BM, DISC, UA-frame with more than 5 octets — after ISTA: RME and identifying the frame by RHCR the RSTA:RDA bit is 1 if the frames had more than 5 octets and 0 if they had exactly 5 octets.

4) A FRMR with not exactly 10 octetsAfter a RME and identifying FRMR by reading RHCR-register, the software has to read RBCH, RBCL. If OV = 1 or RBC11-RBC0 = 0 … 101 then the FRMR did not have exactly 10 octets.

c) An invalid N(R) is one that does not meet the condition

$$V(A) < N(R) < V(S)$$

This condition is automatically checked within the device and in the case of an invalid N(R) an ISTA:PCE-interrupt is generated. An S-field response is done by the ISAC-S TE in all prescribed cases of invalid N(R) automatically.

The processor should read RBCH, RBCL after each RPF, RME interrupt. If after an RPF or RME the byte count exceeds 528 then CMDR:RRES should be written (abort of frame). The frame was invalid in this case but it was not a frame rejection condition. If after a RME the byte count was between 260 and 528 inclusively and no other invalidity condition according to section 1 applies or a data overflow according to section 2 occurred then a frame rejection condition is detected.

Necessary Software Reactions

The software can find out all frame rejection conditions either by receiving PCE or by checking RSTA, SAPR, RHCR, RBCH, RBCL after a RME interrupt, and RBCH, RBCL after an RPF interrupt. In case of U-frames it has to be checked before, whether or not it is an invalid frame and has only to be discarded or, whether it was valid but leads to a frame rejection condition. (Only valid frames can lead to frame rejection conditions according to § 5.8.4 of Q.921).

In case of a frame rejection condition the software has to take the actions defined in § 5.7.2 and issue a MDL-ERROR-INDICATION.

The particular action in § 5.7.2 reads:

§ 5.7.2 Procedures

In all re-establishment situations, the data link layer entity shall follow the procedures defined in § 5.5.1. All locally generated conditions for re-establishment will cause the transmission of the SABME.

In case of data link layer and peer initiated re-establishments, the data link layer entity shall also

- *Issue a MDL-ERROR-INDICATION primitive to the connection management entity: and*
- *rf V(S) > V(A) prior to re-establishment issue a DL-ESTABLISH-INDICATION primitive to layer 3 and discard all I-queues.*

In case of layer-3 initiated re-establishment, or if a DL-ESTABLISH-REQUEST primitive occurs pending re-establishment, the DL-ESTABLISH-CONFIRM primitive shall be used.

A frame rejection condition is not a peer initiated re-establishment.

§ 5.5.1 is pretty voluminous. Here just the necessary actions to be done with the ISAC-S TE shall be given, in case the re-establishment is successful at once:

- the software should set the ISAC-S TE into non-auto mode by writing the Mode register MODE: $6x_H$. Further actions that result from switching to non-auto mode should also be taken according.
- it should write FIFO : 76_H, $6F_H$, CMDR : XTF to send a SABME-command with p = 1.
- upon having received a correct UA-frame it should
  - write CMDR : XRES, RRES to set V(S) = V(A) = V(R) = 0
  - write MODE: $3x_H$ to re-enter auto mode for the multiple-frame established state.

If the re-establishment is not successful at once, in the non-auto mode further software actions according to § 5.5.1 have to be taken.

Further Criteria Leading to a Re-Establishment

Q.921 § 5.7.1: Criteria for Re-Establishment

§ 5.7.1 Criteria for re-establishment

The criteria for re-establishing the multiple frame mode of operation are defined in this section by the following conditions:

a) The receipt while in the multiple frame mode of operation, of an SABME;
*b) The receipt of a DL-ESTABLISH-REQUEST primitive from layer 3 (**see § 5.5.1.1**);*
*c) The occurrence of N200 re-transmission failures while in the timer recovery condition (**see § 5.6.7**)*
d) The occurrence of a frame rejection condition as identified in § 5.8.5;
*e) On the receipt, while in the multiple frame mode of operation of an FRMR response frame (**see § 5.8.6**);*
*f) The receipt, while in the multiple frame mode of operation, of an unsolicited DM response with the F bit set to 0 (**see § 5.8.7**);*
g) The receipt while in the timer recovery condition, of a DM response with the F bit set to 1.

Reaction of the ISAC-S TE

a) after having checked for validity and non-occurrence of a frame rejection condition, the error free SABME can be identified after RME-Interrupt by reading the RHCR-register; the multiple frame est/timer recovery discrimination can be done by reading STAR2: TREC
b) –
c) A TIN-Interrupt occurs (of course MODE: TMD has to have been 1)
d) see section 3
e) see a)
f) see a)
g) see a)

Necessary Software Reactions

The same actions as in section 3 have to be taken. In addition, in case of a) the necessary discrimination for the software is possible by reading STAR2: WFA while still in auto mode. If WFA = 1 then V(S) > V(A); if WFA = 0, then V(S) = V(A).

Further Possible Error Conditions

Appendix II of Q.921: Further Possible Error Conditions

Table 8
Q.921
Management Entity Actions for MDL Error Indications

Error Type	Error Code	Error Condition	Affected States	Network Management Action	User Management Action
Receipt of unsolicited response	A	Supervisory (F = 1)	7	Error log	Dependent on implementation
	B	DM(F = 1)	7, 8	Error log	Dependent on implementation
	C	UA(F = 1)	4, 7, 8	TEI removal procedure or TEI check procedure; then, if TEI: – free, remove TEI – single, no action multiple: TEI removal procedure	TEI identity verify procedure or remove TEI
	D	UA(F = 1)	4, 5, 6, 7, 8		
	E	Receipt of DM response (F = 0)	7, 8	Error log	Dependent on implementation
Peer initiated Re-establishment	F	SABME	7, 8	Error log	Dependent on implementation
Unsuccessful Re-transmission (N200 times)	G	SABME	5	TEI check procedure; then, if TEI: – free, remove TEI – single, no action multiple: TEI removal procedure	TEI identity verify procedure or remove TEI
	H	DISC	6		
	I	Status Inquiry	8	Error log	Dependent on implementation

Table 8
Q.921
Management Entity Actions for MDL Error Indications (cont'd)

Error Type	Error Code	Error Condition	Affected States	Network Management Action	User Management Action
Other	J	N(R) Error	7, 8	Error log	Dependent on implementation
	K	Receipt of FRMR response	7, 8	Error log	Dependent on implementation
	L	Receipt of non implemented frame	4, 5, 6, 7, 8	Error log	Dependent on implementation
	M (**see Note 2**)	Receipt of I-field not permitted	4, 5, 6, 7, 8	Error log	Dependent on implementation
	N	Receipt of frame with wrong size	4, 5, 6, 7, 8	Error log	Dependent on implementation
	O	N201 Error	4, 5, 6, 7, 8	Error log	Dependent on implementation

Note 1: For the description of the affected states see Annex B.

Note 2: According to Q.921 § 5.8.5 this error code will never be generated.

Reactions of the ISAC-S TE and Necessary Software Reactions

As the auto mode is only to be used in states 7, 8 and as it has to be switched to non-auto mode where in states 1-6, we do not have to deal with error code G and H.

A) The ISAC-S does not react at all (our implementation). The software is not informed, as no action is mandatory according to Q.921.
B) see further Criteria Leading to a Reestablishment"
C) see further Criteria Leading to a Reestablishment
D) see further Criteria Leading to a Reestablishment
E) see further Criteria Leading to a Reestablishment
F see further Criteria Leading to a Reestablishment
I) see further Criteria Leading to a Reestablishment
J) see Frame Rejection Condition
K) see further Criteria Leading to a Reestablishment
L) see Frame Rejection Condition
M)
N) see Frame Rejection Condition
O) only internal software timer, no device action.

Conclusion:

For your error-processing with ISAC-S we suggest to implement the software design shown in the following **figures 35** through **38** into your interrupt service routine.

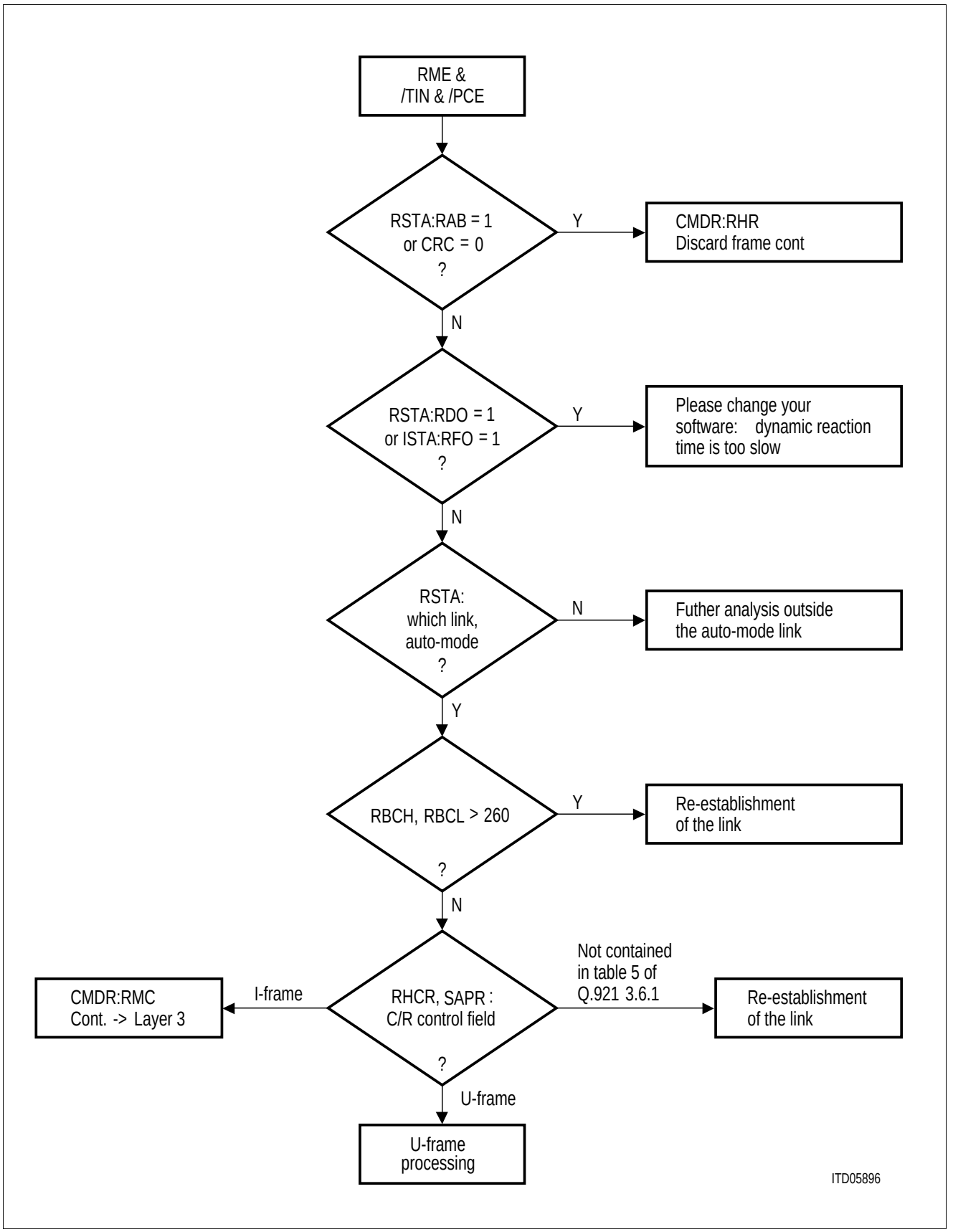


Figure 35
Interrupt Service Routine after RME

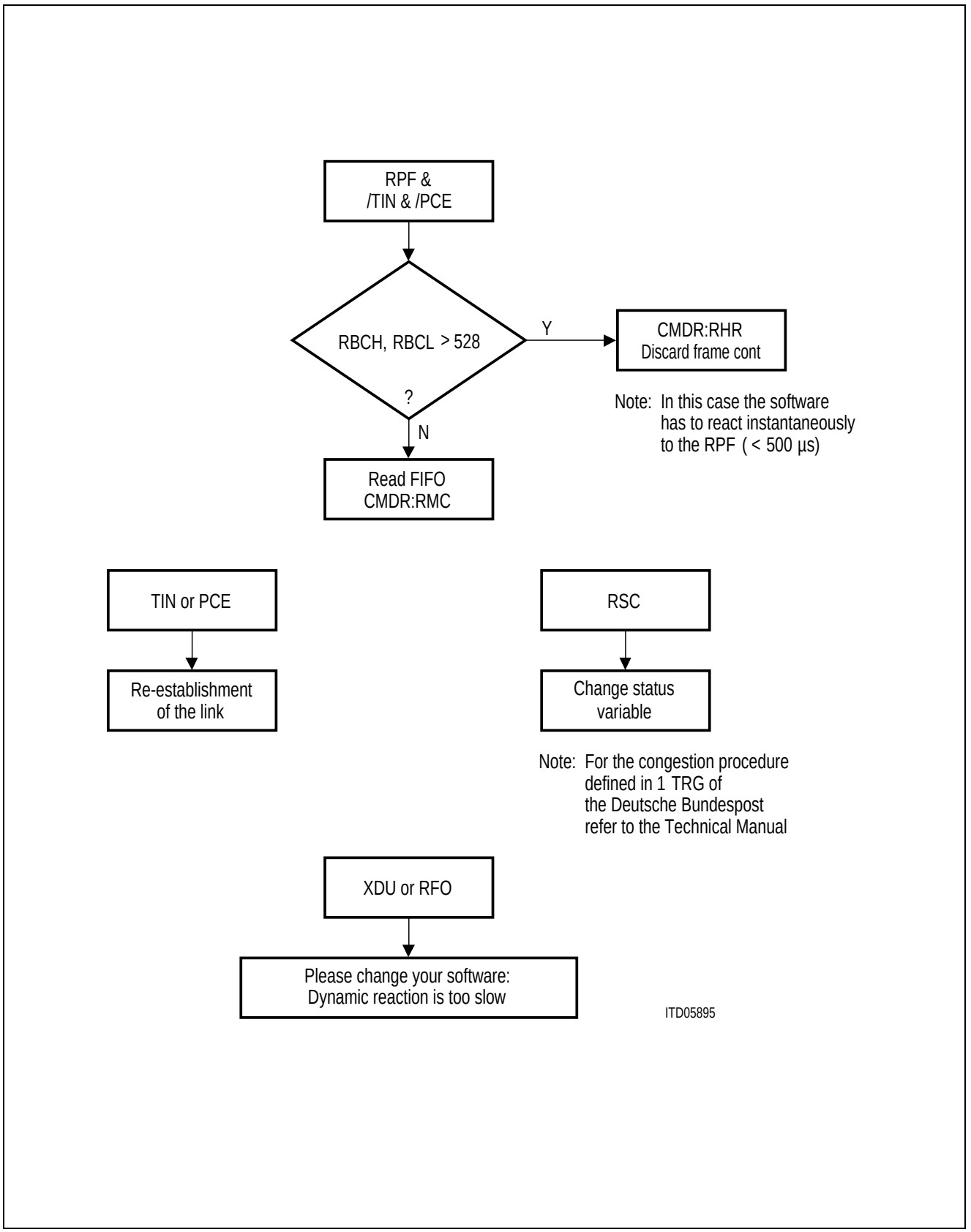


Figure 36
Interrupt Service Routines after RPF (top), TIN or PCE (middle left), RSC (middle right), and XDU or RFO (bottom)

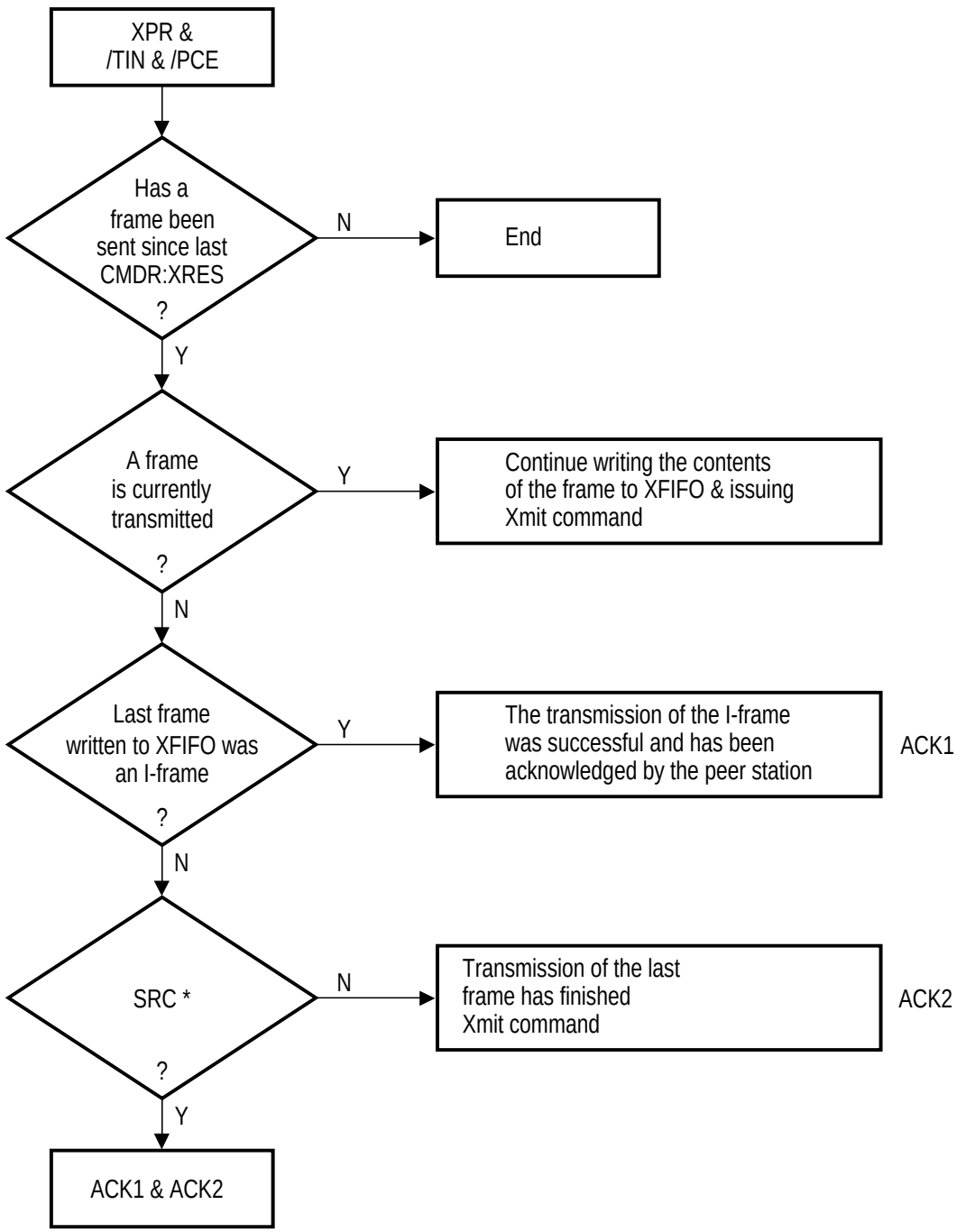


Figure 37
Interrupt Service Routine after XPR

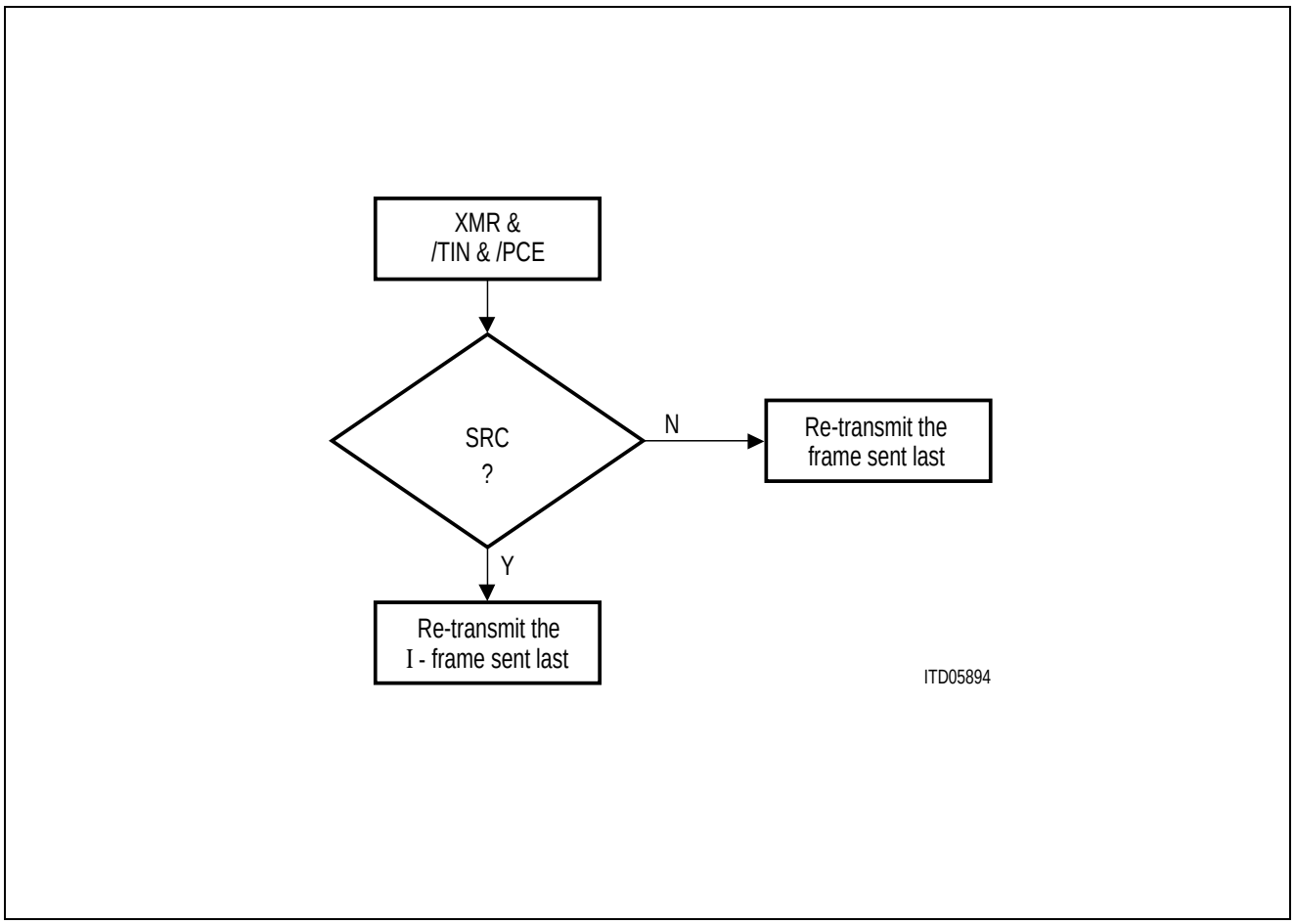


Figure 38
Interrupt Service Routine after XMR

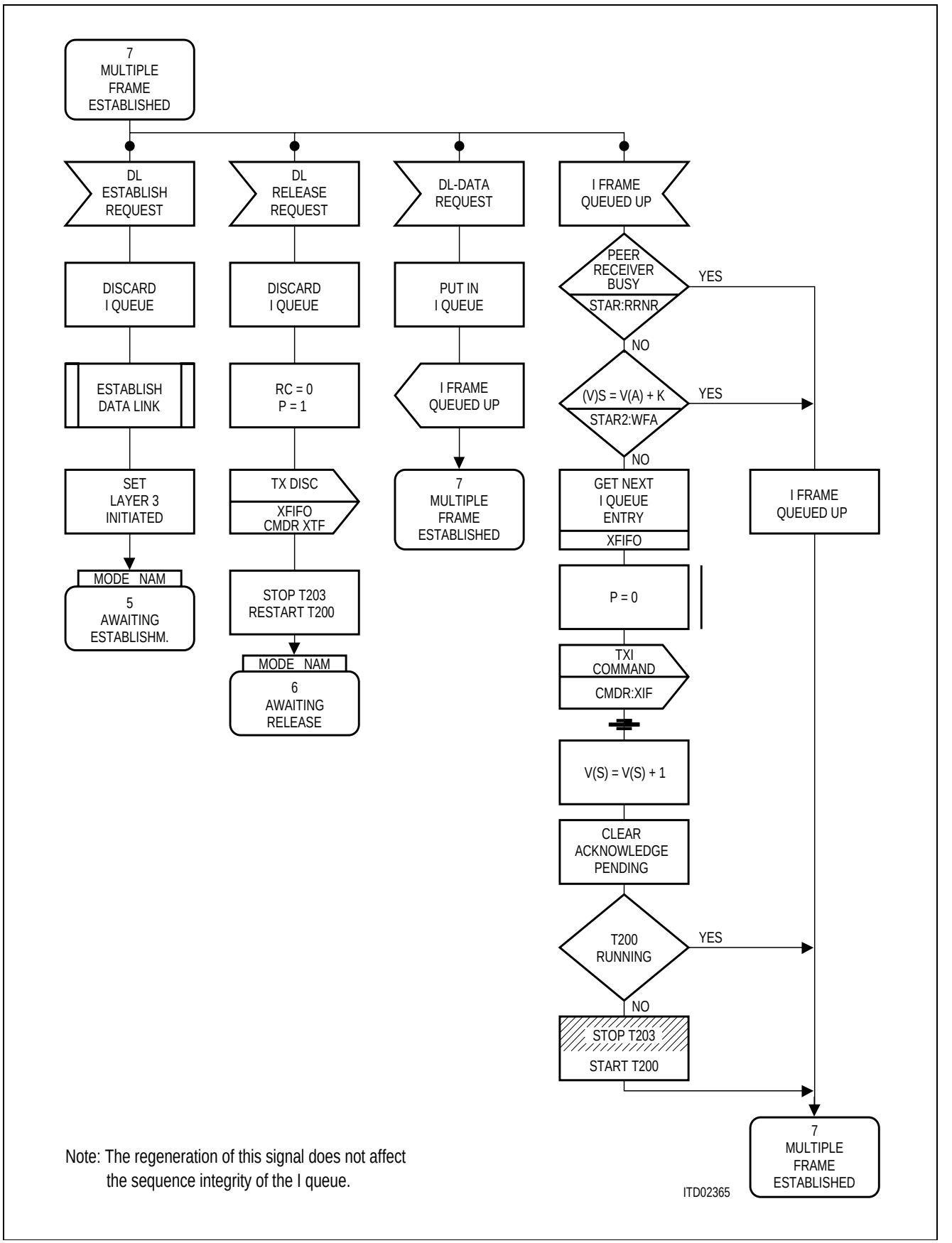

7
MULTIPLE
FRAME
ESTABLISHED
DL
ESTABLISH
REQUEST
DISCARD
I QUEUE
ESTABLISH
DATA LINK
SET
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
DL
RELEASE
REQUEST
DISCARD
I QUEUE
RC = 0
P = 1
TX DISC
XFIFO
CMDR XTF
STOP T203
RESTART T200
MODE NAM
6
AWAITING
RELEASE
DL-DATA
REQUEST
PUT IN
I QUEUE
I FRAME
QUEUED UP
7
MULTIPLE
FRAME
ESTABLISHED
I FRAME
QUEUED UP
PEER
RECEIVER
BUSY
STAR:RRNR
YES
NO
(V)S = V(A) + K
STAR2:WFA
YES
NO
GET NEXT
I QUEUE
ENTRY
XFIFO
P = 0
TXI
COMMAND
CMDR:XIF
V(S) = V(S) + 1
CLEAR
ACKNOWLEDGE
PENDING
T200
RUNNING
YES
NO
STOP T203
START T200
I FRAME
QUEUED UP
7
MULTIPLE
FRAME
ESTABLISHED
Note: The regeneration of this signal does not affect
the sequence integrity of the I queue.
ITD02365

Figure 39a

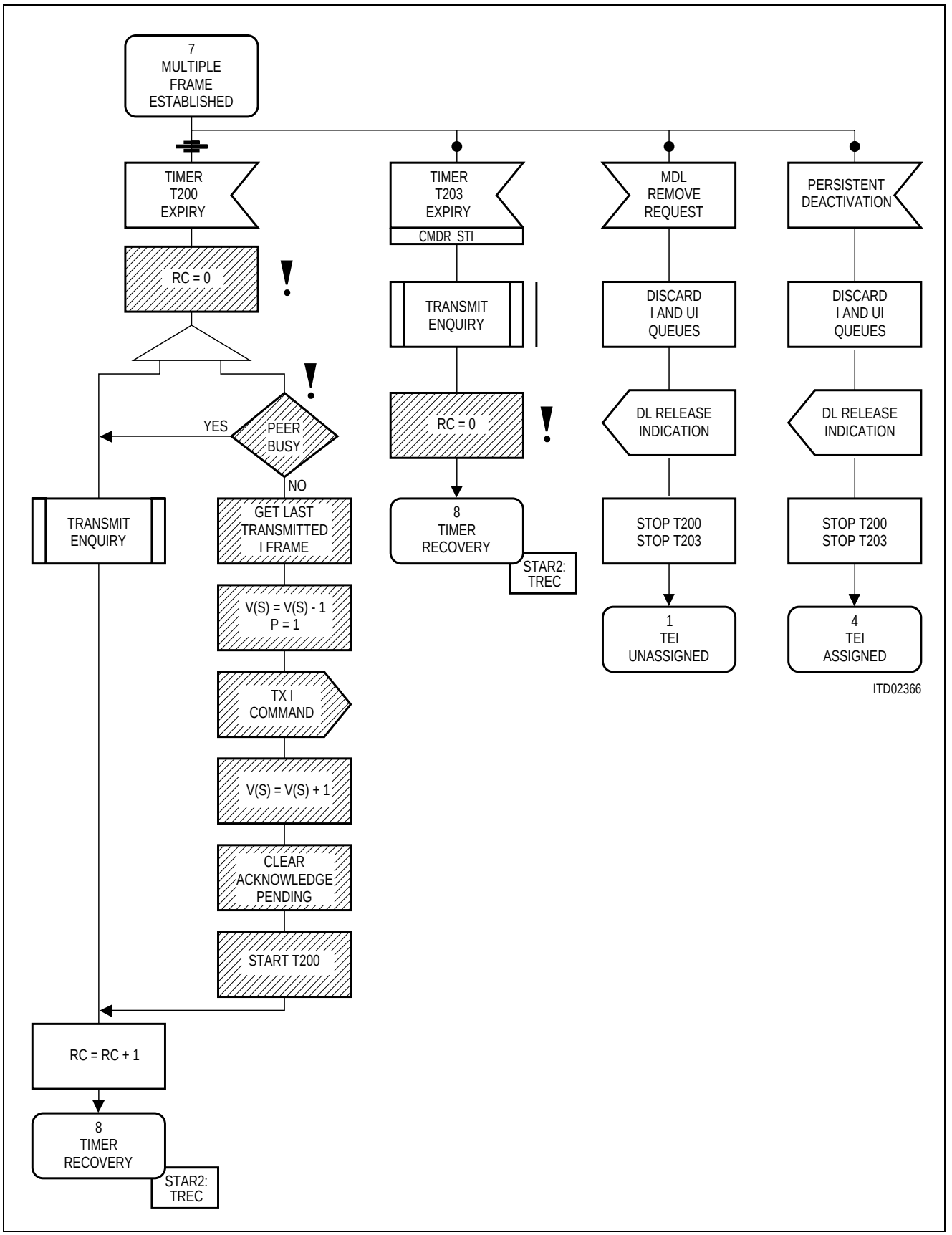

7
MULTIPLE
FRAME
ESTABLISHED
TIMER
T200
EXPIRY
RC = 0
!
YES
PEER
BUSY
!
NO
TRANSMIT
ENQUIRY
GET LAST
TRANSMITTED
I FRAME
V(S) = V(S) - 1
P = 1
TX I
COMMAND
V(S) = V(S) + 1
CLEAR
ACKNOWLEDGE
PENDING
START T200
RC = RC + 1
8
TIMER
RECOVERY
STAR2:
TREC
TIMER
T203
EXPIRY
CMDR STI
TRANSMIT
ENQUIRY
RC = 0
!
8
TIMER
RECOVERY
STAR2:
TREC
MDL
REMOVE
REQUEST
DISCARD
I AND UI
QUEUES
DL RELEASE
INDICATION
STOP T200
STOP T203
1
TEI
UNASSIGNED
PERSISTENT
DEACTIVATION
DISCARD
I AND UI
QUEUES
DL RELEASE
INDICATION
STOP T200
STOP T203
4
TEI
ASSIGNED
ITD02366

Figure 39b

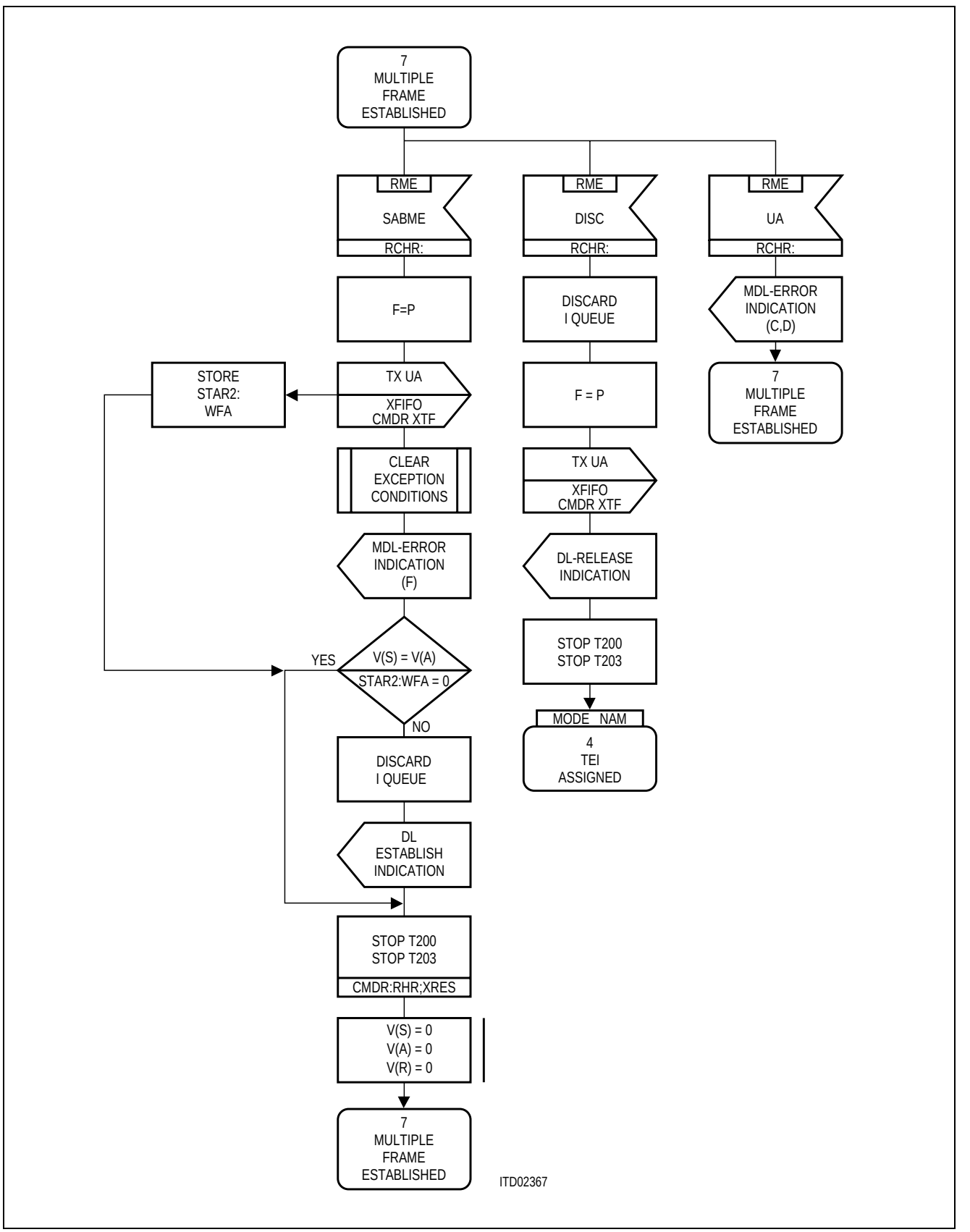

7
MULTIPLE
FRAME
ESTABLISHED
RME
SABME
RCHR:
F=P
TX UA
XFIFO
CMDR XTF
STORE
STAR2:
WFA
CLEAR
EXCEPTION
CONDITIONS
MDL-ERROR
INDICATION
(F)
V(S) = V(A)
STAR2:WFA = 0
YES
NO
DISCARD
I QUEUE
DL
ESTABLISH
INDICATION
STOP T200
STOP T203
CMDR:RHR;XRES
V(S) = 0
V(A) = 0
V(R) = 0
7
MULTIPLE
FRAME
ESTABLISHED
RME
DISC
RCHR:
DISCARD
I QUEUE
F = P
TX UA
XFIFO
CMDR XTF
DL-RELEASE
INDICATION
STOP T200
STOP T203
MODE NAM
4
TEI
ASSIGNED
RME
UA
RCHR:
MDL-ERROR
INDICATION
(C,D)
7
MULTIPLE
FRAME
ESTABLISHED
ITD02367

Figure 39c

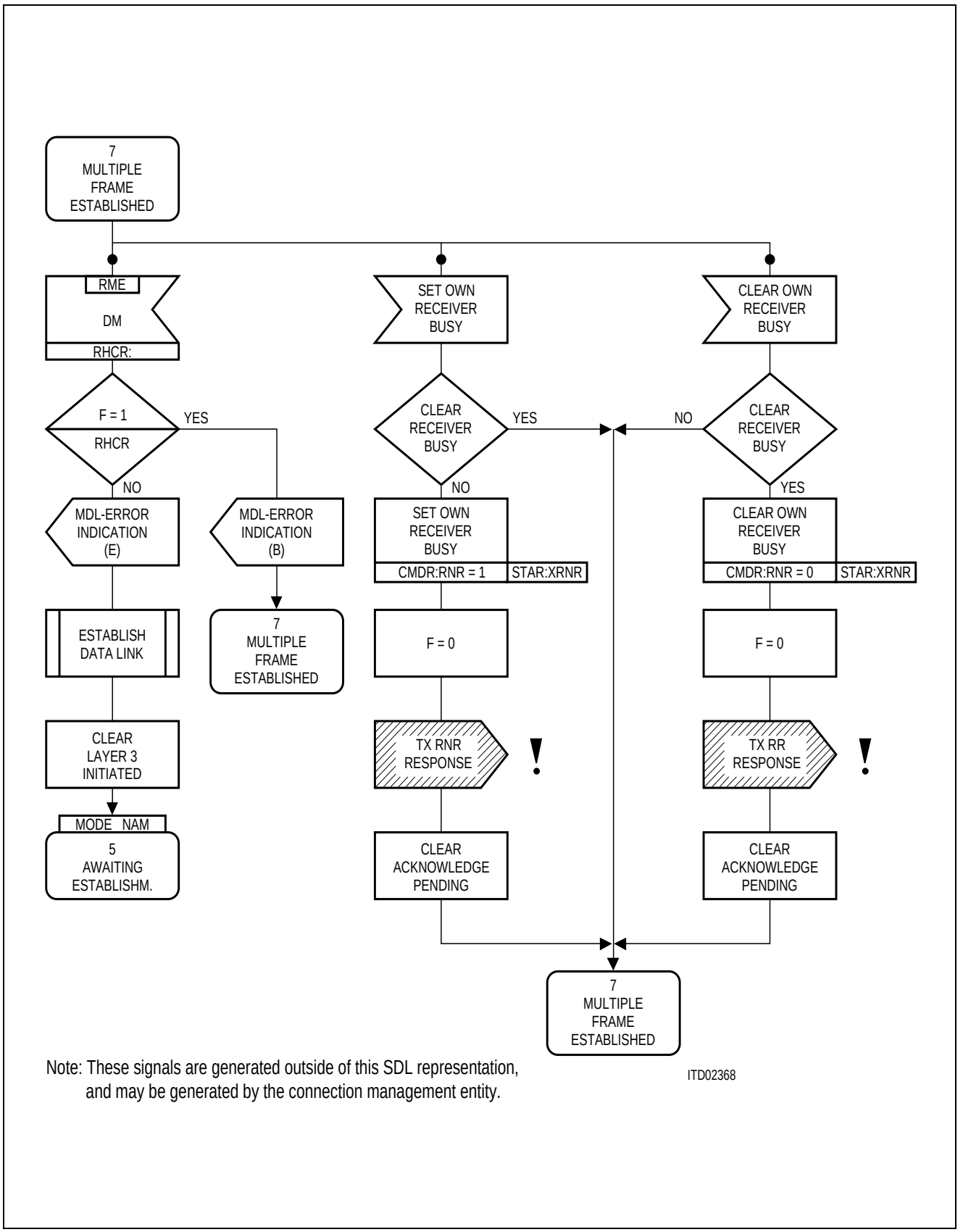

7
MULTIPLE
FRAME
ESTABLISHED
RME
DM
RHCR:
F = 1
RHCR
YES
NO
MDL-ERROR
INDICATION
(E)
MDL-ERROR
INDICATION
(B)
ESTABLISH
DATA LINK
7
MULTIPLE
FRAME
ESTABLISHED
CLEAR
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
SET OWN
RECEIVER
BUSY
CLEAR
RECEIVER
BUSY
YES
NO
SET OWN
RECEIVER
BUSY
CMDR:RNR = 1
STAR:XRNR
F = 0
TX RNR
RESPONSE
!
CLEAR
ACKNOWLEDGE
PENDING
CLEAR OWN
RECEIVER
BUSY
CLEAR
RECEIVER
BUSY
NO
YES
CLEAR OWN
RECEIVER
BUSY
CMDR:RNR = 0
STAR:XRNR
F = 0
TX RR
RESPONSE
!
CLEAR
ACKNOWLEDGE
PENDING
7
MULTIPLE
FRAME
ESTABLISHED
Note: These signals are generated outside of this SDL representation,
and may be generated by the connection management entity.
ITD02368

Figure 39d

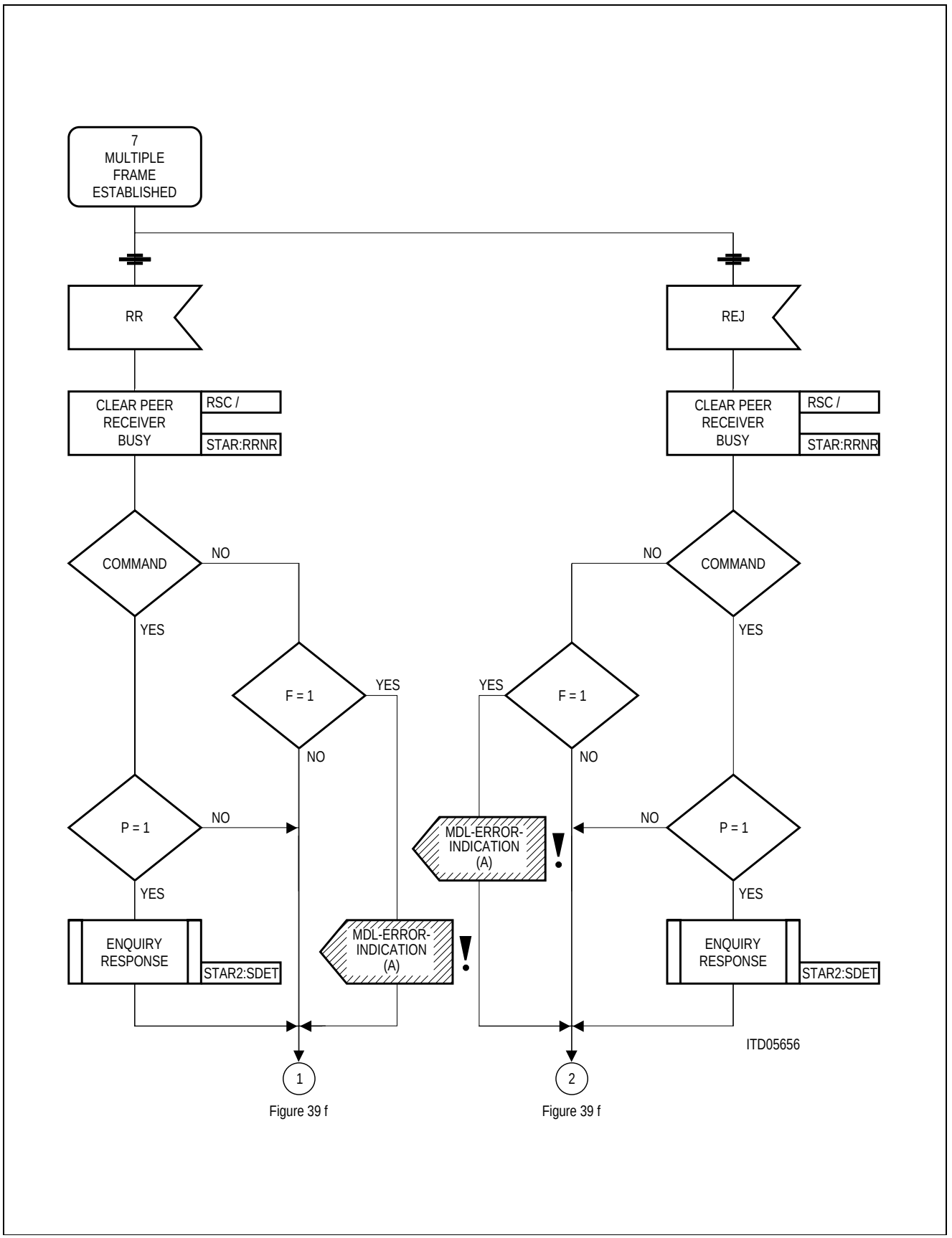


Figure 39 f

Figure 39 f

Figure 39e

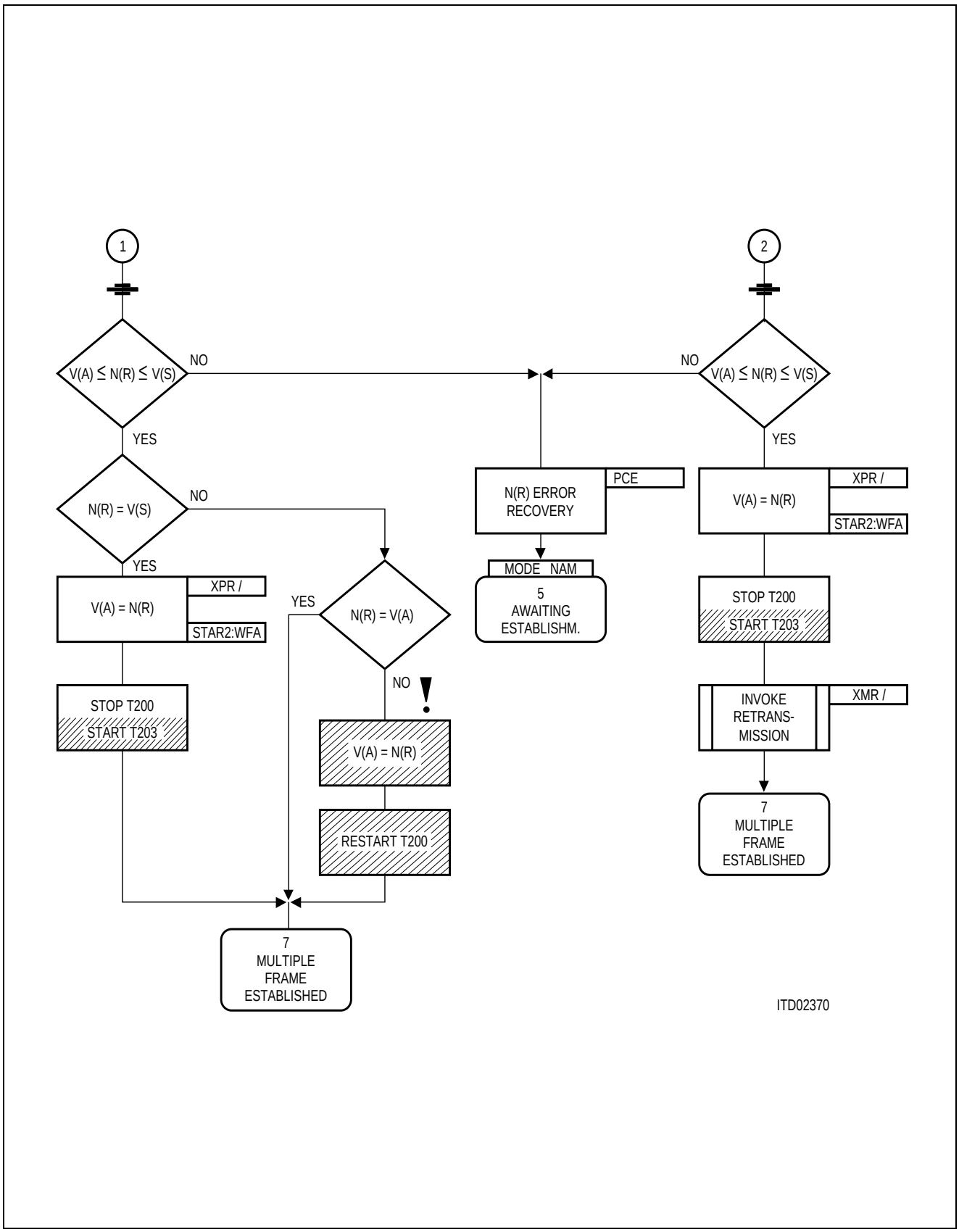

1
2
V(A) ≤ N(R) ≤ V(S)
NO
YES
N(R) = V(S)
NO
YES
V(A) = N(R)
XPR /
STAR2:WFA
STOP T200
START T203
YES
N(R) = V(A)
NO
!
V(A) = N(R)
RESTART T200
7
MULTIPLE
FRAME
ESTABLISHED
N(R) ERROR
RECOVERY
PCE
MODE NAM
5
AWAITING
ESTABLISHM.
NO
V(A) ≤ N(R) ≤ V(S)
YES
V(A) = N(R)
XPR /
STAR2:WFA
STOP T200
START T203
INVOKE
RETRANS-
MISSION
XMR /
7
MULTIPLE
FRAME
ESTABLISHED
ITD02370

Figure 39f

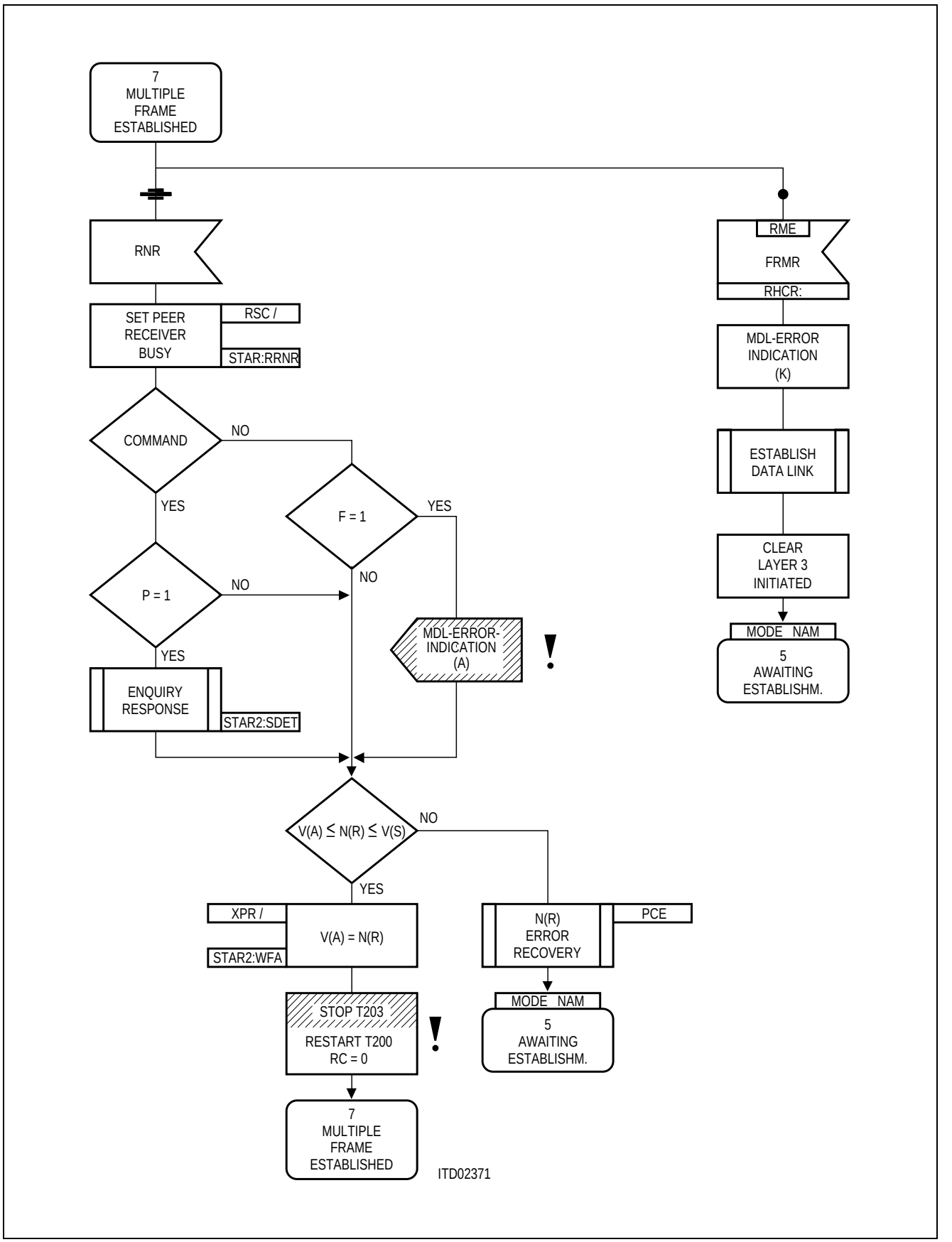
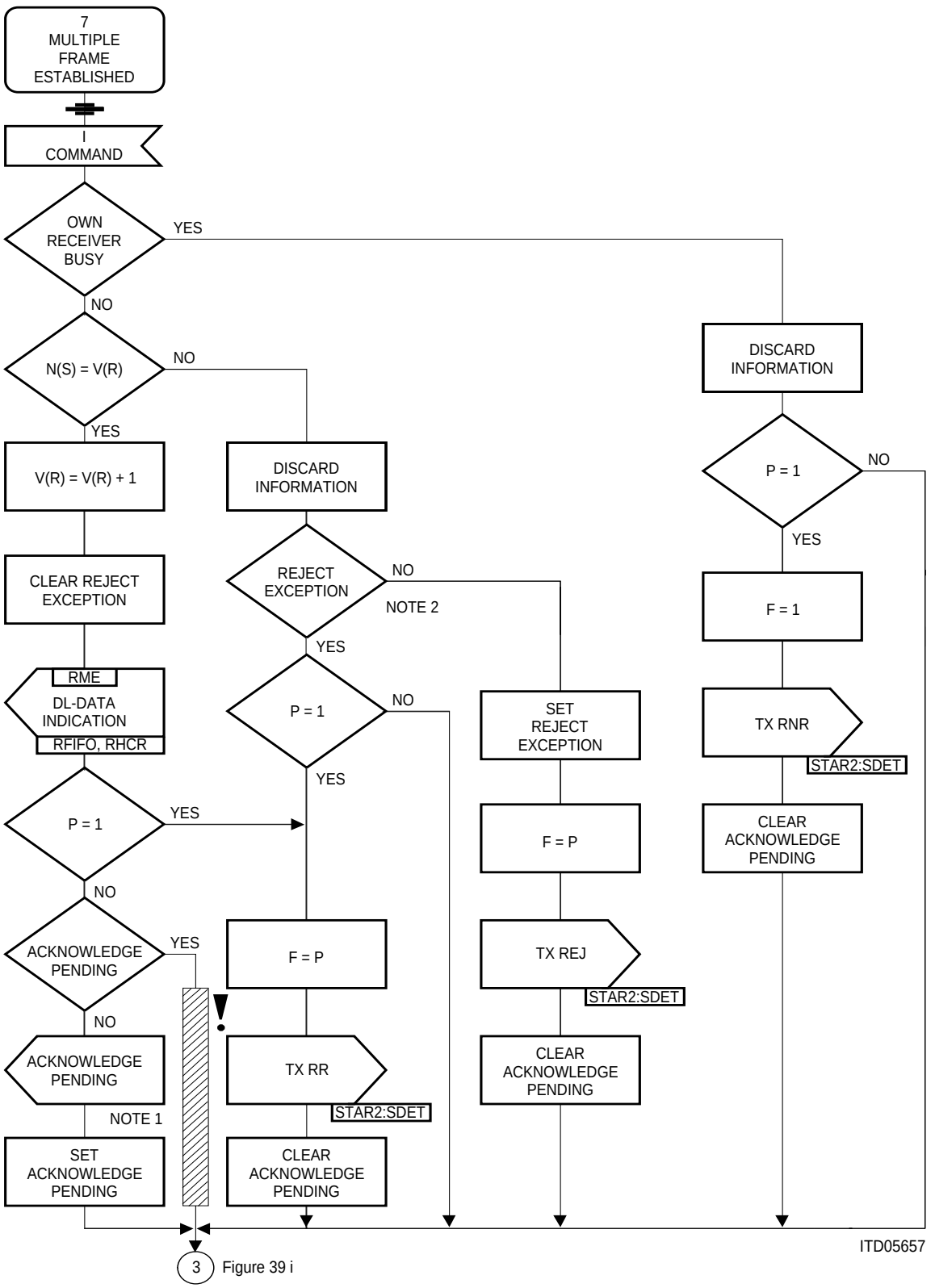

7
MULTIPLE
FRAME
ESTABLISHED
RNR
SET PEER
RECEIVER
BUSY
RSC /
STAR:RRNR
COMMAND
NO
YES
P = 1
NO
YES
ENQUIRY
RESPONSE
STAR2:SDET
F = 1
YES
NO
MDL-ERROR-
INDICATION
(A)
!
V(A) ≤ N(R) ≤ V(S)
NO
YES
XPR /
STAR2:WFA
V(A) = N(R)
STOP T203
RESTART T200
RC = 0
!
7
MULTIPLE
FRAME
ESTABLISHED
N(R)
ERROR
RECOVERY
PCE
MODE NAM
5
AWAITING
ESTABLISHM.
RME
FRMR
RHCR:
MDL-ERROR
INDICATION
(K)
ESTABLISH
DATA LINK
CLEAR
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
ITD02371

Figure 39g



Note 1: Processing of acknowledge pending is descripted on figure 39 i

Note 2: This SDL representation does not include the optional procedure in Appendix I.

Figure 39h

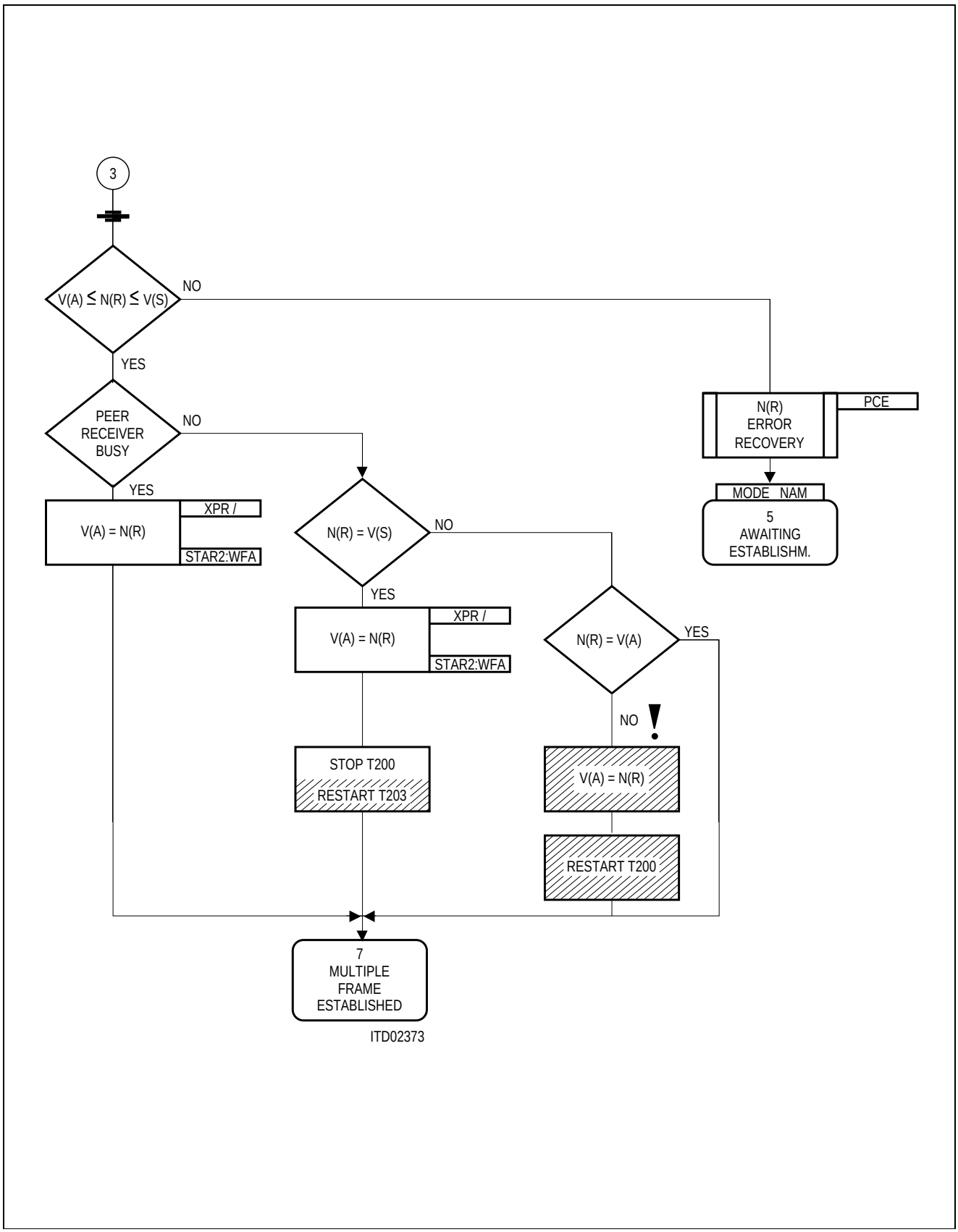

3
V(A) ≤ N(R) ≤ V(S)
NO
YES
PEER RECEIVER BUSY
NO
YES
V(A) = N(R)
XPR /
STAR2:WFA
N(R) = V(S)
NO
YES
V(A) = N(R)
XPR /
STAR2:WFA
N(R) = V(A)
YES
NO
STOP T200
RESTART T203
V(A) = N(R)
RESTART T200
N(R) ERROR RECOVERY
PCE
MODE NAM
5
AWAITING ESTABLISHM.
7
MULTIPLE FRAME ESTABLISHED
ITD02373

Figure 39i

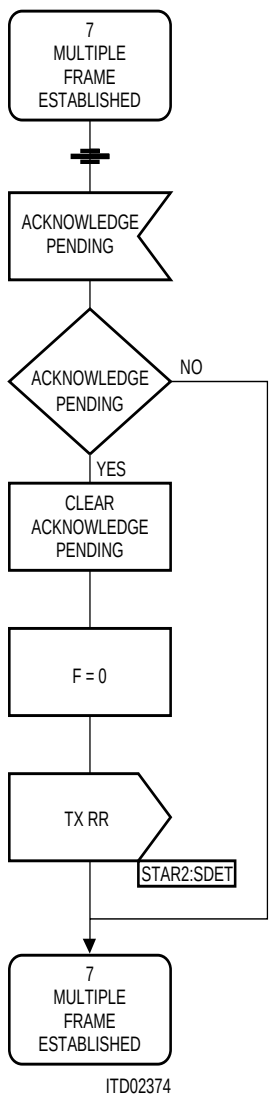

7
MULTIPLE
FRAME
ESTABLISHED
ACKNOWLEDGE
PENDING
ACKNOWLEDGE
PENDING
NO
YES
CLEAR
ACKNOWLEDGE
PENDING
F = 0
TX RR
STAR2:SDET
7
MULTIPLE
FRAME
ESTABLISHED
ITD02374

Figure 39j

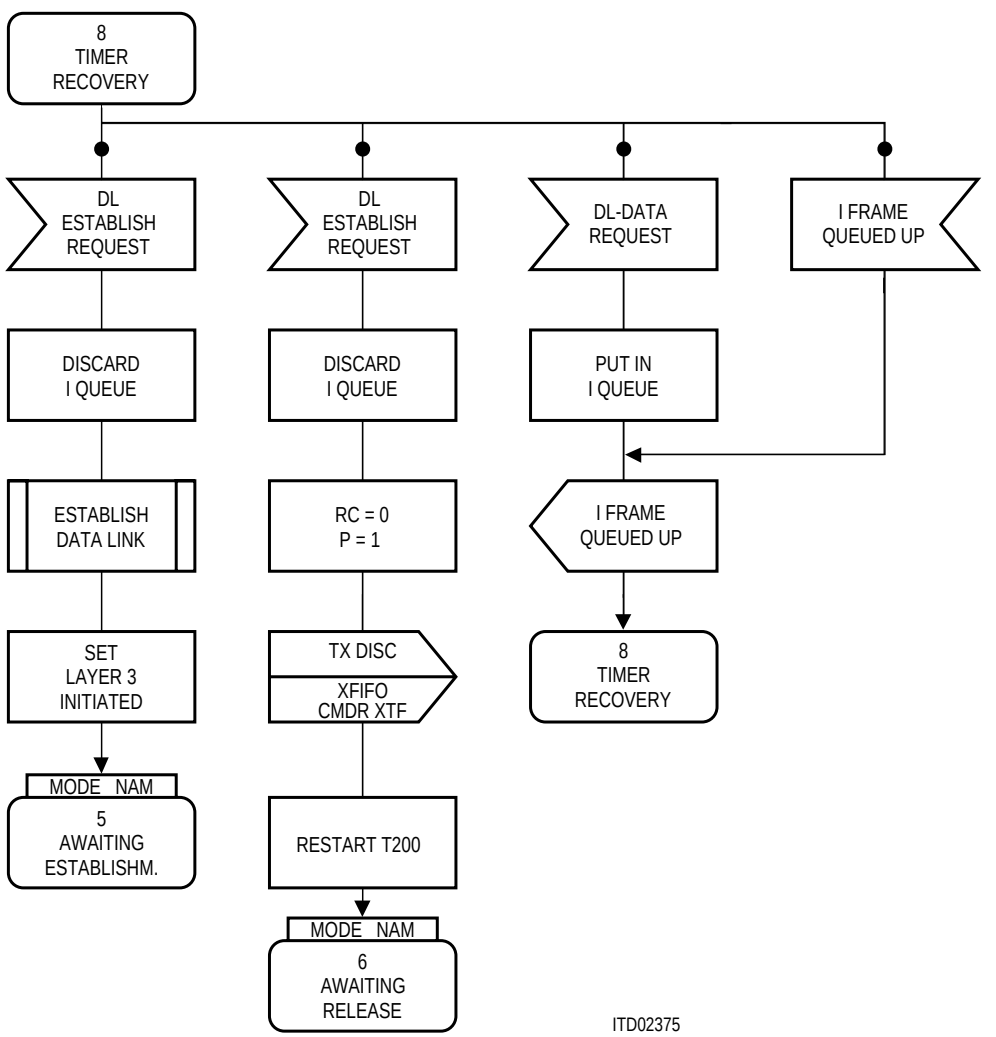

8
TIMER
RECOVERY
DL
ESTABLISH
REQUEST
DL
ESTABLISH
REQUEST
DL-DATA
REQUEST
I FRAME
QUEUED UP
DISCARD
I QUEUE
DISCARD
I QUEUE
PUT IN
I QUEUE
ESTABLISH
DATA LINK
RC = 0
P = 1
I FRAME
QUEUED UP
SET
LAYER 3
INITIATED
TX DISC
XFIFO
CMDR XTF
8
TIMER
RECOVERY
MODE NAM
5
AWAITING
ESTABLISHM.
RESTART T200
MODE NAM
6
AWAITING
RELEASE
ITD02375

Figure 40a

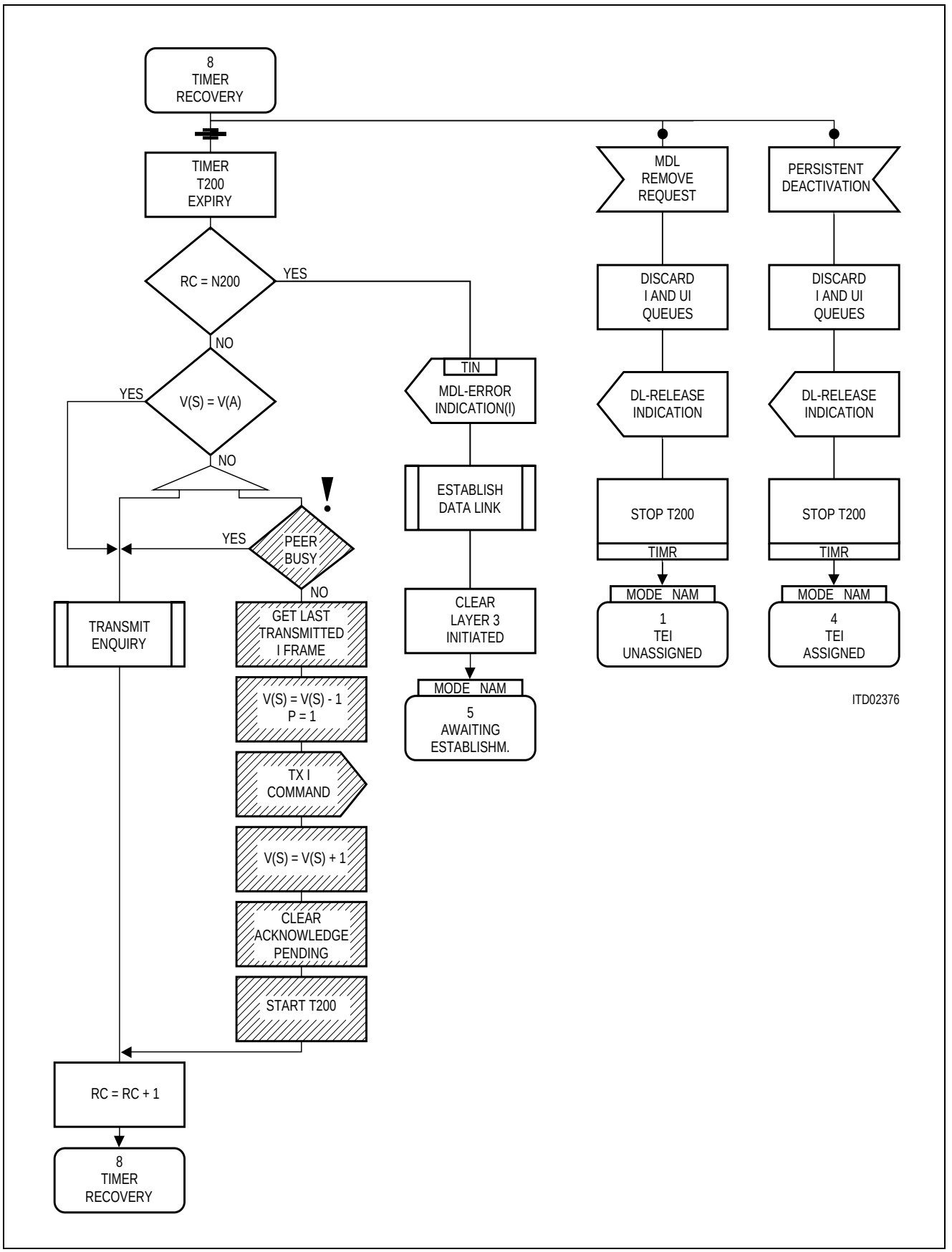

8
TIMER
RECOVERY
TIMER
T200
EXPIRY
RC = N200
YES
NO
V(S) = V(A)
YES
NO
PEER
BUSY
YES
NO
TRANSMIT
ENQUIRY
GET LAST
TRANSMITTED
I FRAME
V(S) = V(S) - 1
P = 1
TX I
COMMAND
V(S) = V(S) + 1
CLEAR
ACKNOWLEDGE
PENDING
START T200
RC = RC + 1
8
TIMER
RECOVERY
TIN
MDL-ERROR
INDICATION(I)
ESTABLISH
DATA LINK
CLEAR
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
MDL
REMOVE
REQUEST
DISCARD
I AND UI
QUEUES
DL-RELEASE
INDICATION
STOP T200
TIMR
MODE NAM
1
TEI
UNASSIGNED
PERSISTENT
DEACTIVATION
DISCARD
I AND UI
QUEUES
DL-RELEASE
INDICATION
STOP T200
TIMR
MODE NAM
4
TEI
ASSIGNED
ITD02376

Figure 40b

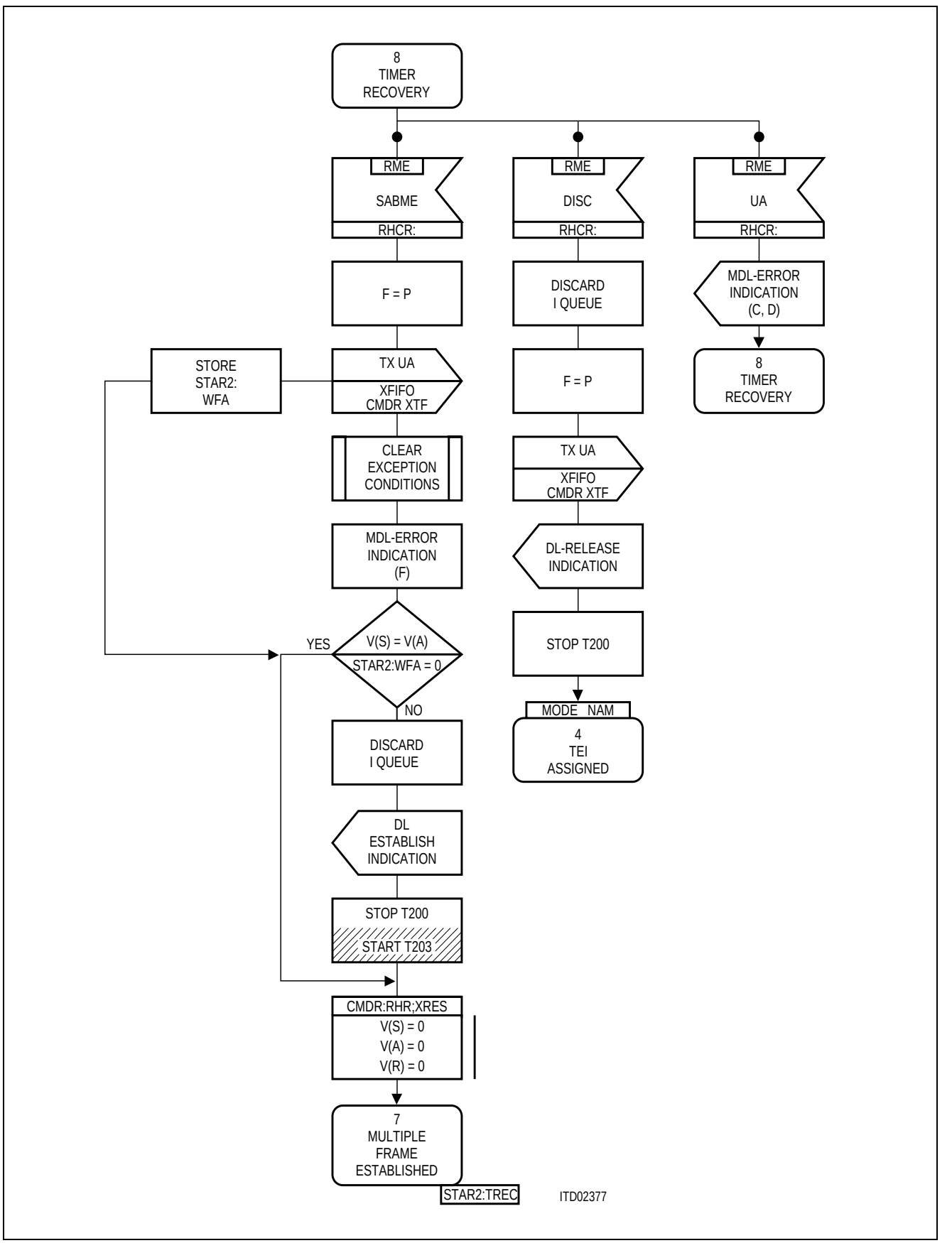

8
TIMER
RECOVERY
RME
SABME
RHCR:
F = P
TX UA
XFIFO
CMDR XTF
STORE
STAR2:
WFA
CLEAR
EXCEPTION
CONDITIONS
MDL-ERROR
INDICATION
(F)
YES
V(S) = V(A)
STAR2:WFA = 0
NO
DISCARD
I QUEUE
DL
ESTABLISH
INDICATION
STOP T200
START T203
CMDR:RHR;XRES
V(S) = 0
V(A) = 0
V(R) = 0
7
MULTIPLE
FRAME
ESTABLISHED
STAR2:TREC
RME
DISC
RHCR:
DISCARD
I QUEUE
F = P
TX UA
XFIFO
CMDR XTF
DL-RELEASE
INDICATION
STOP T200
MODE NAM
4
TEI
ASSIGNED
RME
UA
RHCR:
MDL-ERROR
INDICATION
(C, D)
8
TIMER
RECOVERY
ITD02377

Figure 40c

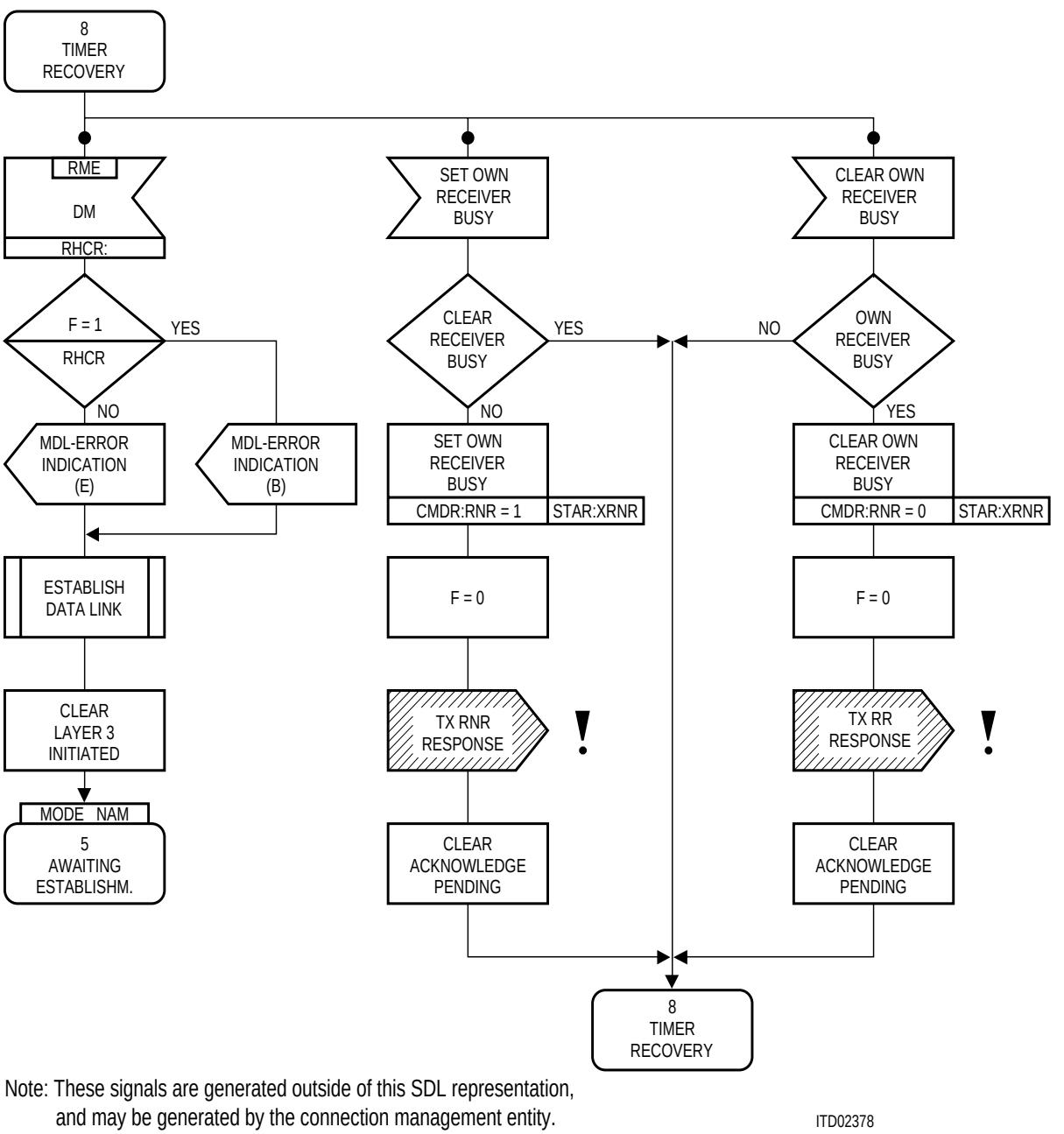

8
TIMER
RECOVERY
RME
DM
RHCR:
F = 1
RHCR
YES
NO
MDL-ERROR
INDICATION
(E)
MDL-ERROR
INDICATION
(B)
ESTABLISH
DATA LINK
CLEAR
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
SET OWN
RECEIVER
BUSY
CLEAR
RECEIVER
BUSY
YES
NO
SET OWN
RECEIVER
BUSY
CMDR:RNR = 1
STAR:XRNR
F = 0
TX RNR
RESPONSE
!
CLEAR
ACKNOWLEDGE
PENDING
CLEAR OWN
RECEIVER
BUSY
OWN
RECEIVER
BUSY
NO
YES
CLEAR OWN
RECEIVER
BUSY
CMDR:RNR = 0
STAR:XRNR
F = 0
TX RR
RESPONSE
!
CLEAR
ACKNOWLEDGE
PENDING
8
TIMER
RECOVERY
Note: These signals are generated outside of this SDL representation,
and may be generated by the connection management entity.
ITD02378

Figure 40d

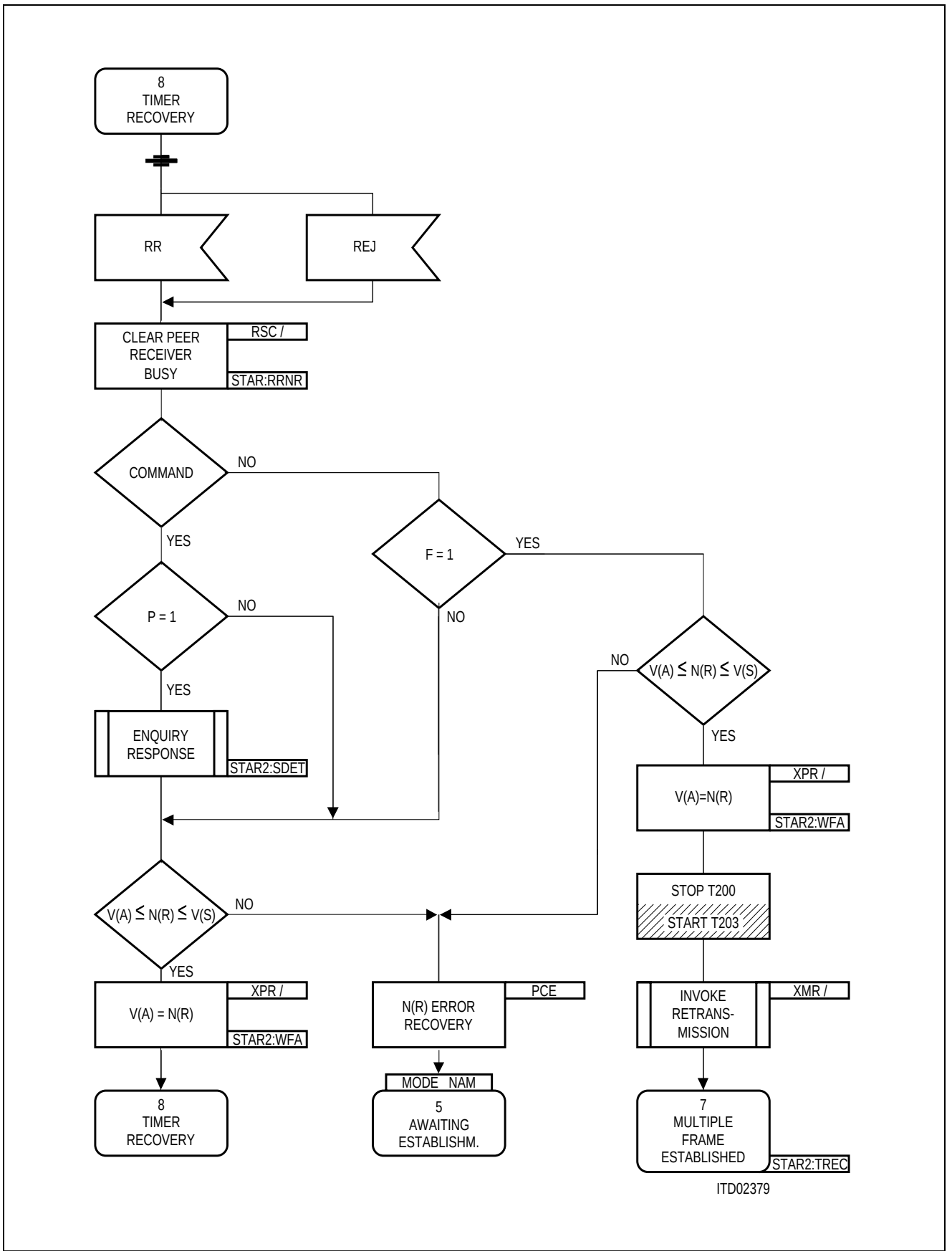

8
TIMER
RECOVERY
RR
REJ
CLEAR PEER
RECEIVER
BUSY
RSC /
STAR:RRNR
COMMAND
NO
YES
P = 1
NO
YES
ENQUIRY
RESPONSE
STAR2:SDET
F = 1
YES
NO
V(A) ≤ N(R) ≤ V(S)
NO
YES
V(A)=N(R)
XPR /
STAR2:WFA
STOP T200
START T203
V(A) ≤ N(R) ≤ V(S)
NO
YES
V(A) = N(R)
XPR /
STAR2:WFA
N(R) ERROR
RECOVERY
PCE
INVOKE
RETRANS-
MISSION
XMR /
8
TIMER
RECOVERY
MODE NAM
5
AWAITING
ESTABLISHM.
7
MULTIPLE
FRAME
ESTABLISHED
STAR2:TREC
ITD02379

Figure 40e

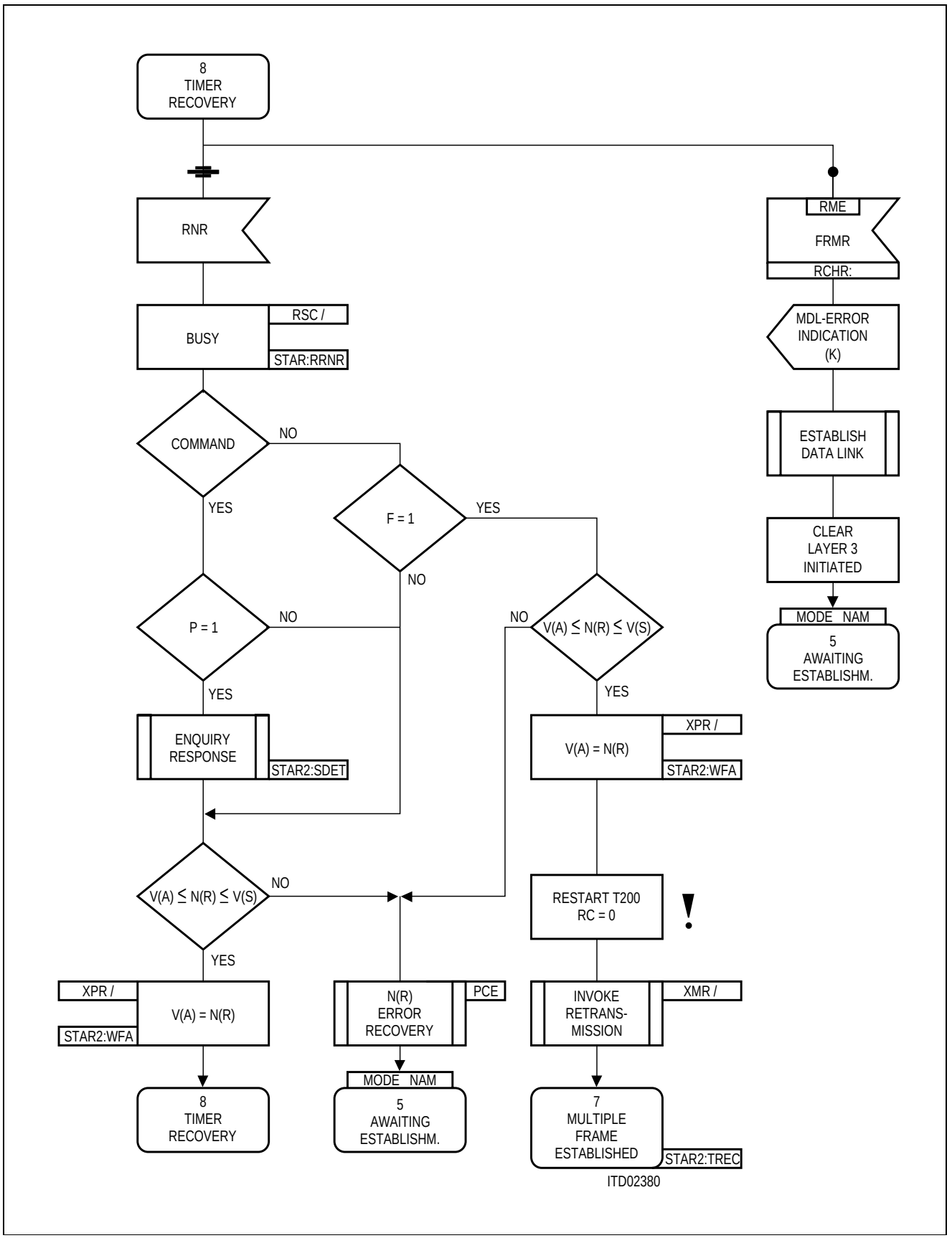
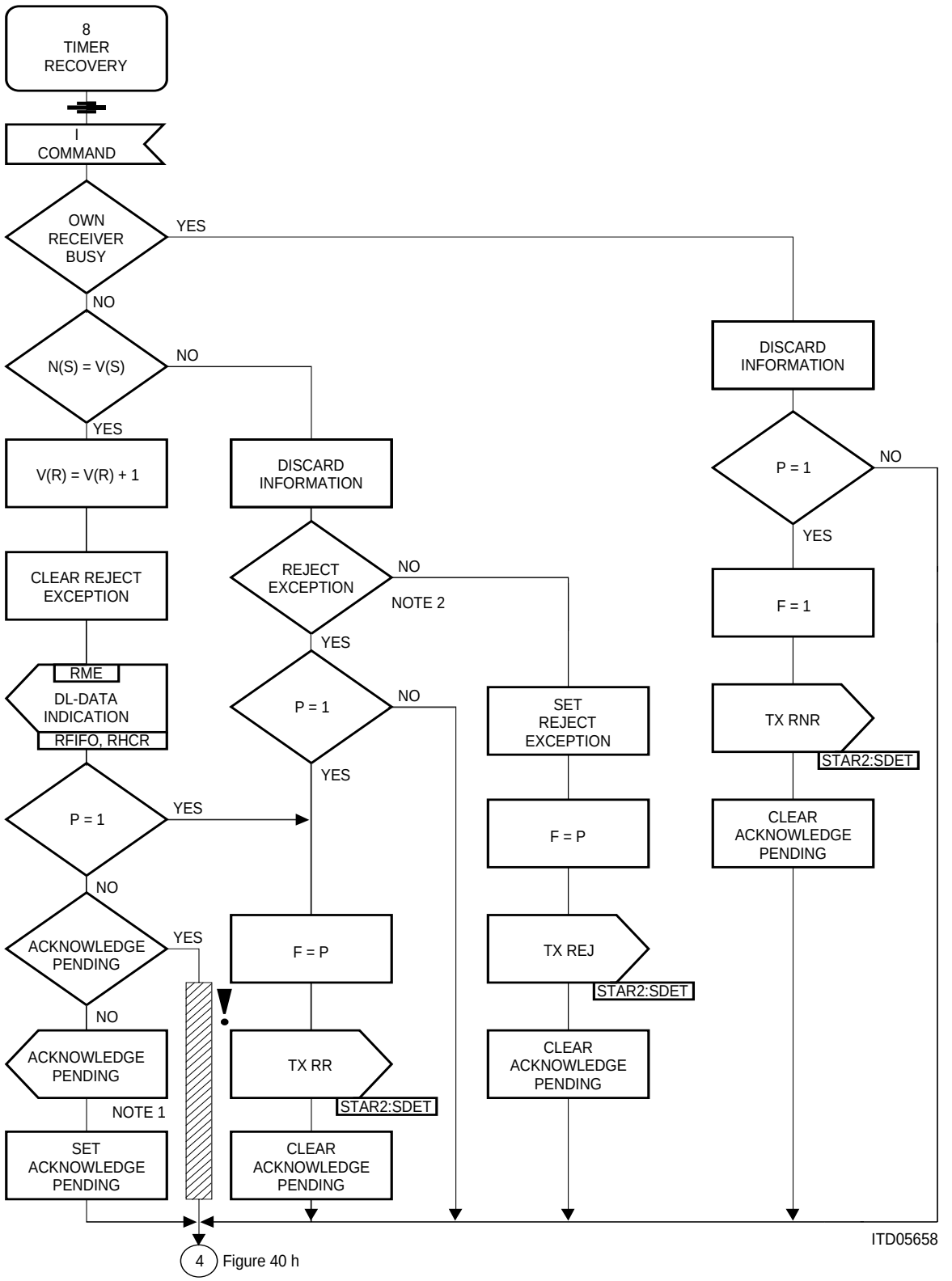

8
TIMER
RECOVERY
RNR
BUSY
RSC /
STAR:RRNR
COMMAND
NO
YES
P = 1
NO
YES
ENQUIRY
RESPONSE
STAR2:SDET
V(A) ≤ N(R) ≤ V(S)
NO
YES
XPR /
V(A) = N(R)
STAR2:WFA
8
TIMER
RECOVERY
F = 1
YES
NO
V(A) ≤ N(R) ≤ V(S)
NO
YES
RESTART T200
RC = 0
!
N(R)
ERROR
RECOVERY
PCE
MODE NAM
5
AWAITING
ESTABLISHM.
INVOKE
RETRANS-
MISSION
XMR /
7
MULTIPLE
FRAME
ESTABLISHED
STAR2:TREC
RME
FRMR
RCHR:
MDL-ERROR
INDICATION
(K)
ESTABLISH
DATA LINK
CLEAR
LAYER 3
INITIATED
MODE NAM
5
AWAITING
ESTABLISHM.
ITD02380

Figure 40f



Note 1: Processing of acknowledge pending is descripted on figure 40 i

Note 2: This SDL representation does not include the optional procedure in Appendix I.

Figure 40g

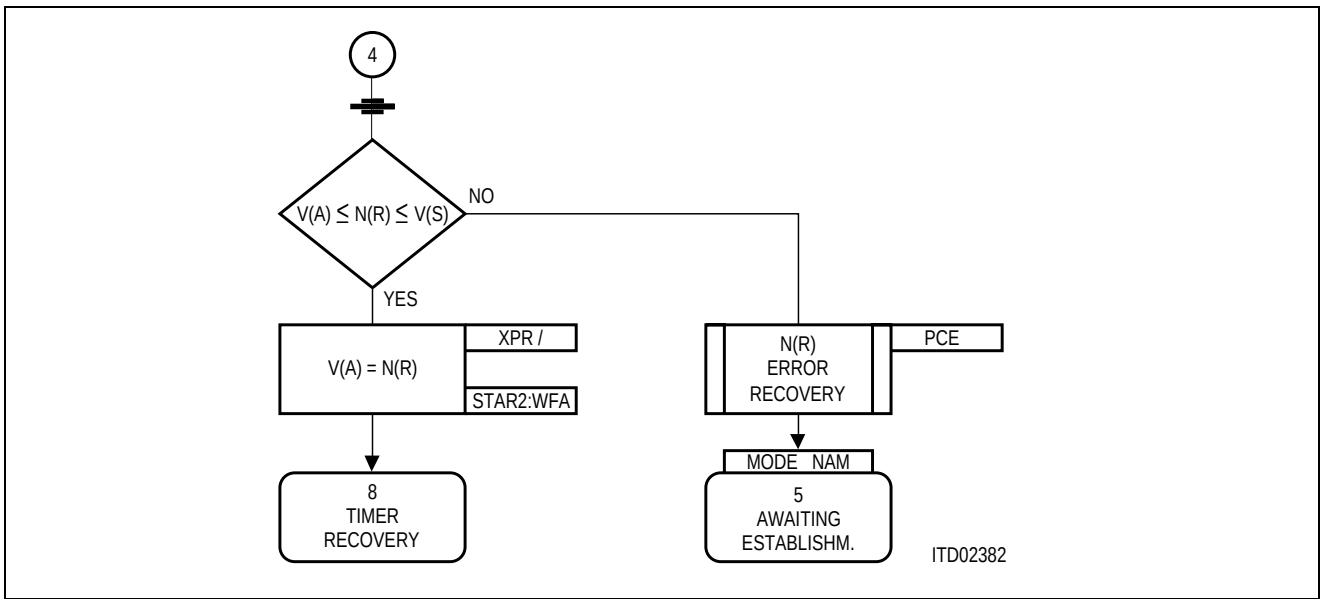


Figure 40h

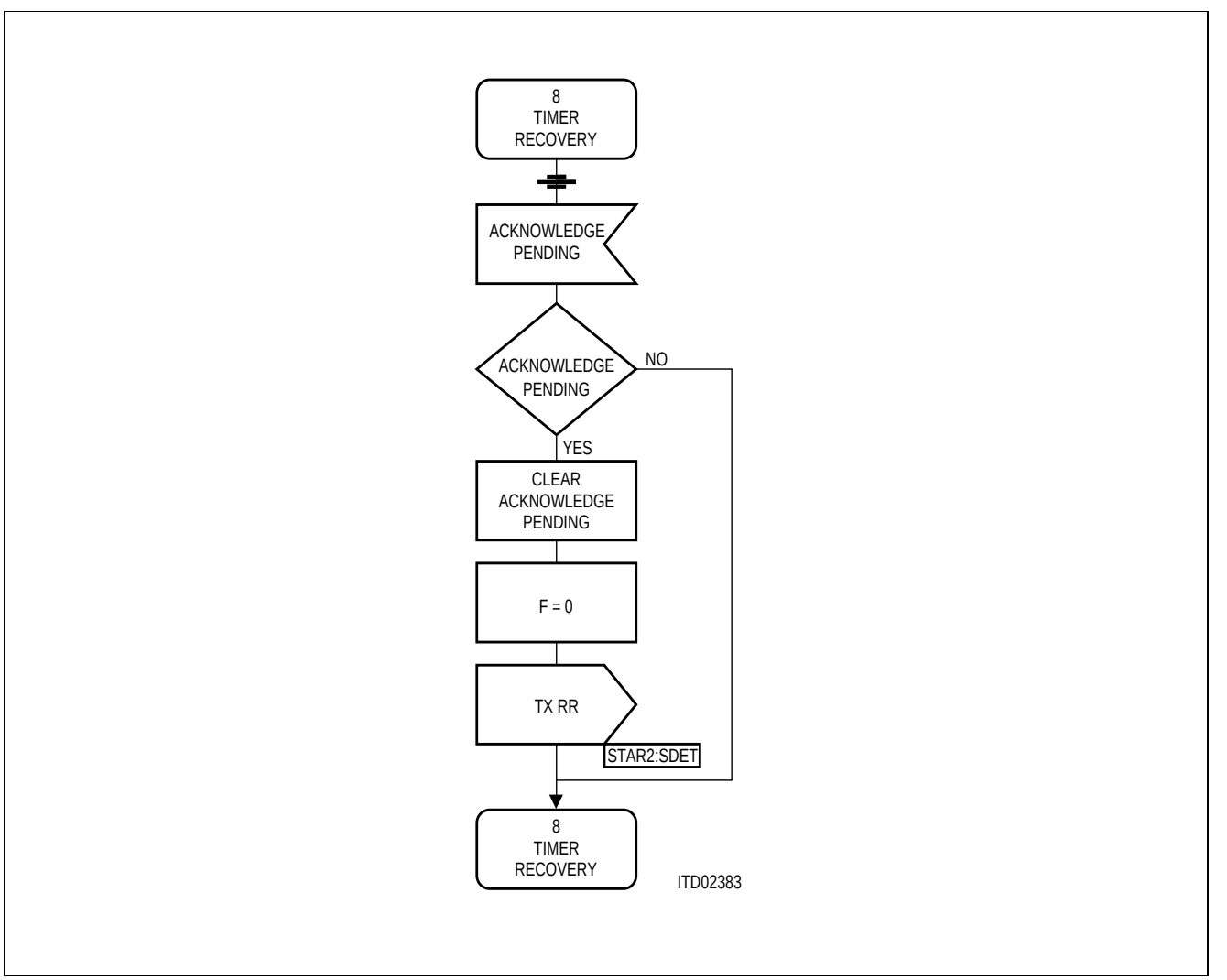
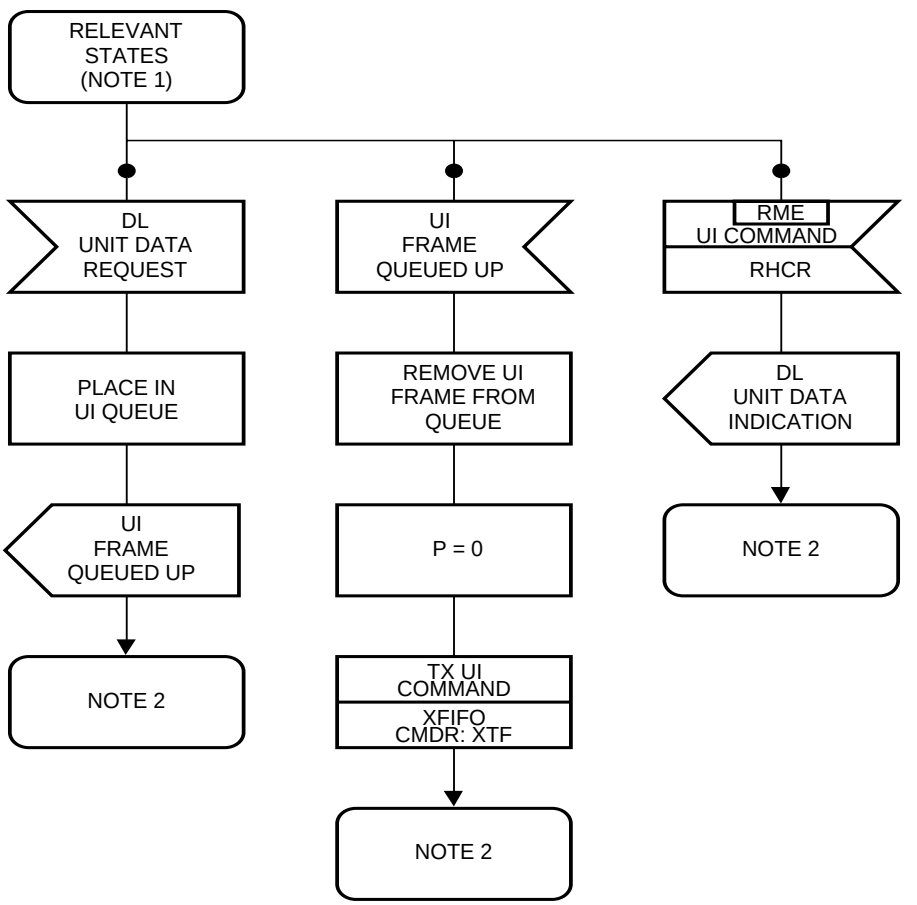


Figure 40i



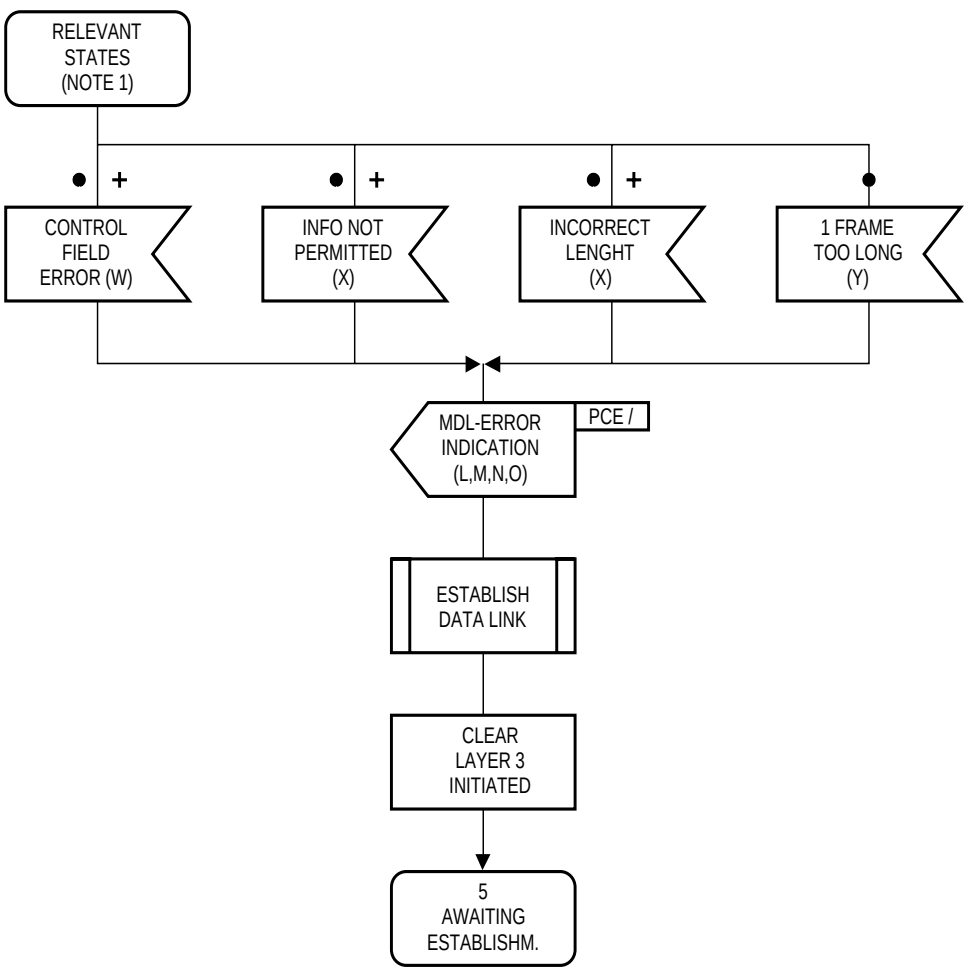
Note 1: The relevant states are as follows

4 TEI-assigned
5 Awaiting-establishement
6 Awaiting-release
7 Multiple-frame-established
8 Timer-recovery

Note 2: The data link layer returns to the state it was in prior to the events shown.

ITD02384

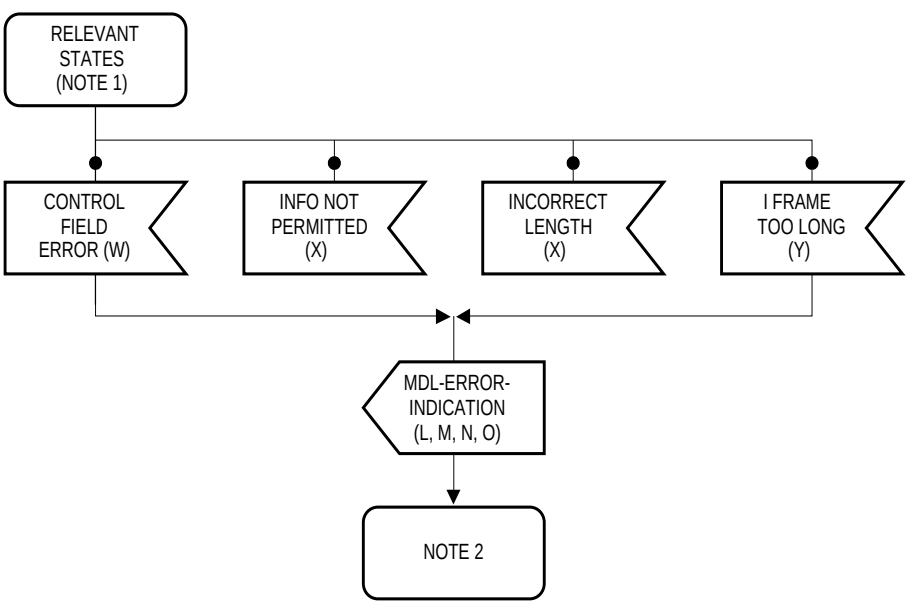
Figure 41a



Note 1: The relevant states are as follows
7 Multiple-frame-established
8 Timer-recovery

ITD02385

Figure 41b



Note 1: The relevant states are as follows:
4 TEI-assigned
5 Awaiting-establishment
6 Awaiting-release

Note 2: The data link layer returns to the state it was in prior to the events shown

ITD02577

Figure 41c

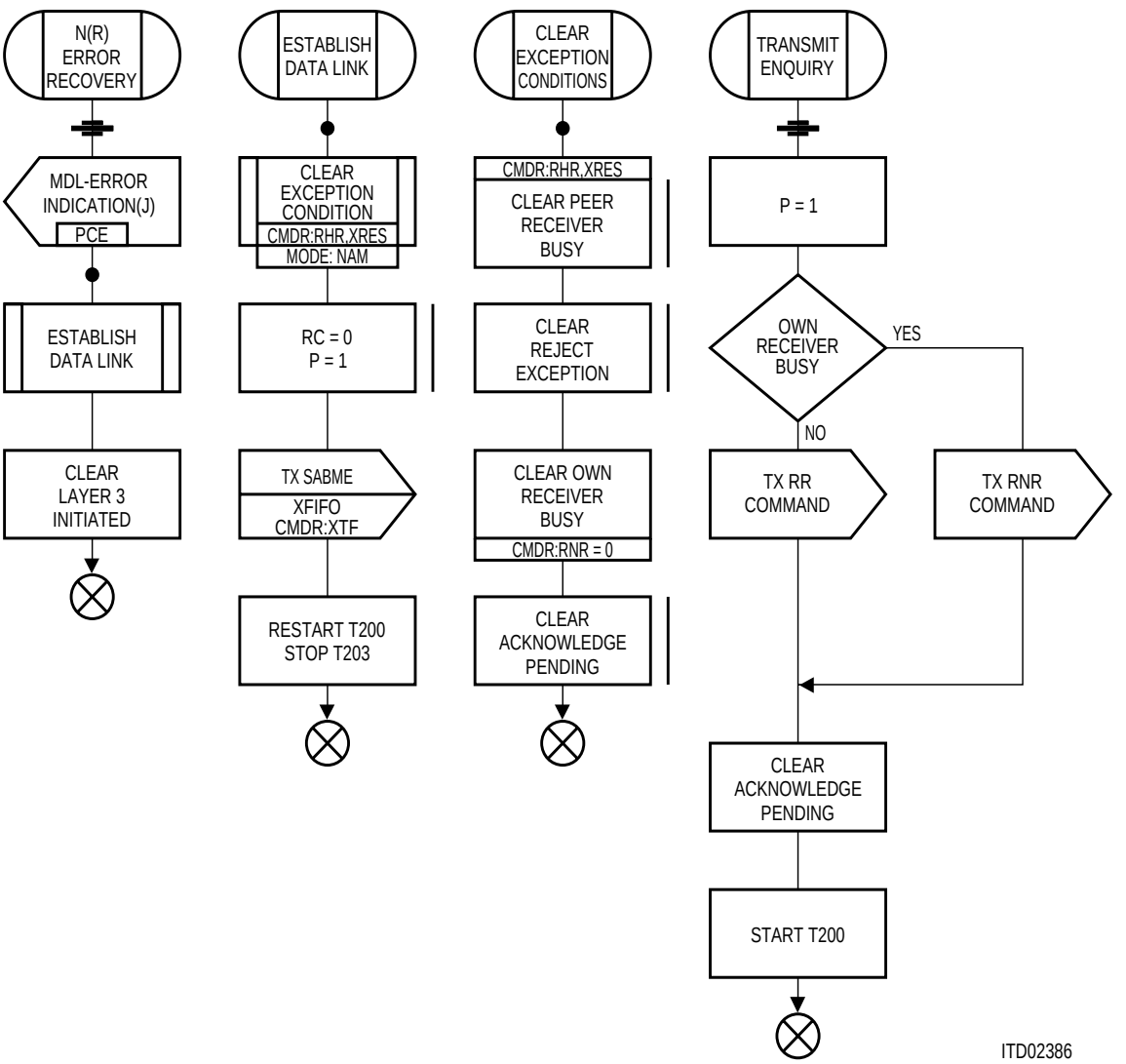
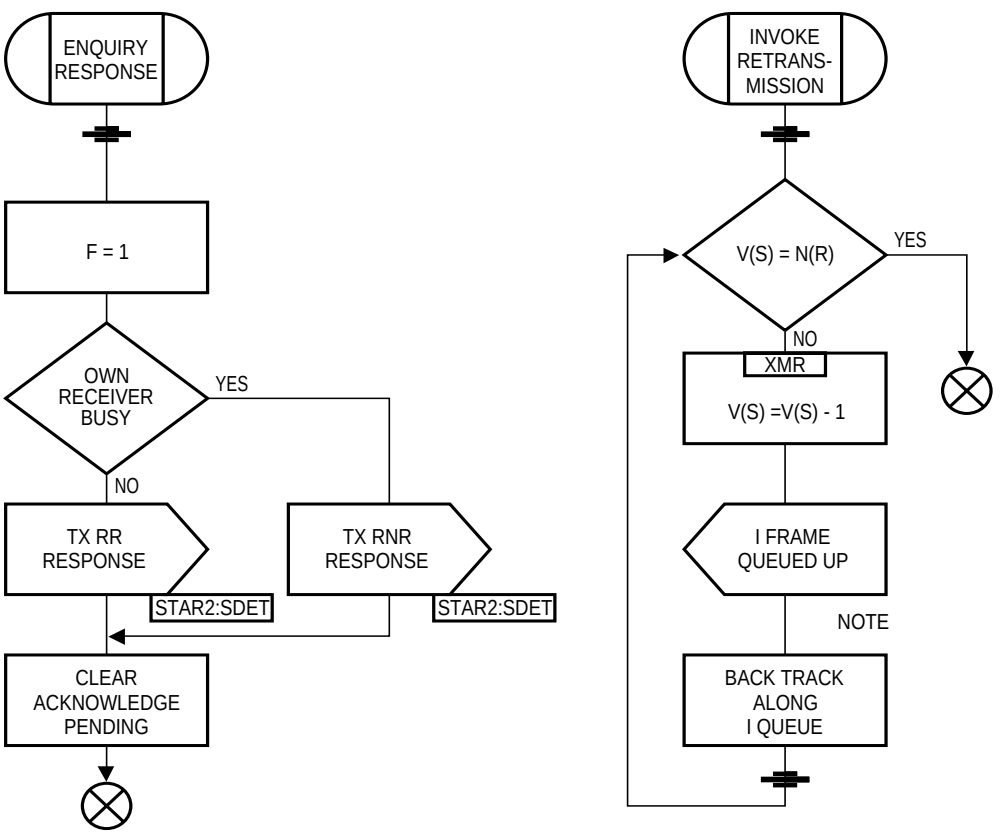

N(R)
ERROR
RECOVERY
MDL-ERROR
INDICATION(J)
PCE
ESTABLISH
DATA LINK
CLEAR
LAYER 3
INITIATED
ESTABLISH
DATA LINK
CLEAR
EXCEPTION
CONDITION
CMDR:RHR,XRES
MODE: NAM
RC = 0
P = 1
TX SABME
XFIFO
CMDR:XTF
RESTART T200
STOP T203
CLEAR
EXCEPTION
CONDITIONS
CMDR:RHR,XRES
CLEAR PEER
RECEIVER
BUSY
CLEAR
REJECT
EXCEPTION
CLEAR OWN
RECEIVER
BUSY
CMDR:RNR = 0
CLEAR
ACKNOWLEDGE
PENDING
TRANSMIT
ENQUIRY
P = 1
OWN
RECEIVER
BUSY
YES
NO
TX RR
COMMAND
TX RNR
COMMAND
CLEAR
ACKNOWLEDGE
PENDING
START T200
ITD02386

Figure 41d



Note: The generation of the correct number of signals in order to cause the required retransmission of I frames does not alter their sequence integrity.

ITD02387

Figure 41e

3 Operational Description

The ISAC-S TE, designed for the connection of subscribers to an ISDN using a standard S/T interface, has the following application, corresponding to the operating mode explained in **chapter 2**:

Terminal Equipment TE1, TA

e.g. ISDN-feature telephone,
ISDN-voice/data workstation
Terminal Adapter for non-ISDN terminals (TE2)

3.1 Microprocessor Interface Operation

The ISAC-S TE is programmed via an 8-bit parallel microcontroller interface. Easy and fast microprocessor access is provided by 8-bit address decoding on the chip. Depending on the chip package (P-DIP-40, P-LCC-44, P-MQFP-64) either one or three types of µP buses are provided:

P-DIP-40 package:

The ISAC-S TE microcontroller interface is of the **Siemens/Intel multiplexed** address/data bus type with control signals **$\overline{\text{CS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$, ALE**.

P-LCC-44/P-MQFP-64 package:

The ISAC-S TE microcontroller interface can be selected to be either of the

(1) – Motorola type with control signals **$\overline{\text{CS}}$, R/$\overline{\text{W}}$, $\overline{\text{DS}}$**

(2) – Siemens/Intel non-multiplexed bus type with control signals **$\overline{\text{CS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$**

(3) – or of the **Siemens/Intel multiplexed** address/data bus type with control signals **$\overline{\text{CS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$, ALE**.

The selection is performed via pin ALE as follows:

ALE tied to V_{DD} ⇒ (1)

ALE tied to V_{SS} ⇒ (2)

Edge on ALE ⇒ (3).

The occurrence of an edge on ALE, either positive or negative, at any time during the operation immediately selects interface type (3). A return to one of the other interface types is possible only if a hardware reset is issued.

Notes: 1) If the multiplexed address/data bus type (3) is selected, the unused address pins A0-A5 are internally pulled low and may thus be left open. It is however recommended to tie the unused input pins to a V_{DD} voltage level.

2) If the non-multiplexed bus types (1) or (2) are selected, the EAW line can no longer be used since pin 10 EAW/A5 has the function of an address pin (PLCC-44 only).

The microprocessor interface signals are summarized in **table 9**.

Table 9
µP Interface of the ISAC®-S TE

Pin No. P-DIP-40	Pin No. P-LCC-44	Pin No. P-MQFP-64	Symbol	Input (I) Output (O) Open Drain (OD)	Function
37 38 39 40 1 2 3 4	41 42 43 44 1 2 3 4	37 38 39 40 41 42 43 44	AD0/D0 AD1/D1 AD2/D2 AD3/D3 AD4/D4 AD5/D5 AD6/D6 AD7/D7	I/O I/O I/O I/O I/O I/O I/O I/O	**Multiplexed Bus Mode:** Address/Data bus. Transfers addresses from the µP system to the ISAC-S TE and data between the µP system and the ISAC-S TE. **Non-Multiplexed Bus Mode:** Data bus. Transfers data between the µP system and the ISAC-S TE.
34	37	27	$\overline{CS}$	I	**Chip Select.** A 0 ("low") on this line selects the ISAC-S TE for a read/write operation.
	38	28	R/$\overline{W}$	I	**Read/Write.** A 1 ("high"), identifies a valid µP access as a read operation. A 0, identifies a valid µP access as a write operation (Motorola bus mode).
35	38	28	$\overline{WR}$	I	**Write.** This signal indicates a write operation (Siemens/Intel bus mode).
	39	29	$\overline{DS}$	I	**Data Strobe.** The rising edge marks the end of a valid read or write operation (Motorola bus mode).
36	39	29	$\overline{RD}$	I	**Read.** This signal indicates a read operation (Siemens/Intel bus mode).
20	23	8	$\overline{INT}$	OD	**Interrupt Request.** The signal is activated when the ISAC-S TE requests an interrupt. It is an open drain output.
33	36	26	ALE	I	**Address Latch Enable.** A high on this line indicates an address on the external address bus (multiplexed bus type only). ALE also selects interface mode.
	40	30	A0	I	**Address Bit 0** (non-multiplexed bus type).
	6	51	A1	I	**Address Bit 1** (non-multiplexed bus type).
	5	50	A2	I	**Address Bit 2** (non-multiplexed bus type).
	18	64	A3	I	**Address Bit 3** (non-multiplexed bus type).
	17	63	A4	I	**Address Bit 4** (non-multiplexed bus type).
	10	55	A5	I	**Address Bit 5** (non-multiplexed bus type).

3.2 Interrupt Structure and Logic

Since the ISAC-S TE provides only one interrupt request output ($\overline{INT}$), the cause of an interrupt is determined by the microprocessor by reading the Interrupt Status Register ISTA. In this register, seven interrupt sources can be directly read. The LSB of ISTA points to eight non-critical interrupt sources which are indicated in the Extended Interrupt Register EXIR (**figure 42**).

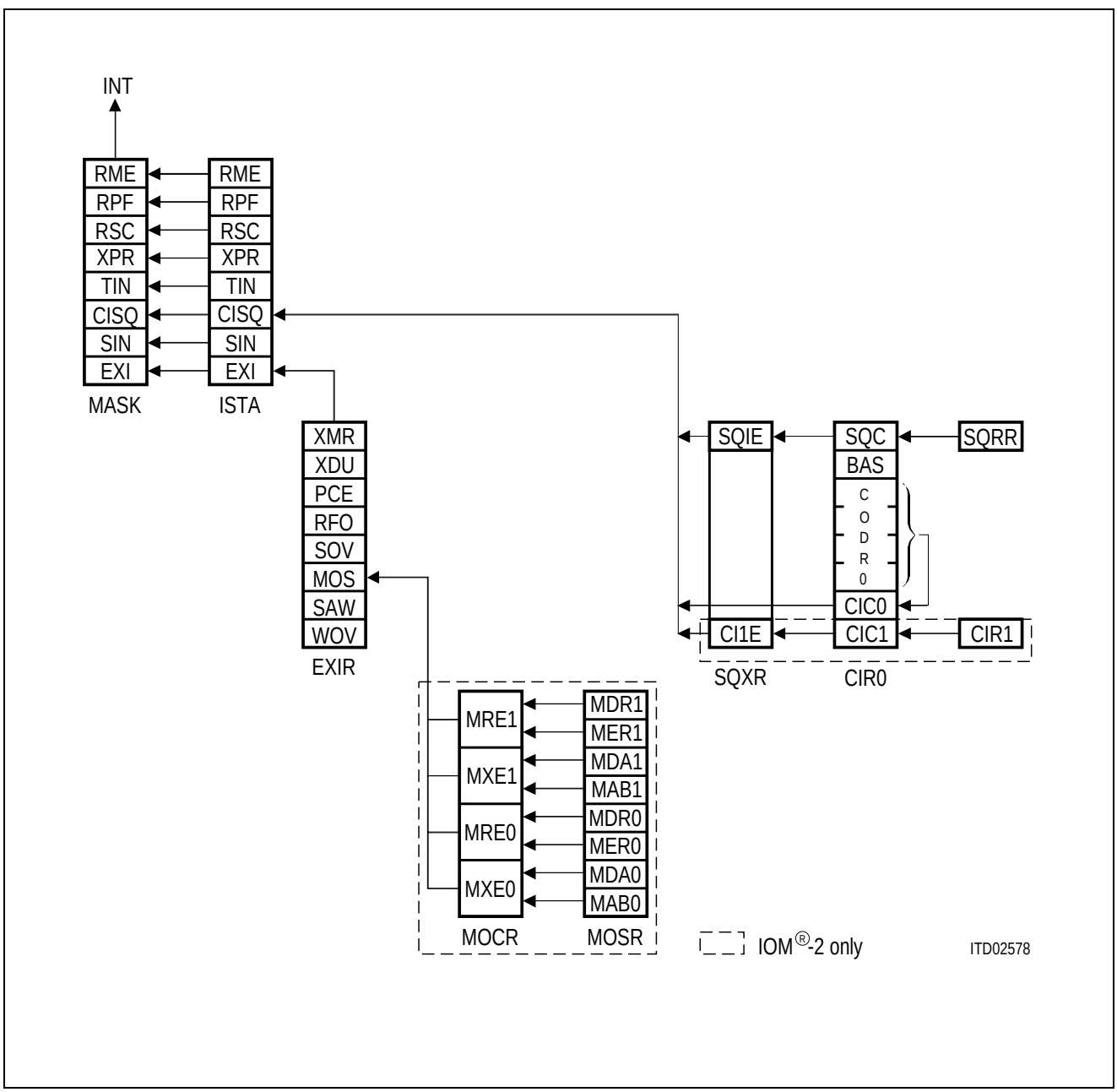


Figure 42
ISAC®-S TE Interrupt Structure

A read of the ISTA register clears all bits except EXI and CISQ. CISQ is cleared by reading CIR0. A read of EXIR clears the EXI bit in ISTA as well as the EXIR register.

When all bits in ISTA are cleared, the interrupt line ($\overline{INT}$) is deactivated.

Each interrupt source in ISTA register can be selectively masked by setting to "1" the corresponding bit in MASK. Masked interrupt status bits are not indicated when ISTA is read. Instead, they remain internally stored and pending, until the mask bit is reset to zero. Reading the ISTA while a mask bit is active has no effect on the pending interrupt.

In the event of an extended interrupt and of a C/I- or S/Q channel change, EXI and CISQ are set even when the corresponding mask bits in MASK are active, but no interrupt ($\overline{INT}$) is generated.

Except for CISQ and MOS all interrupt sources are directly determined by a read of ISTA and (possibly) EXIR.

CISQ-Interrupt Logic

– A CISQ interrupt may originate

– from a change in the received S/Q code (SQC)

– from a change in the received C/I channel 0 code (CIC0)

or (in the case of IOM-2 terminal mode only)

– from a change in the received C/I channel 1 code (CIC1).

The three corresponding status bits SQC, CIC0 and CIC1 are read in the CIR0 register. SQC and CIC1 can be individually disabled by clearing the enable bit SQIE (SQXR register) or, respectively, CI1E (SQXR register). In this case the occurrence of a code change in SQRR/ CIR1 will not be displayed by SQC/CIC1 until the corresponding enable bit has been set to one.

Bits SQC, CIC0 and CIC1 are cleared by a read of CIR0.

An interrupt status is indicated every time a valid new code is loaded in SQRR, CIR0 or CIR1. But in case of a code change, the new code is not loaded until the previous contents have been read. When this is done and a second code change has already occurred, a new interrupt is immediately generated and the new code replaces the previous one in the register. The code registers are buffered with a FIFO size of two. Thus, if several consecutive codes are detected, only the first and the last code is obtained at the first and second register read, respectively.

MOS-Interrupt Logic

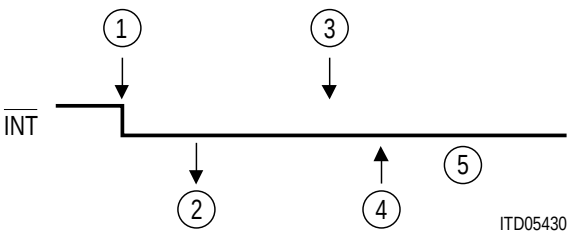
The MONITOR Data Receive (MDR) and the MONITOR End of Reception (MER) interrupt status bits have two enable bits, MONITOR Receive interrupt Enable (MRE) and MR bit Control (MRC). The MONITOR channel Data Acknowledged (MDA) and MONITOR channel Data Abort (MAB) interrupt status bits have a common enable bit MONITOR Interrupt Enable (MXE).

MRE prevents the occurrence of the MDR status, including when the first byte of a packet is received. When MRE is active (1) but MRC is inactive, the MDR-interrupt status is generated only for the first byte of a receive packet. When both MRE and MRC are active, MDR is generated and all received monitor bytes – marked by a 1-to-0 transition in MX bit – are stored. (Additionally, an active MRC enables the control of the MR handshake bit according to the MONITOR channel protocol.)

Control of Edge-Triggered Interrupt Controllers

The $\overline{\text{INT}}$ output is level active. It stays active until all interrupt sources have been serviced. If a new status bit is set while an interrupt is serviced, the $\overline{\text{INT}}$ line stays active. This may cause problems if the ISAC-S TE is connected to edge-triggered interrupt controllers (**figure 43**).

To avoid these problems, it is recommended to mask all interrupts at the end of the interrupt service program and to enable the interrupts again. This is done by writing FF_H to the MASK register and to write back the old value of the MASK register (**figure 44**).



① A status bit is set. This causes an interrupt.

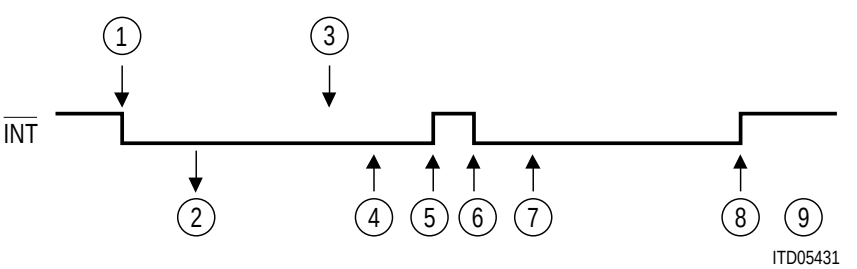
② The microprocessor starts its service routine and reads the status registers.

③ A new status bit is set before the first status bit has been read.

④ The first status bit is read.

⑤ The $\overline{INT}$ output stays active but the interrupt controller will not serve the interrupt (edge triggered).

Figure 43
INT Handling



① to ④ see above

⑤ 'FF' is written to the MASK register. This masks all interrupts and returns the $\overline{INT}$ output to its inactive state.

⑥ The old value is written to the MASK register. This will activate the $\overline{INT}$ output if an interrupt source is still active.

⑦ The microprocessor starts a new interrupt service program.

⑧ The last status bit is read.

⑨ The $\overline{INT}$ output is inactive.

Figure 44
Service Program for Edge-Triggered Interrupt Controllers

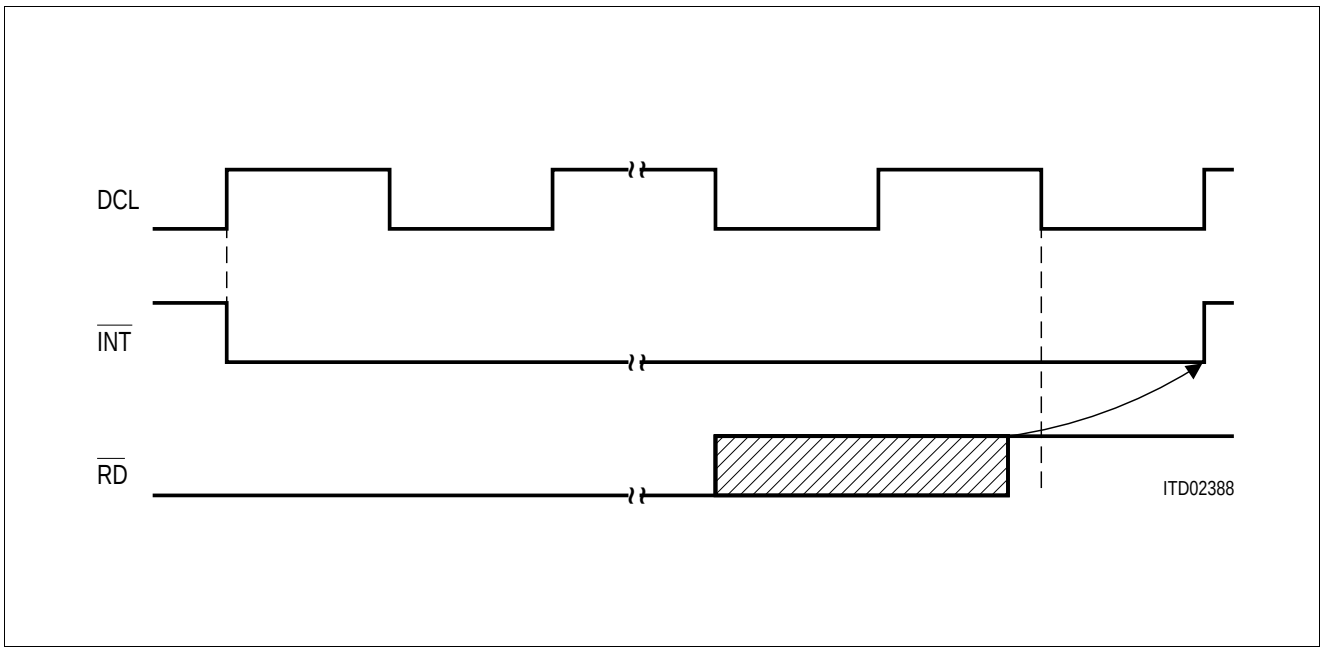


Figure 45
Timing of $\overline{\text{INT}}$ Pin

The $\overline{\text{INT}}$ line is switched with the rising edge of DCL. If no pending interrupts are internally stored, a reading of ISTA respectively EXIR or CIR0 switches the $\overline{\text{INT}}$ line to high as indicated in **figure 45**.

3.3 Control of Layer 1

3.3.1 Activation/Deactivation of IOM® Interface

The IOM interface can be switched off in the inactive state, reducing power consumption to a minimum. In this deactivated state the clock line is low and the data lines are high.

The IOM interface can be kept active while the S interface is deactivated by setting the CFS bit to "0" (SQXR register). This is the case after a hardware reset. If the IOM interface should be switched off while the S interface is deactivated, the CFS bit should be set to "1". In this case the internal oscillator is disabled when no signal (info 0) is present on the S bus. If the TE wants to activate the line, it has first to activate the IOM interface either by using the "Software Power-Up" function (SPCR:SPU bit) or by setting the CFS bit to "0" again.

For the TE case the deactivation procedure is shown in **figure 46**. After detecting the code DIU (Deactivate Indication Upstream, i.e. from TE to NT/LT-S) the layer 1 of the ISAC-S TE responds by transmitting DID (Deactivate Indication Downstream) during subsequent frames and stops the timing signals synchronously with the end of the last C/I (C/I0) channel bit of the fourth frame.

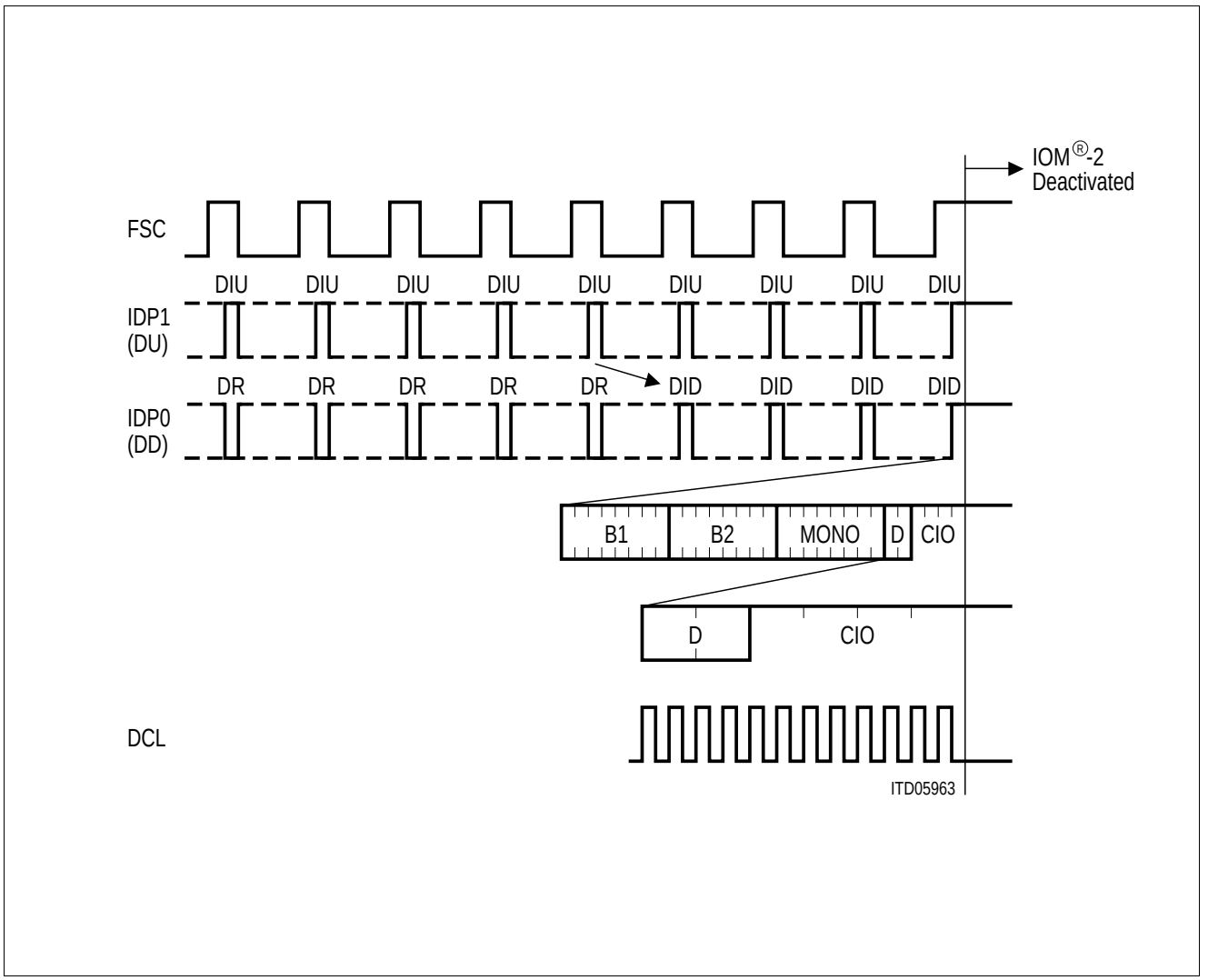
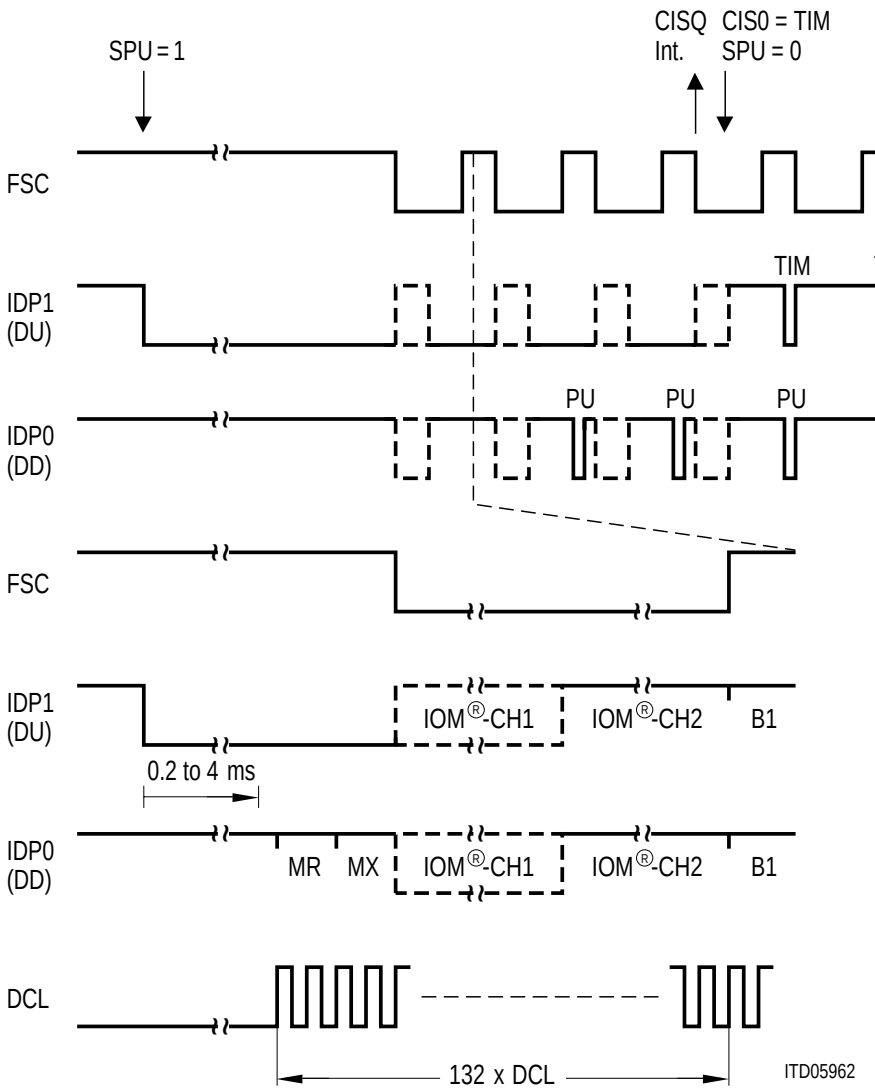


Figure 46
Deactivation of the IOM® Interface

The clock pulses will be enabled again when the IDP1 line is pulled low (bit SPU, SPCR register) i.e. the C/I command TIM = "0000" is received by layer 1, or when a non-zero level on the S-line interface is detected. The clocks are turned on after approximately 0.2 to 4 ms depending on the capacitances on XTAL 1/2.

DCL is activated such that its first rising edge occurs with the beginning of the bit following the C/I (C/I0) channel.

After the clocks have been enabled this is indicated by the PU code in the C/I channel and, consequently, by a CISQ interrupt. The IDP1 line may be released by resetting the Software Power-Up bit SPCR:SPU=0, and the C/I code written in CIX0 is output on IDP1.



Note: IDP0 is input and IDP1 is low during IOM®-CH1 if SQXR:IDC = 1
IDP0 is low and IDP1 is input during IOM®-CH1 if SQXR:IDC = 0

Figure 47
Activation of the IOM® Interface (CFS=1, register SQXR)

The ISAC-S TE supplies IOM timing signals as long as there is no DIU command in the C/I (C/I0) channel. If timing signals are no longer required and activation is not yet requested, this is indicated by programming DIU in the CIX0 register.

As an alternative to activation via IDP1 (DU), the IOM interface can be activated by setting the CFS bit to "0". The activation of FSC1 and DCL in this case is similar to **figure 47**. Note that the IOM interface can be deactivated through DIU (power-down state, **figure 46**) only if CFS is set to logical "1".

3.3.2 Activation/Deactivation of S/T Interface

Assuming the ISAC-S TE has been initialized with the required features of the application, it is now ready to transmit and receive messages in the D channel (LAPD support).

But as a prerequisite, the layer 1 has to be activated.

The layer-1 functions are controlled by commands issued via the CIXR/CIX0 register. These commands, sent over the IOM C/I channel 0 to layer 1, trigger certain procedures, such as activation/deactivation, switching of test loops and transmission of special pulse patterns. These are governed by layer-1 state diagrams in accordance with CCITT I.430. Responses from layer 1 are obtained by reading the CIRR/CIR0 register after a CISQ interrupt (ISTA).

The state diagrams are shown in **figure 49**. The activation/deactivation implemented by the ISAC-S TE agrees with the requirements set forth in CCITT recommendations. State identifiers F1-F8 (TE) are in accordance with CCITT I.430.

In the state diagrams a notation is employed which explicitly specifies the inputs and outputs on the S interface and in the C/I channel: **see figure 48**.

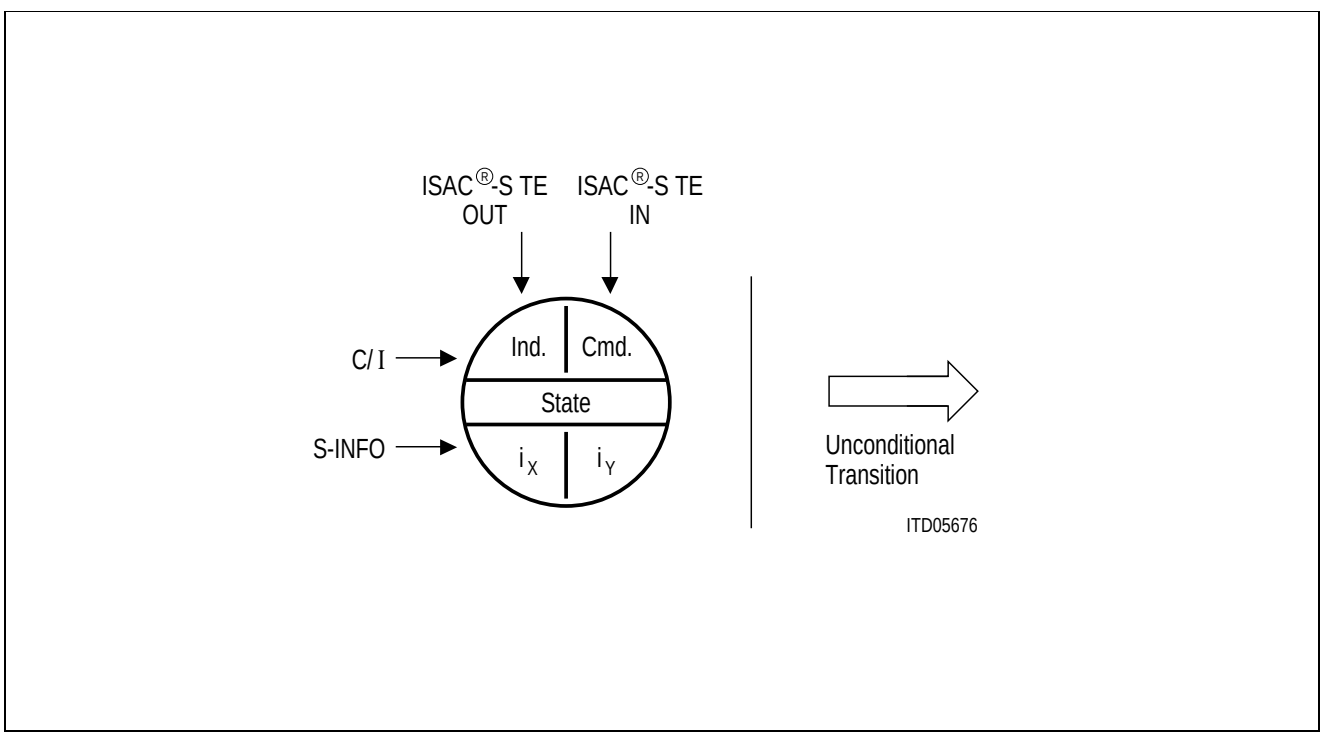


Figure 48

3.3.2.1 Layer-1 Command/Indication Codes and State Diagrams

Table 10
Commands

Command (upstream)	Abbr.	Code	Remarks
Timing	TIM	0000	Activation of all output clocks is requested
Reset	RS	0001	(x)
Send continuous zeros	SCZ	0100	Transmission of pseudo-ternary pulses at 96-kHz frequency (x)
Send single zeros	SSZ	0010	Transmission of pseudo-ternary pulses at 2-kHz frequency (x)
Activate request, set priority 8	AR8	1000	Activation command, set D-channel priority to 8 (**see note**)
Activate request, set priority 10	AR10	1001	Activation command, set D-channel priority to 10 (**see note**)
Activate request loop	ARL	1010	Activation of test loop 3 (x)
Deactivate indication upstream	DIU	1111	IOM-interface clocks can be disabled

(x) unconditional commands

Important Note: When in the activated state (AI8/AI10 indication) the 2B+D channels are only transferred from the IOM-2 to the S/T interface if an "Activate Request" command is written to the CIX0 register.

Table 11
Indications

Indication (downstream)	Abbr.	Code	Remarks
Power-up	PU	0111	IOM clocking is provided
Deactivate request	DR	0000	Deactivation request by S interface
Error indication	EI	0110	Either: (pin RST = 1 and bit CFS = 0) or RS
Level detected	RSY	0100	Signal received, receiver not synchronous
Activate request downstream	ARD	1000	Info 2 received
Test indication	TI	1010	Test loop 3 activated or continuous zeros transmitted
Awake test indication	ATI	1011	Level detected during test loop
Activate indication with priority class 8	AI8	1100	Info 4 received, D-channel priority is 8 or 9
Activate indication with priority class 10	AI10	1101	Info 4 received, D-channel priority is 10 or 11
Deactivate indication downstream	DID	1111	Clocks will be disabled in TE, quiescent state

F3 Power-Down

This is the deactivated state of the physical protocol. The receive line awake unit is active except during an $\overline{RST}$ pulse. Clocks are disabled if SQXR:CFS=1. The power consumption in this state is approximately 80 mW when the clock is running, and 8 mW otherwise.

F3 Power-Up

This state is identical to "F3 power-down", except for the C/I-output message. The state is invoked by a C/I command TIM = "0000" (or IDP1 static low). After the subsequent activation of the clocks the PU message is outputted. This occurs 0.5 ms to 4 ms after application of TIM, depending on crystal capacitances.

F3 Pending Deactivation

The ISAC-S TE reaches this state after receiving INFO0 (from states F5 to F8) for 16 ms (64 frames). This time constant is a "flywheel" to prevent accidental deactivation. From this state an activation is only possible from the line (transition "F3 pend. deact." to "F5 unsynchronized"). A power-down state may be reached only after receiving DIU.

F4 Pending Activation

Activation has been requested from the terminal, INFO1 is transmitted, INFO0 is still received, "Power-Up" is transmitted in the C/I channel. This state is stable: timer T3 (I.430) is to be implemented in software.

F5 Unsynchronized

At the reception of any signal from the NT, the ISAC-S TE ceases to transmit INFO1 and awaits identification of INFO2 or INFO4. This state is reached at most 50 µs after a signal different from INFO0 is present at the receiver of the ISAC-S TE.

F6 Synchronized

When the ISAC-S TE receives an activation signal (INFO2), it responds with INFO3 and waits for normal frames (INFO4). This state is reached at most 6 ms after an INFO2 arrives at the ISAC-S TE (when the oscillator was disabled in "F3 power-down").

F7 Activated

This is the normal active state with the layer-1 protocol activated in both directions. Note that in IOM-2 mode the 2B+D channels can only be transmitted to the S/T interface if an "Activation Request" command is written to the CIX0 register. From state "F6 synchronized", state F7 is reached at most 0.5 ms after reception of INFO4. From state "F3 power-down" with the oscillator disabled, state F7 is reached at most 6 ms after the ISAC-S TE is directly activated by INFO4.

F8 Lost Framing

This is the condition where the ISAC-S TE has lost frame synchronization and is awaiting re-synchronization by INFO2 or INFO4 or deactivation by INFO0.

Unconditional States

Loop 3 Closed

On Activate Request Loop command, INFO3 is sent by the line transmitter internally to the line receiver (INFO0 is transmitted to the line). The receiver is not yet synchronized.

Loop 3 Activated

The receiver is synchronized on INFO3 which is looped back internally from the transmitter. Data may be sent. The indication "TI" or "ATI" is output depending on whether or not a signal different from INFO0 is detected on the S interface.

Test Mode Continuous Pulses

Continuous alternating pulses are sent.

Test Mode Single Pulses

Single alternating pulses are sent (2-kHz repetition rate).

Reset State

A software reset (RS) forces the ISAC-S TE to an idle state where the analog components are disabled (transmission of INFO0) and the S line awake detector is inactive. Thus activation from the NT is not possible. Clocks are still supplied (TE mode) and the outputs are in a low impedance state.

The reset state should be left only with a "Deactivation Indication Upstream" (DIU) command before any other command is given.

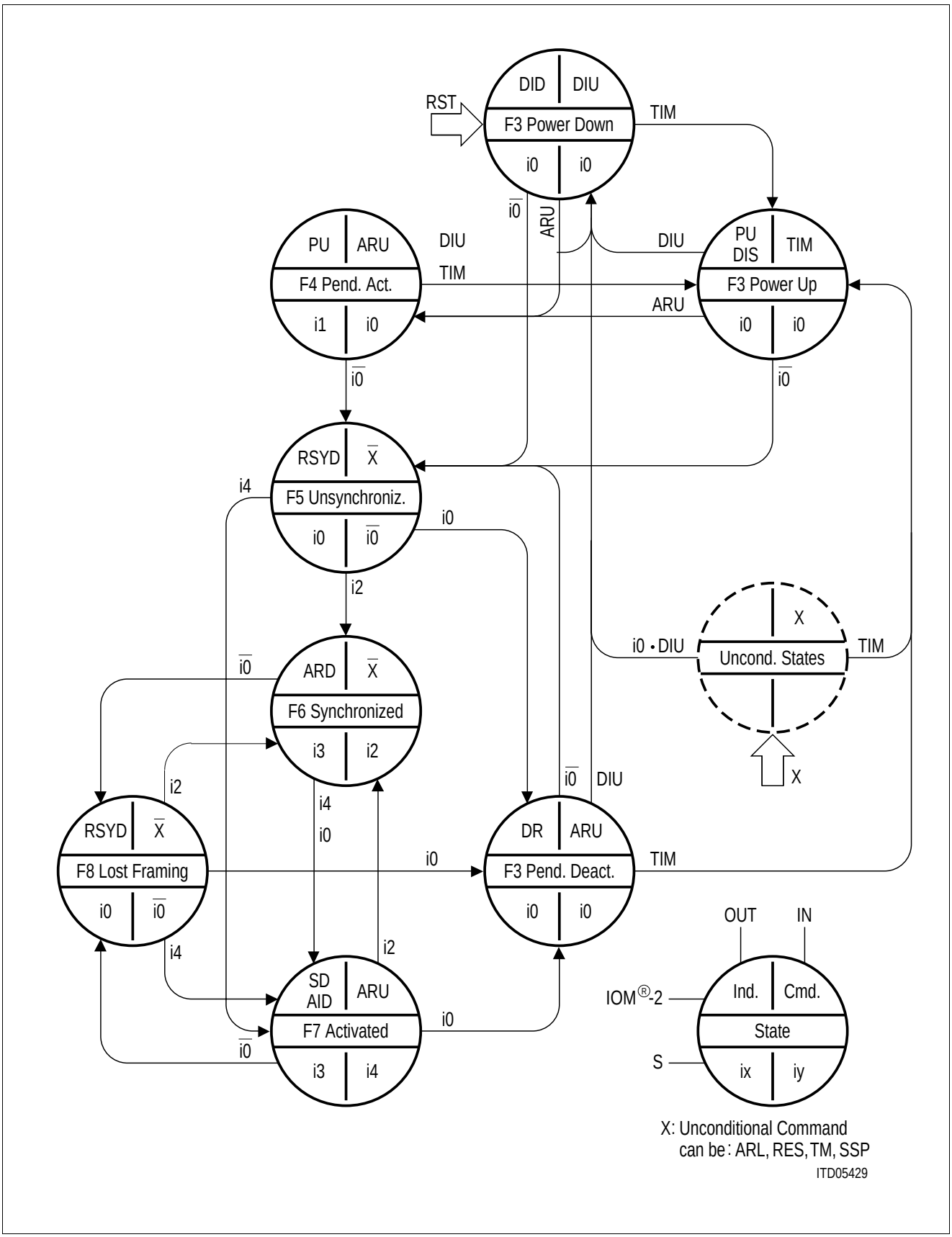


Figure 49a
State Diagram

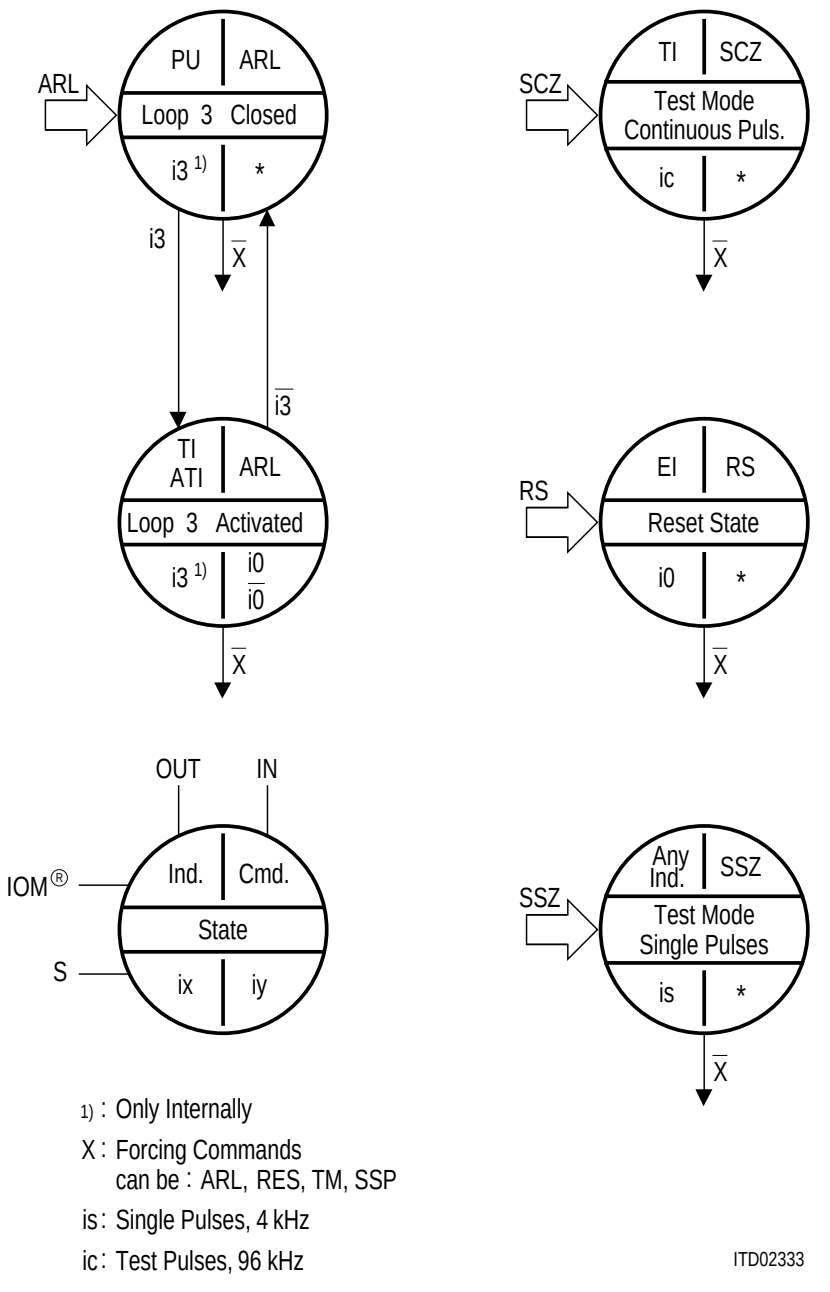


Figure 49b
State Diagram: Unconditional Transitions

3.3.3 Example of Activation/Deactivation

An example of an activation/deactivation of the S interface, with the time relationships mentioned in the previous chapters, is shown in **figure 50**, in the case of an ISAC®-S TE in TE and an ISAC-S in LT-S Mode.

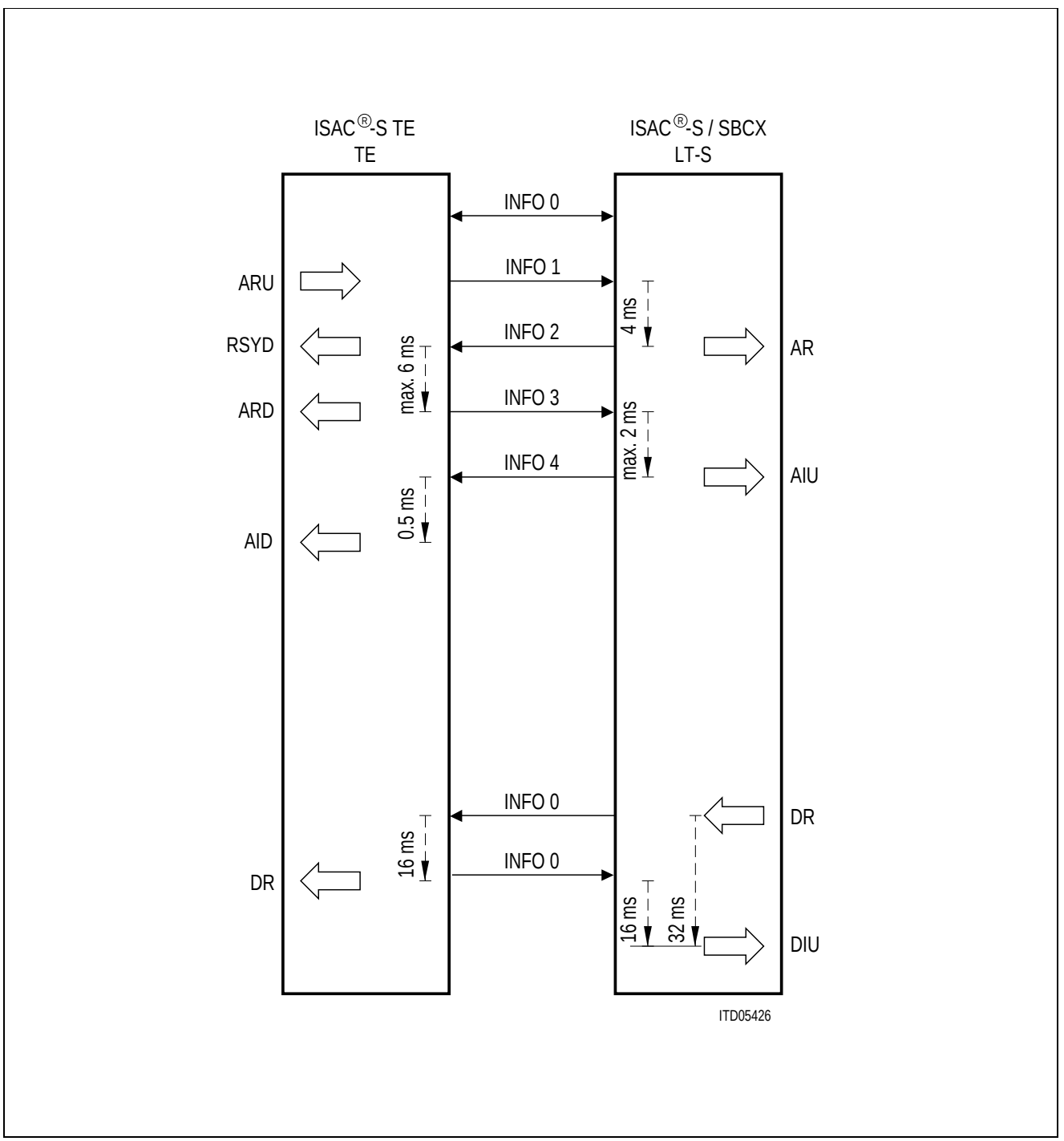


Figure 50
Example of Activation/Deactivation

3.4 Control of Layer-2 Data Transfer

The control of the data transfer phase is mainly done by commands from the µP to ISAC-S TE via the Command Register (CMDR).

Table 12 gives a summary of possible interrupts from the HDLC controller and the appropriate reaction to these interrupts.

Table 13 lists the most important commands which are issued by a microprocessor by setting one or several bits in CMDR.

The powerful FIFO logic, which consists of a 2×32 byte receive and 2×32 byte transmit FIFO, as well as an intelligent FIFO controller, builds a flexible interface to the upper protocol layers implemented in the microcontroller.

The extent of LAPD protocol support is dependent on the selected message transfer mode, **see section 2.3.2**.

Table 12
Interrupts from ISAC®-S TE HDLC Controller

Mnemonic	Register (addr. hex)	Meaning	Reaction
		Layer-2 Receive	
RPF	ISTA (20)	Receive Pool Full. Request to read received bytes of an uncompleted HDLC frame from RFIFO	Read 32 bytes from RFIFO and acknowledge with RMC.
RME	ISTA (20)	Receive Message End. Request to read received bytes of a complete HDLC frame (or the last part of a frame) from RFIFO.	Read RFIFO (number of bytes given by RBCL4-0) and status information and acknowledge with RMC.
RFO	EXIR (24)	Receive Frame Overflow. A complete frame has been lost because storage space in RFIFO was not available.	Error report for statistical purposes. Possible cause: deficiency in software.
PCE	EXIR (24)	Protocol Error. S- or I frame with incorrect N(R) or S frame with I field received (in auto-mode only) or an I frame which is not a command or S frame with an undefined control field.	Link re-establishment. Indication to layer 3.

Table 12 (cont'd)

Mnemonic	Register (addr. hex)	Meaning	Reaction
		Layer-2 Transmit	
XPR	ISTA (20)	Transmit Pool Ready. Further octets of an HDLC frame can be written to XFIFO. If XIFC was issued (auto mode), indicates that the message was successfully acknowledged with S frame.	Write data bytes in the XFIFO if the frame currently being transmitted is not finished or a new frame is to be transmitted, and issue an XIF, XIFC, XTF or XTFC command. In auto mode applications read the information in chapter 2.4.5.2.
XMR	EXIR (24)	Transmit Message Repeat. Frame must be repeated because of a transmission error (all HDLC message transfer modes) or a received negative acknowledgement (auto mode only) from peer station.	Transmission of the frame must be repeated. No indication to layer 3.
XDU	EXIR (24)	Transmit Data Underrun. Frame has been aborted because the XFIFO holds no further data and XME (XIFC or XTFC) was not issued.	Transmission of the frame must be repeated. Possible cause: excessive software reaction times.
RSC	ISTA (20)	Receive Status Change. A status change from peer station has been received (RR or RNR frame), auto-mode only.	Stop sending new I frames.
TIN	ISTA (20)	Timer Interrupt. External timer expired or, in auto-mode, internal timer (T200) and repeat counter (N200) both expired.	Link re-establishment. Indication to layer 3. (auto-mode)

Table 13
List of Commands (CMDR (21) Register)

Command Mnemonic	HEX	Bit 7…0		Meaning
RMC	80	1000	0000	Receive Message Complete. Acknowledges a block (RPF) or a frame (RME) stored in the RFIFO.
RRES	40	0100	0000	Reset HDLC Receiver. The RFIFO is cleared. The transmit and receive counters (V(S), V(R)) are reset (auto-mode).
RNR	20	0010	0000	Receiver Not Ready (auto-mode). An I- or S frame will be acknowledged with RNR frame.
STI	10	0001	0000	Start Timer.
XTFC (XTF+XME)	0A	0000	1010	Transmit Transparent Frame and Close. Enables the "transparent" transmission of the block entered last in the XFIFO. The frame is closed with a CRC and a flag.
XIFC (XIF+XME)	06	0000	0110	Transmit I frame and Close. Enables the "auto-mode" transmission of the block entered last in the XFIFO. The frame is closed with a CRC and a flag.
XTF	08	0000	1000	Transmit Transparent Frame. Enables the "transparent" transmission of the block entered last in the XFIFO without closing the frame.
XIF	04	0000	0100	Transmit I frame. Enables the "auto-mode" transmission of the block entered last in the XFIFO without closing the frame.
XRES	01	0000	0001	Reset HDLC Transmitter. The XFIFO is cleared. A frame currently in transmission will be aborted and closed by an abort sequence (7 "1").

3.4.1 HDLC-Frame Reception

Assuming a normally running communication link (layer-1 activated, layer-2 link established, TEI assigned), **figure 51** illustrates the transfer of an I frame via the D channel. The transmitter is shown on the left and the receiver on the right, with the interaction between the microcontroller system and the ISAC-S TE in terms of interrupt and command stimuli.

When the frame (excluding the CRC field) is not longer than 32 bytes, the whole frame is transferred in one block. The reception of the frame is reported via the Receive Message End (RME) interrupt. The number of bytes stored in RFIFO can be read out from RBCL. The Receive Status Register (RSTA) includes information about the frame, such as frame aborted yes/no or CRC valid yes/no and, if complete or partial address recognition is selected, the identification of the frame address.

Depending on the HDLC-message transfer mode, the address and control field of the frame can be read from auxiliary registers (SAPR and RHCR), as shown in **figure 52**.

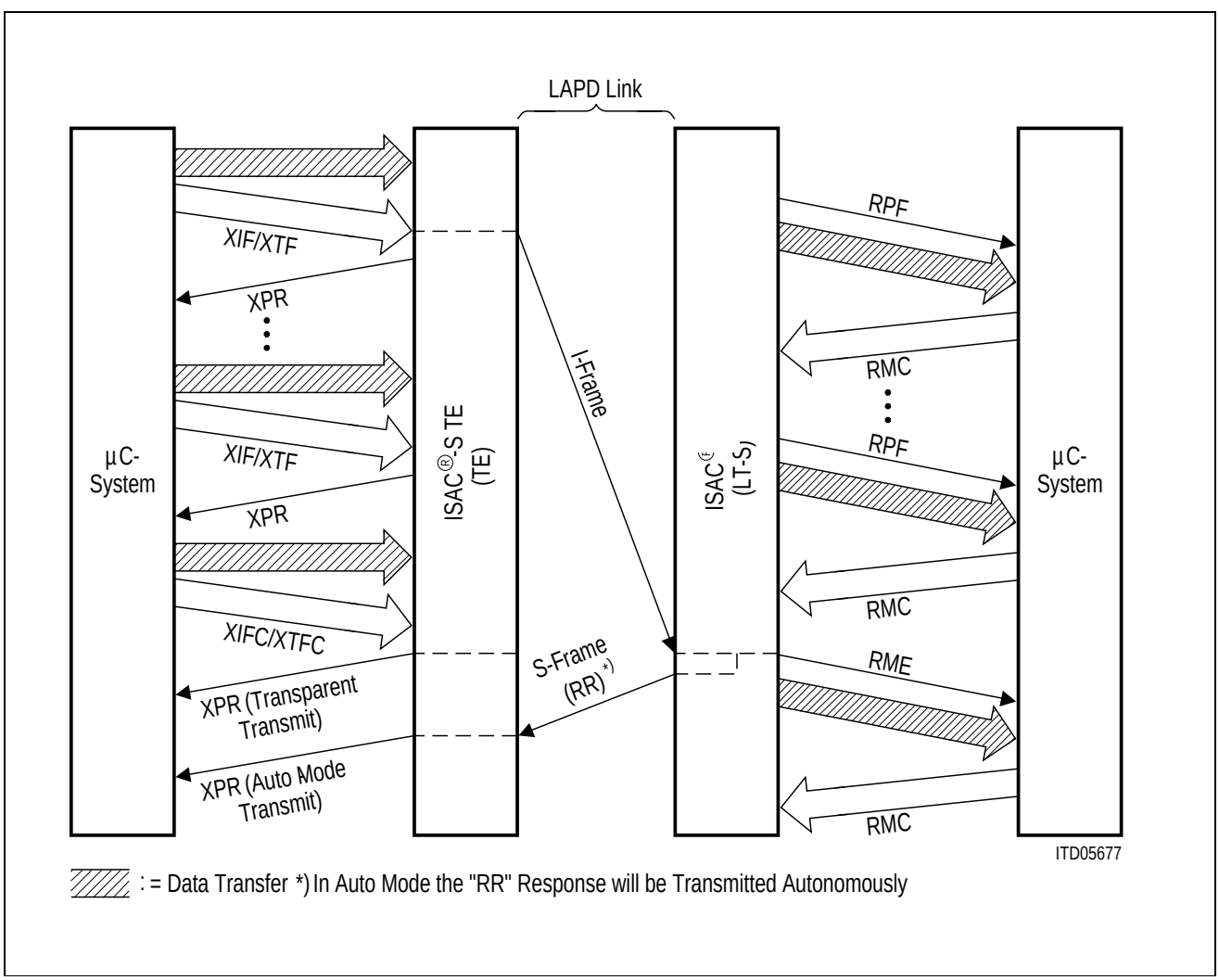


Figure 51
Transmission of an I Frame in the D Channel (Subscriber to Exchange)

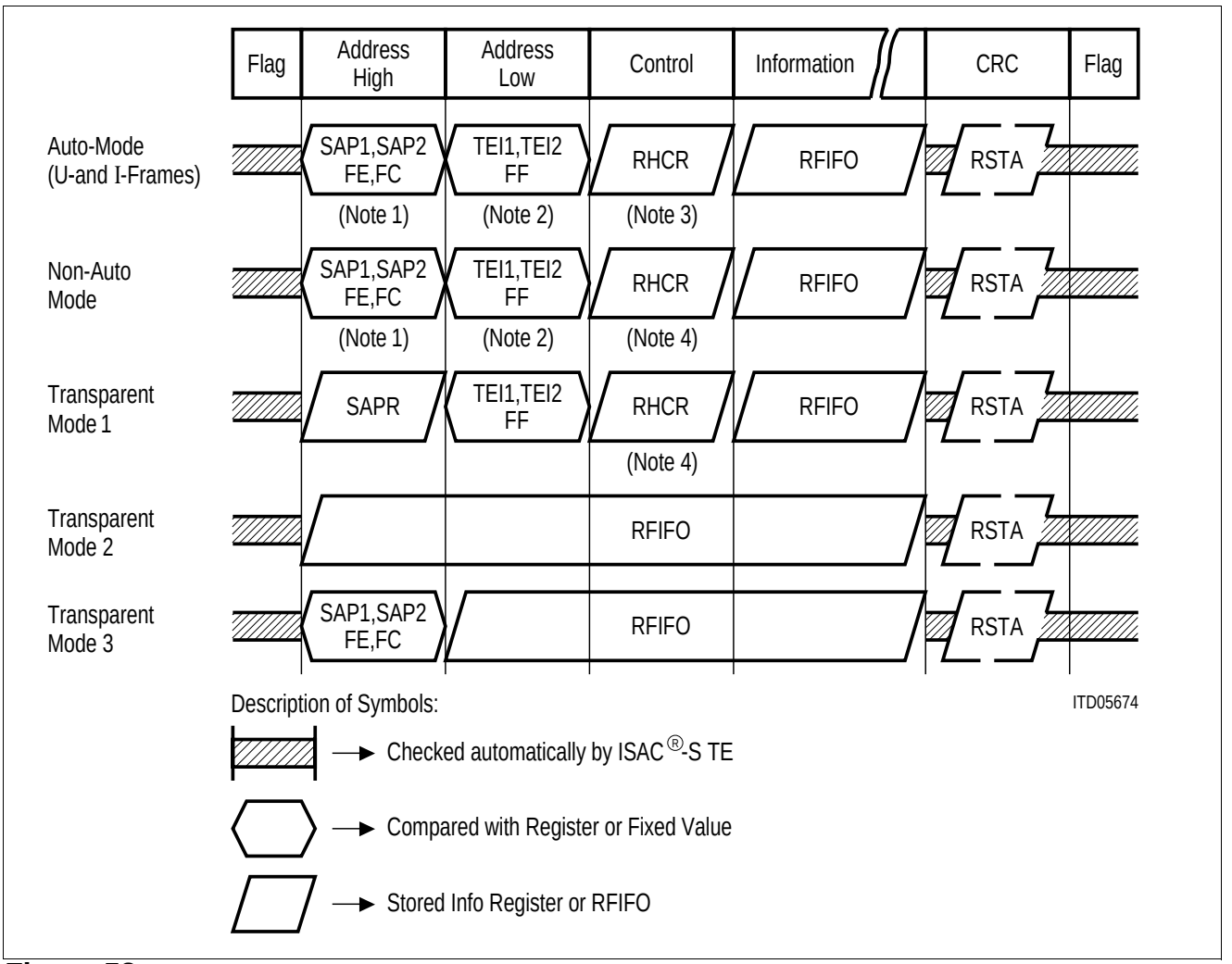


Figure 52
Receive Data Flow

Note 1 Only if a 2-byte address field is defined (MDS0 = 1 in MODE register).

Note 2 Comparison with Group TEI (FF_H) is only made if a 2-byte address field is defined (MDS0 = 1 in MODE register).

Note 3 In the case of an extended, modulo 128 control field format (MCS = 1 in SAP2 register) the control field is stored in RHCR in compressed form (I frames).

Note 4 In the case of an extended control field, only the first byte is stored in RHCR, the second in RFIFO.

A frame longer than 32 bytes is transferred to the microcontroller in blocks of 32 bytes plus one remainder block of length 1 to 32 bytes. The reception of a 32-byte block is reported by a Receive Pool Full (RPF) interrupt and the data in RFIFO remains valid until this interrupt is acknowledged (RMC). This process is repeated until the reception of the remainder block is completed, as reported by RME (**figure 51**). When the total frame length exceeds 4095 bytes, bit OV (RBCH) is set but the counter is not blocked. If the second RFIFO pool has been filled or an end-of-frame is received while a previous RPF or RME interrupt is not yet acknowledged by RMC, the corresponding interrupt will be generated only when RMC has been issued. When

RME has been indicated, bits 0-4 of the RBCL register represent the number of bytes stored in the RFIFO. Bits 7-5 of RBCL and bits 0 to 3 of RBCH indicate the total number of 32-byte blocks which where stored until the reception of the remainder block.

The contents of RBCL, RBCH and RSTA registers are valid only after the occurrence of the RME interrupt, and remain valid until the microprocessor issues an acknowledgement (RMC). The contents of RHCR and/or SAPR, also remain valid until acknowledgement.

If a frame could not be stored due to a full RFIFO, the microcontroller is informed of this via the Receive Frame Overflow interrupt (RFO).

3.4.2 HDLC-Frame Transmission

After the XFIFO status has been checked by polling the Transmit FIFO Write Enable (XFW) bit or after a Transmit Pool Ready (XPR) interrupt, up to 32 bytes may be entered in XFIFO. Transmission of an HDLC frame is started when a transmit command (**see table 13**) is issued. The opening flag is generated automatically. In the case of an auto-mode transmission (XIF or XIFC), the control field is also generated by the ISAC-S TE, and the contents of register XAD1 (and, for LAPD, XAD2) are transmitted as the address, as shown in **figure 53**.

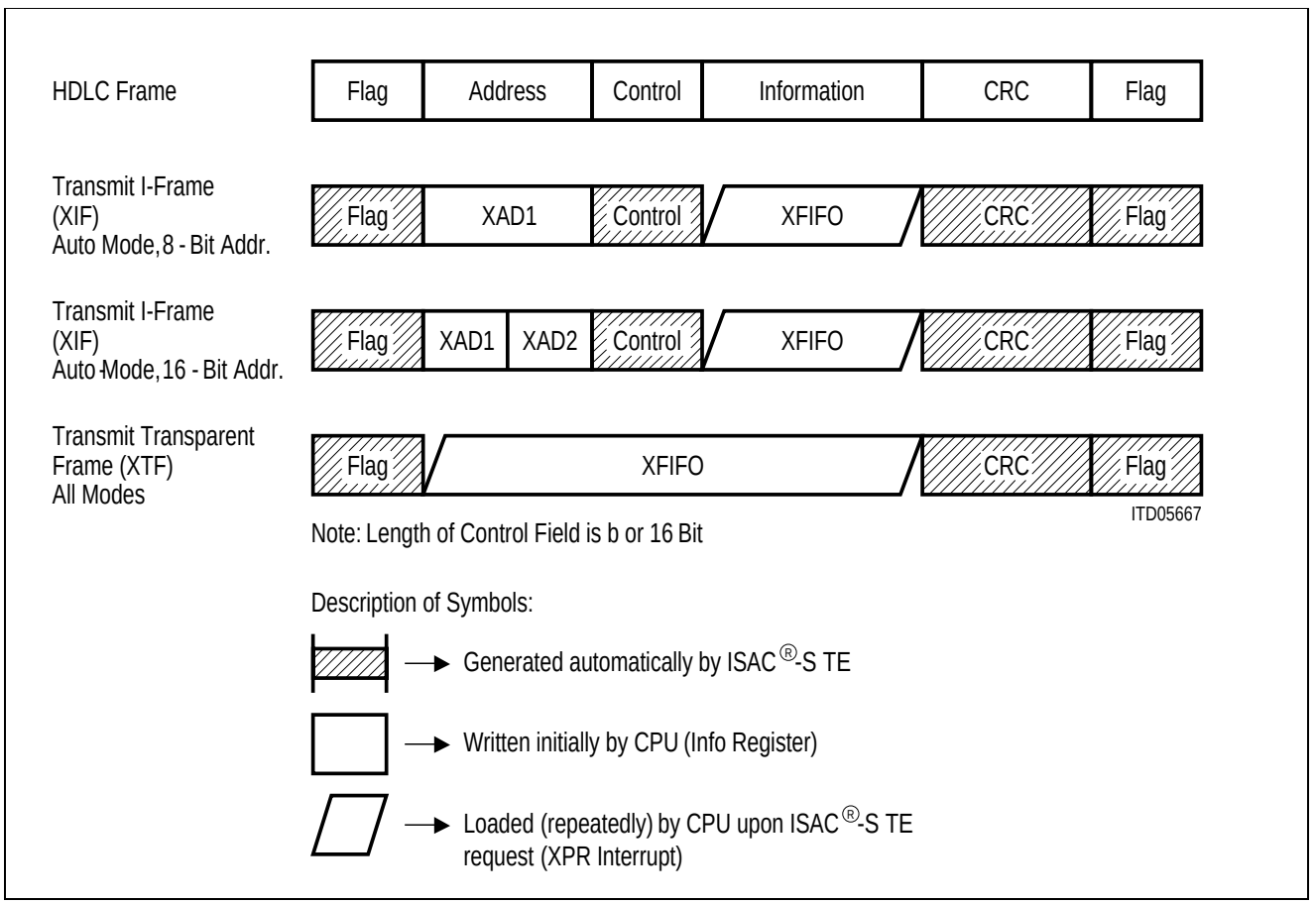


Figure 53
Transmit Data Flow

The HDLC controller will request another data block by an XPR interrupt if there are no more than 32 bytes in XFIFO and the frame close command bit (Transmit Message End XME) has not been set. To this the microcontroller responds by writing another pool of data and re-issuing a transmit command for that pool. When XME is set, all remaining bytes in XFIFO are transmitted, the CRC field and the closing flag of the HDLC frame are appended and the controller generates a new XPR interrupt.

The microcontroller does not necessarily have to transfer a frame in blocks of 32 bytes. As a matter of fact, the sub-blocks issued by the microcontroller and separated by a transmit command, can be between 0 and 32 bytes long.

If the XFIFO runs out of data and the XME command bit has not been set, the frame will be terminated with an abort sequence (seven 1's) followed by inter-frame time fill, and the microcontroller will be advised by a Transmit Data Underrun (XDU) interrupt. An HDLC frame may also be aborted by setting the Transmitter Reset (XRES) command bit.

3.5 Reset

After a hardware reset (pin RST), layer 1 will have reached the following state:

– F3 standby

according to CCITT I.430.

F3 standby state means that the internal oscillator, the DCL clock and FSC1 are active.

During the reset pulse pin SDS1 is "low", all other pins are in high impedance state. The S/T interface awake detector is active after reset. The F3 power down state, where the internal oscillator itself is disabled, can be reached by setting the CFS bit (SQXR register) to logical "1".

A subset of ISAC-S TE registers with defined reset values is listed in **table 14.**

Table 14
State of ISAC®-S TE Registers after Hardware Reset

Register (address (hex))	Value after Reset (hex)	Meaning
ISTA (20)	00	No interrupts
MASK (20)	00	All interrupts enabled
EXIR (24)	00	No interrupts
STAR (21)	48 (4A)	– XFIFO is ready to be written to – RFIFO is ready to receive at least 16 octets of a new message
CMDR (21)	00	No command

Table 14 (cont'd)

Register (address (hex))	Value after Reset (hex)	Meaning
MODE (22)	00	– auto-mode – 1-octet address field – external timer mode – receiver inactive
RBCL (25) RBCH (2A)	00 XXX00000$_2$	– no frame bytes received
SPCR (30)	00	– IDP1 pin = "High" – Timing mode 0 – IOM interface test loop deactivated – SDS1 pin = "Low"
CIR0 (31)	7C	– no change in S/Q channel – another device occupies the D and C/I channels – received C/I code = "1111" – no C/I code change
CIX0 (31)	3C	– TIC bus is not requested for transmitting a C/I code – transmitted C/I code = "1111"
STCR (37)	00	– terminal specific functions disabled – TIC-bus address = "0000" – no synchronous transfer
ADF1 (38)	00	– no test mode – active clock signals (standby) in TE mode – no prefilter – inter-frame time fill = continuous "1"
ADF2 (39)	00	– IOM-1 interface mode selected – SDS1 low
SQXR (3B)	0F/00	– S, Q interrupt not enabled

3.6 Initialization

During initialization a subset of registers have to be programmed to set the configuration parameters according to the application and desired features. They are listed in **table 15**. After reset, the ISAC-S TE is in IOM-1 mode. As a result, the fist microcontroller operation has to be an access to ADF2 to program IOM-2 interface mode.

Table 15

Register (address)	Bit	Effect	Application	Restricted to
ADF2 (39_H)	IMS	Program IOM-2 interface mode		
	D1C2-0	Polarity of SDS1		
	ODS	IOM-output driver tristate/open drain		
SPCR (30_H) (Note)	SPU	Set the ISAC-S TE in standby by requesting clocks (if CFS = 1, register SQXR)		
	TLP	IOM-interface test loop		
	C2C1-0 C1C1-0	B-channel switching or B/IC channel connect		
SQXR ($3B_H$)	IDC	IOM-Data Port IDP0,1 direction control (must be set to "0" for normal operation)		
	CFS	0 Permanent standby 1 Power-down state enabled		
ADF1 (38_H)	TEM	Test Mode	Tests with layer 1 disabled	
	PFS	Prefilter enable	TE	
	IOF	IOM OFF/ON		IOM-2
CIX0 (31_H)	RSS	Hardware reset generated by either subscriber/ exchange awake or watchdog timer	TE specific functions (TSF = 1)	

Table 15 (cont'd)

Register	(address)	Bit	Effect	Application	Restricted to
STCR	(37_H)	TSF	Terminal specific function enable		
		TBA2-0	TIC-bus address	Bus configuration for D + C/I (TIC)	
MODE	(22_H)	MDS2-0	HDLC-message transfer mode 2 bytes/1 byte address		
		TMD	Timer mode external/internal	Auto-mode only	
		DIM2-0	Point-to-point/TIC-bus configuration on IOM interface, for D + C/I channel arbitration Point-to-point/bus configuration on S/T interface, for D-channel access.		
TIMR	(23_H)	CNT VALUE	N1 and T1 in internal timer mode (TMD = 1) T2 in external timer mode		
XAD1 XAD2	(24_H) (25_H)		SAPI, TEI Transmit frame address	Auto-mode only	
SAP1/2 TEI1/2	(26_H/27_H) (28_H/29_H)		Receive SAPI, TEI address values for internal address recognition		

Note: After a hardware reset the pin SDS1 is "low", until the SPCR is written to for the first time. From that moment on, the function taken on by these pins depends on the state of the IOM Mode Select bit IMS (ADF2 register).

4 Detailed Register Description

The parameterization of the ISAC-S TE and the transfer of data and control information between the µP and ISAC-S TE is performed through two register sets.

The register set in the address range 00-2B$_H$ pertains to the HDLC transceiver and LAPD controller. It includes the two FIFOs having an identical address range from 00-1F$_H$.

The register set ranging from 30-3B$_H$ pertains to the control of layer-1 functions and of the IOM interface.

The address map and a register summary are shown in the following tables:

Table 16
ISAC®-S TE Address Map 00-2B$_H$

Address (hex)	Read		Write	
	Name	**Description**	**Name**	**Description**
00 . . 1F	RFIFO	Receive FIFO	XFIFO	Transmit FIFO
20	ISTA	Interrupt Status Register	MASK	Mask Register
21	STAR	Status Register	CMDR	Command Register
22	MODE	Mode Register		
23	TIMR	Timer Register		
24	EXIR	Extended Interrupt Register	XAD1	Transmit Address 1
25	RBCL	Receive Frame Byte Count Low	XAD2	Transmit Address 2
26	SAPR	Received SAPI	SAP1	Individual SAPI 1
27	RSTA	Receive Status Register	SAP2	Individual SAPI 2
28			TEI1	Individual TEI 1
29	RHCR	Receive HDLC Control	TEI2	Individual TEI 2
2A	RBCH	Receive Frame Byte Count High		
2B	STAR2	Status Register 2		

Table 17
ISAC®-S TE Address Map 30-3B$_H$

Address (hex)	Read		Write	
	Name	**Description**	**Name**	**Description**
30	SPCR	Serial Port Control Register		
31	CIR0	Command/Indication Receive 0	CIX0	Command/Indication Transmit 0
32	MOR0	MONITOR Receive 0	MOX0	MONITOR Transmit 0
33	CIR1	Command/Indication Receive 1	CIX1	Command/Indication Transmit 1
34	MOR1	MONITOR Receive 1	MOX1	MONITOR Transmit 1
35	C1R	Channel Register 1		
36	C2R	Channel Register 2		
37	B1CR	B1-Channel Register	STCR	Sync Transfer Control Register
38	B2CR	B2-Channel Register	ADF1	Additional Feature Register 1
39	ADF2	Additional Feature Register 2		
3A	MOSR	MONITOR Status Register	MOCR	MONITOR Control Register
3B	SQRR	S-, Q-Channel Receive Register	SQXR	S-, Q-Channel Transmit Register

Table 18
Register Summary: HDLC Operation and Status Registers

	7							0		
20_H	RME	RPF	RSC	XPR	TIN	CISQ	SIN	EXI	ISTA	R
20_H	RME	RPF	RSC	XPR	TIN	CISQ	SIN	EXI	MASK	W
21_H	XDOV	XFW	XRNR	RRNR	MBR	MAC1	X	MAC0	STAR	R
21_H	RMC	RRES	RNR	STI	XTF	XIF	XME	XRES	CMDR	W
22_H	MDS2	MDS1	MDS0	TMD	RAC	DIM2	DIM1	DIM0	MODE	R/W
23_H		CNT				VALUE			TIMR	R/W
24_H	XMR	XDU	PCE	RFO	SOV	MOS	SAW	WOV	EXIR	R
24_H									XAD1	W
25_H	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	RBCL	R
25_H									XAD2	W
26_H									SAPR	R
26_H			SAPI1				CRI	0	SAP1	W
27_H	RDA	RDO	CRC	RAB	SA1	SA0	C/R	TA	RSTA	R
27_H			SAPI2				MCS	0	SAP2	W
28_H				TEI1				EA	TEI1	W
29_H									RHCR	R
29_H				TEI2				EA	TEI2	W
$2A_H$	XAC	VN1	VN0	OV	RBC1	RBC1	RBC9	RBC8	RBCH	R
$2B_H$	0	0	0	0	WFA	MULT	TREC	SDET	STAR2	R
$2B_H$	0	0	0	0	0	MULT	0	0	STAR2	W

Table 19
Register Summary: Special Purpose Register IOM®-2 Mode

IOM®-2:

	7							0		
30_H	SPU	0	0	TLP	C1C1	C1C0	C2C1	C2C0	SPCR	R/W
31_H	SQC	BAS		CODR0			CIC0	CIC1	CIR0	R
31_H	RSS	BAC		CODX0			1	1	CIX0	W
32_H									MOR0	R
32_H									MOX0	W
33_H			CODR1				MR1	MX1	CIR1	R
33_H			CODX1				1	1	CIX1	W
34_H									MOR1	R
34_H									MOX1	W
35_H									C1R	R/W
36_H									C2R	R/W
37_H									B1CR	R
37_H	TSF	TBA2	TBA1	TBA0	ST1	ST0	SC1	SC0	STCR	W
38_H									B2CR	R
38_H	WTC1	WTC2	TEM	PFS	IOF	0	0	ITF	ADF1	W
39_H	IMS	0	0	0	ODS	D1C2	D1C1	D1C0	ADF2	R/W
$3A_H$	MDR1	MER1	MDA1	MAB1	MDR0	MER0	MDA0	MAB0	MOSR	R
$3A_H$	MRE1	MRC1	MXE1	MXC1	MRE0	MRC0	MXE0	MXC0	MOCR	W
$3B_H$	IDC	CFS	CI1E	SYN	SQR1	SQR2	SQR3	SQR4	SQRR	R
$3B_H$	IDC	CFS	CI1E	SQIE	SQX1	SQX2	SQX3	SQX4	SQXR	W

4.1 HDLC Operation and Status Registers

4.1.1 Receive FIFO RFIFO Read Address 00-1F$_H$

A read access to any address within the range 00-1F$_H$ gives access to the "current" FIFO location selected by an internal pointer which is automatically incremented after each read access. This allows for the use of efficient 'move string' type commands by the processor.

The RFIFO contains up to 32 bytes of received frame.

After an ISTA:RPF interrupt, exactly 32 bytes are available.

After an ISTA:RME interrupt, the number of bytes available can be obtained by reading the RBCL register.

4.1.2 Transmit FIFO XFIFO Write Address 00-1F$_H$

A write access to any address within the range 00-1F$_H$ gives access to the "current" FIFO location selected by an internal pointer which is automatically incremented after each write access. This allows for the use of efficient 'move string' type commands by the processor.

Up to 32 bytes of transmit data can be written into the XFIFO following an ISTA:XPR interrupt.

4.1.3 Interrupt Status Register ISTA Read Address 20$_H$

Value after reset: 00$_H$

7							0
RME	RPF	RSC	XPR	TIN	CISQ	SIN	EXI

RME Receive Message End

One complete frame of length less than or equal to 32 bytes, or the last part of a frame of length greater than 32 bytes has been received. The contents are available in the RFIFO. The message length and additional information may be obtained from RBCH + RBCL and the RSTA register.

RPF Receive Pool Full

A 32-byte block of a frame longer than 32 bytes has been received and is available in the RFIFO. The frame is not yet complete.

RSC Receive Status Change. Used in auto-mode only.

A status change in the receiver of the remote station – Receiver Ready/Receiver Not Ready – has been detected (RR or RNR S frame).

The actual status of the remote station can be read from the STAR register (RRNR bit).

XPR Transmit Pool Ready

A data block of up to 32 bytes can be written to the XFIFO.

An XPR interrupt will be generated in the following cases:

– after an XTF or XIF command, when one transmit pool is emptied and the frame is not yet complete

– after an XTF together with an XME command is issued, when the whole transparent frame has been transmitted

– after an XIF together with an XME command is issued, when the whole I frame has been transmitted and a positive acknowledgement from the remote station has been received, (auto-mode).

TIN Timer Interrupt

The internal timer and repeat counter has expired (see TIMR register).

CISQ C/I- or S/Q-Channel Change

A change in C/I channel 0, C/I channel 1 (only in IOM-2 TE mode) or S/Q channel has been recognized. The actual value can be read from CIR0, CIR1 or SQRR.

SIN Synchronous Transfer Interrupt

When programmed (STCR register), this interrupt is generated to enable the processor to lock on to the IOM timing, for synchronous transfers.

EXI Extended Interrupt

This bit indicates that one of six non-critical interrupts has been generated. The exact interrupt cause can be read from EXIR.

Note: A read of the ISTA register clears all bits except EXI and CISQ. EXI is cleared by reading the EXIR register, CISQ is cleared by reading CIRR/CIR0.

4.1.4 Mask Register MASK Write Address 20_H

Value after reset: 00_H

7							0
RME	RPF	RSC	XPR	TIN	CISQ	SIN	EXI

Each interrupt source in the ISTA register can be selectively masked by setting to "1" the corresponding bit in MASK. Masked interrupt status bits are not indicated when ISTA is read. Instead, they remain internally stored and pending, until the mask bit is reset to zero.

Note: In the event of an extended interrupt and of a C/I- or S/Q-channel change, EXI and CISQ are set in ISTA even if the corresponding mask bits in MASK are active, but no interrupt ($\overline{INT}$ pin) is generated.

4.1.5 Status Register STAR Read Address 21_H

Value after reset: 48_H or $4A_H$

7							0
XDOV	XFW	XRNR	RRNR	MBR	MAC1	X	MAC0

XDOV Transmit Data Overflow

More than 32 bytes have been written in one pool of the XFIFO, i.e. data has been overwritten.

XFW Transmit FIFO Write Enable

Data can be written in the XFIFO. This bit may be polled instead of (or in addition to) using the XPR interrupt.

XRNR Transmit RNR. Used in auto-mode only

In auto-mode, this bit indicates whether the ISAC-S TE receiver is in the "ready" (0) or "not ready" (1) state. When "not ready", the ISAC-S TE sends an RNR S frame autonomously to the remote station when an I frame or an S frame is received.

RRNR Receive RNR. Used in auto-mode only

In the auto-mode, this bit indicates whether the ISAC-S TE has received an RR or an RNR frame, this being an indication of the current state of the remote station: receiver ready (0) or receiver not ready (1).

MBR Message Buffer Ready

This bit signifies that temporary storage is available in the RFIFO to receive at least the first 16 bytes of a new message.

MAC1 MONITOR Transmit Channel 1 Active (IOM-2 terminal mode only)

Data transmission is in progress in MONITOR channel 1.

MAC0 MONITOR Transmit Channel 0 Active. Used in IOM-2 mode only.

Data transmission is in progress in MONITOR channel 0.

Note: Bit 1 may toggle dependend the time of access.

4.1.6 Command Register CMDR Write Address 21_H

Value after reset: 00_H

7							0
RMC	RRES	RNR	STI	XTF	XIF	XME	XRES

Note: The maximum time between writing to the CMDR register and the execution of the command is 2.5 DCL-clock cycles. During this time no further commands should be written to the CMDR register to avoid any loss of commands.

RMC Receive Message Complete

Reaction to RPF (Receive Pool Full) or RME (Receive Message End) interrupt. By setting this bit, the processor confirms that it has fetched the data, and indicates that the corresponding space in the RFIFO may be released.

RRES Receiver Reset

HDLC receiver is reset, the RFIFO is cleared of any data.

In addition, in auto-mode, the transmit and receive counters (V(S), V(R)) are reset

RNR Receiver Not Ready
Used in auto-mode only.

Determines the state of the ISAC-S TE HDLC receiver.

When RNR = "0", a received I or S-frame is acknowledged by an RR supervisory frame, otherwise by an RNR supervisory frame.

STI Start Timer

The ISAC-S TE hardware timer is started when STI is set to one. In the internal timer mode (TMD bit, MODE register) an S command (RR, RNR) with poll bit set is transmitted in addition. The timer may be stopped by a write of the TIMR register.

XTF Transmit Transparent Frame

After having written up to 32 bytes in the XFIFO, the processor initiates the transmission of a transparent frame by setting this bit to "1". The opening flag is automatically added to the message by the ISAC-S TE.

XIF Transmit I Frame
Used in auto-mode only

After having written up to 32 bytes in the XFIFO, the processor initiates the transmission of an I frame by setting this bit to "1". The opening flag, the address and the control field are automatically added by the ISAC-S TE.

XME Transmit Message End

By setting this bit to "1" the processor indicates that the data block written last in the XFIFO completes the corresponding frame. The ISAC-S TE terminates the transmission by appending the CRC and the closing flag sequence to the data.

XRES Transmitter Reset

HDLC transmitter is reset and the XFIFO is cleared of any data.

This command can be used by the processor to abort a frame currently in transmission.

Notes:
- After an XPR interrupt further data has to be written in the XFIFO and the appropriate Transmit Command (XTF or XIF) has to be written in the CMDR register again to continue transmission, when the current frame is not yet complete (see also XPR in ISTA).
- During frame transmission, the 0-bit insertion according to the HDLC bit-stuffing mechanism is done automatically.

4.1.7 Mode Register MODE Read/Write Address 22_H

Value after reset: 00_H

7							0
MDS2	MDS1	MDS0	TMD	RAC	DIM2	DIM1	DIM0

MDS2-0 Mode Select

Determines the message transfer mode of the HDLC controller, as follows:

MDS2 MDS1 MDS0	Mode	Number of Address Bytes	Address Comparison 1. Byte	2. Byte	Remark
0 0 0	Auto-mode	1	TEI1, TEI2	–	One-byte address compare. HDLC-protocol handling for frames with address TEI1
0 0 1	Auto-mode	2	SAP1, SAP2, SAPG	TEI1, TEI2, TEIG	Two-byte address compare. LAPD-protocol handling for frames with address SAP1 + TEI1
0 1 0	Non-auto mode	1	TEI1, TEI2	–	One-byte address compare.
0 1 1	Non-auto mode	2	SAP1, SAP2, SAPG	TEI1, TEI2, TEIG	Two-byte address compare.
1 0 0	Reserved				
1 0 1	Transparent mode 1	>1	–	TEI1, TEI2, TEIG	Low-byte address compare.
1 1 0	Transparent mode 2	–	–	–	No address compare. All frames accepted.
1 1 1	Transparent mode 3	>1	SAP1, SAP2, SAPG	–	High-byte address compare.

Note: SAP1, SAP2: two programmable address values for the first received address byte (in the case of an address field longer than 1 byte); SAPG = fixed value FC/FE_H.

TEI1, TEI2: two programmable address values for the second (or the only, in the case of a one-byte address) received address byte; TEIG = fixed value FF_H.

TMD Timer Mode

Sets the operating mode of the ISAC-S TE timer. In the external mode (0) the timer is controlled by the processor. It is started by setting the STI bit in CMDR and it is stopped by a write of the TIMR register.

In the internal mode (1) the timer is used internally by ISAC-S TE for timeout and retry conditions (handling of LAPD/HDLC protocol in auto-mode).

RAC Receiver Active

The HDLC receiver is activated when this bit is set to "1".

DIM2-0 Digital Interface Mode

These bits define the characteristics of the IOM-Data Ports (IDP0, IDP1) according to following tables:

IOM®-2 Modes (ADF2:IMS = 1)

	DIM2-0				
Characteristics	**000**	**001**	**010**	**011**	**100-111**
IOM-2 terminal mode SPCR:SPM = 0	×	×	×	×	
Last octet of IOM channel 2 used for TIC-bus access	×	×			
Stop/go bit evaluated for D-channel access handling		×		×	
Reserved					×
Applications					
TE mode		×		×	

4.1.8 Timer Register TIMR Read/Write Address 23_H

Value after reset: undefined (previous value)

7							0
	CNT				VALUE		

CNT The meaning depends on the selected timer mode (TMD bit, MODE register).

*** internal Timer Mode (TMD = 1)**

CNT indicates the maximum number of S commands "N1" which are transmitted autonomously by the ISAC-S TE after expiration of time period T1 (retry, according to HDLC).

The internal timer procedure will be **started** in auto-mode:

– after start of an I-frame transmission

or

– after an "RNR" S frame has been received.

After the last retry, a timer interrupt (TIN bit in ISTA) is generated.

The timer procedure will be **stopped** when

– a TIN interrupt is generated. The time between the start of an I-frame transmission or reception of an "RNR" S frame and the generation of a TIN interrupt is equal to: (CNT+1) × T1.

– or the TIMR is written

– or a positive or negative acknowledgement has been received.

Note: The maximum value of CNT can be 6. If CNT is set to 7, the number of retries is unlimited.

*** External Timer Mode (TMD = 0)**

CNT together with VALUE determine the time period T2 after which a TIN interrupt will be generated:

CNT × 2.048 s + T1

with T1 = (VALUE + 1) × 0.064 s,

in the normal case, and

T2 = 16348 × CNT × DCL + T1

with T1 = 512 × (VALUE + 1) × DCL

when TLP = 1 (test loop activated, SPCR register).

DCL denotes the period of the DCL clock.

The timer can be started by setting the STI bit in CMDR and will be stopped when a TIN interrupt is generated or the TIMR register is written.

Note: If CNT is set to 7, a TIN interrupt is indefinitely generated after every expiration of T1.

VALUE Determines the Time Period T1:

T1 = (VALUE + 1) × 0.064 s (SPCR:TLP = 0, normal mode)

T1 = 512 × (VALUE + 1) × DCL (SPCR:TLP = 1, test mode).

4.1.9 Extended Interrupt Register EXIR Read Address 24_H

Value after reset: 00_H

7							0
XMR	XDU	PCE	RFO	SOV	MOS	SAW	WOV

XMR Transmit Message Repeat

The transmission of the last frame has to be repeated because:

– the ISAC-S TE has received a negative acknowledgement to an I frame in auto-mode (according to HDLC/LAPD)

– or a collision on the S bus has been detected after the 32nd data byte of a transmit frame.

XDU Transmit Data Underrun

The current transmission of a frame is aborted by transmitting seven "1's" because the XFIFO holds no further data. This interrupt occurs whenever the processor has failed to respond to an XPR interrupt (ISTA register) quickly enough, after having initiated a transmission and the message to be transmitted is not yet complete.

Note: When an XMR or and XDU interrupt is generated, it is not possible to send transparent frames or I frames until the interrupt has been acknowledged by reading EXIR.

PCE Protocol Error

Used in auto-mode only.

A protocol error has been detected in auto-mode due to a received

– S- or I frame with an incorrect sequence number N(R) or

– S frame containing an I field.

– I frame which is not a command.

– S frame with an undefined control field.

RFO Receive Frame Overflow

The received data of a frame could not be stored, because the RFIFO is occupied. The whole message is lost.

This interrupt can be used for statistical purposes and indicates that the processor does not respond quickly enough to an RPF or RME interrupt (ISTA).

SOV Synchronous Transfer Overflow

The synchronous transfer programmed in STCR has not been acknowledged in time via the SC0/SC1 bit.

MOS MONITOR Status

A change in the MONITOR Status Register (MOSR) has occured.

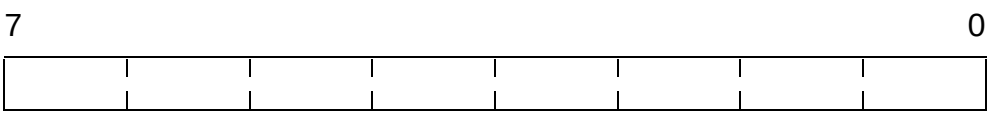
SAW Subscriber Awake
Used only if terminal specific functions are enabled (STCR:TSF = 1).

Indicates that a falling edge on the EAW line has been detected, in case the terminal specific functions are enabled (TSF bit in STCR).

WOV Watchdog Timer Overflow
Used only if terminal specific functions are enabled (STCR:TSF = 1).

Signals the expiration of the watchdog timer, which means that the processor has failed to set the watchdog timer control bits WTC1 and WTC2 (ADF1 register) in the correct manner. A reset pulse has been generated by the ISAC-S TE.

4.1.10 Transmit Address 1 XAD1 Write Address 24_H



Used in auto-mode only.

XAD1 contains a programmable address byte which is appended automatically to the frame by the ISAC-S TE in auto-mode. Depending on the selected address mode XAD1 is interpreted as follows:

*** 2-Byte Address Field**

XAD1 is the high byte (SAPI in the ISDN) of the 2-byte address field. Bit 1 is interpreted as the command/response bit "C/R". It is automatically generated by the ISAC-S TE following the rules of ISDN LAPD protocol and the CRI bit value in SAP1 register. Bit 1 has to be set to "0".

C/R Bit			
Command	**Response**	**Transmitting End**	**CRI Bit**
0	1	subscriber	0
1	0	network	1

In the ISDN LAPD the address field extension bit "EA", i.e. bit 0 of XAD1 has to be set to "0".

*** 1-Byte Address Field**

According to the X.25 LAPB protocol, XAD1 is the address of a command frame.

Note: In standard ISDN applications only 2-byte address fields are used.

4.1.11 Receive Frame Byte Count Low RBCL Read Address 25_H

Value after reset: 00_H

7							0
RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0

RBC7-0 Receive Byte Count

Eight least significant bits of the total number of bytes in a received message. Bits RBC4-0 indicate the length of the data block currently available in the RFIFO, the other bits (together with RBCH) indicate the number of whole 32-byte blocks received.

If exactly 32 bytes are received RBCL holds the value 20_H.

4.1.12 Transmit Address 2 XAD2 Write Address 25_H

7							0

Used in auto-mode only.

XAD2 contains the second programmable address byte, whose function depends on the selected address mode:

*** 2-Byte Address Field**

XAD2 is the low byte (TEI in the ISDN) of the 2-byte address field.

*** 1-Byte Address Field**

According to the X.25 LAPB protocol, XAD2 is the address of a response frame.

Note: See note to XAD1 register description.

4.1.13 Received SAPI Register SAPR Read Address 26_H

7							0

When transparent mode 1 is selected, SAPR contains the value of the first address byte of a receive frame.

4.1.14 SAPI1 Register SAP1 Write Address 26_H

7						0
		SAPI1			CRI	0

SAPI1 SAPI1 Value

Value of the first programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD protocol.

CRI Command/Response Interpretation

CRI defines the end of the ISDN user-network interface the ISAC-S TE is used on, for the correct identification of "Command" and "Response" frames. Depending on the value of CRI the C/R bit will be interpreted by the ISAC-S, when receiving frames in auto-mode, as follows:

		C/R Bit	
CRI Bit	**Receiving End**	**Command**	**Response**
0	subscriber	1	0
1	network	0	1

For transmitting frames in auto-mode, the C/R-bit manipulation will also be done automatically, depending on the value of the CRI bit (refer to XAD1-register description).

In message transfer modes with SAPI address recognition the first received address byte is compared with the programmable values in SAP1, SAP2 and the fixed group SAPI.

In 1-byte address mode, the CRI bit is to be set to "0".

4.1.15 Receive Status Register RSTA Read Address 27_H

Value after reset: undefined

7							0
RDA	RDO	CRC	RAB	SA1	SA0	C/R	TA

RDA Receive Data

A "1" indicates that data is available in the RFIFO. After an RME interrupt, a "0" in this bit means that data is available in the internal registers RHCR or SAPR only (e.g. S frame). See also RHCR-register description table.

RDO Receive Data Overflow

At least one byte of the frame has been lost, because it could not be stored in RFIFO (1).

CRC CRC Check

The CRC is correct (1) or incorrect (0).

RAB Receive Message Aborted

The receive message was aborted by the remote station (1), i.e. a sequence of 7 1's was detected.

SA1-0 SAPI Address Identification

TA TEI Address Identification

SA1-0 are significant in auto-mode and non-auto mode with a two-byte address field, as well as in transparent mode 3. TA is significant in all modes except in transparent modes 2 and 3.

Two programmable SAPI values (SAP1, SAP2) plus a fixed group SAPI (SAPG of value FC/FE_H), and two programmable TEI values (TEI1, TEI2) plus a fixed group TEI (TEIG of value FF_H), are available for address comparison.

The result of the address comparison is given by SA1-0 and TA, as follows

				Address Match with	
	SA1	**SA0**	**TA**	**1st Byte**	**2nd Byte**
Number of address bytes = 1	x x	x x	0 1	TEI2 TEI1	– –
Number of address bytes = 2	0 0 0 0 1 1 1	0 0 1 1 0 0 1	0 1 0 1 0 1 x	SAP2 SAP2 SAPG SAPG SAP1 SAP1 reserved	TEIG TEI2 TEIG TEI1 or TEI2 TEIG TEI1

Notes:
- If the SAPI values programmed to SAP1 and SAP2 are identical the reception of a frame with SAP2/TEI2 results in the indication SA1 = 1, SA0 = 0, TA = 1.
- Normally RSTA should be read by the processor after an RME interrupt in order to determine the status of the received frame. The contents of RSTA are valid only after an RME interrupt, and remain so until the frame is acknowledged via the RMC bit.

C/R Command/Response

The C/R bit identifies a receive frame as either a command or a response, according to the LAPD rules:

Command	Response	Direction
0	1	Subscriber to network
1	0	Network to subscriber

4.1.16 SAPI2 Register SAP2 Write Address 27_H

7						0
		SAPI2			MCS	0

SAPI2 SAPI2 Value

Value of the second programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD protocol.

MCS Modulo Count Select

Used in auto-mode only.

This bit determines the HDLC-control field format as follows:

0: One-byte control field (modulo 8)

1: Two-byte control field (modulo 128)

4.1.17 TEI1 Register 1 TEI1 Write Address 28_H

7							0
			TEI1				EA

EA Address Field Extension Bit

This bit has to be set "1" according to HDLC/LAPD.

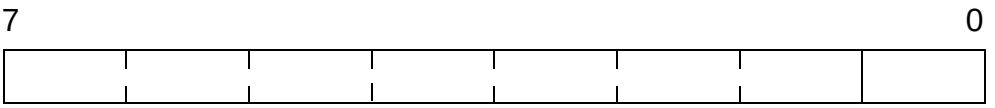
In all message transfer modes except in transparent modes 2 and 3, TEI1 is used by the ISAC-S for address recognition. In the case of a two-byte address field, it contains the value of the first programmable Terminal Endpoint Identifier according to the ISDN LAPD protocol.

In the auto-mode with a two-byte address field, numbered frames with the address SAPI1-TEI1 are handled autonomously by the ISAC-S TE according to the LAPD protocol.

Note: If the value FF_H is programmed in TEI1, received numbered frames with address SAPI1-TEI1 (SAPI1-TEIG) are not handled autonomously by the ISAC-S TE.

In auto and non-auto-modes with one-byte address field, TEI1 is a command address, according to X.25 LAPB.

4.1.18 Receive HDLC Control Register RHCR Read Address 29_H



In all modes except transparent modes 2 and 3, this register contains the control field of a received HDLC frame. In transparent modes 2 and 3, the register is not used.

	Contents of RHCR		
Mode	**Modulo 8 (MCS = 0)**	**Modulo 128 (MCS = 1)**	**Contents of RFIFO**
Auto-mode, 1-byte address (U/I frames) (Note 1)	Control field	U-frames only: Control field (Note 2)	From 3rd byte after flag (Note 4)
Auto-mode, 2-byte address (U/I frames) (Note 1)	Control field	U-frames only: Control field (Note 2)	From 4th byte after flag (Note 4)
Auto-mode, 1-byte address (I frames)		Control field in compressed form (Note 3)	From 4th byte after flag (Note 4)
Auto-mode, 2-byte address (I frames)		Control field in compressed form (Note 3)	From 5th byte after flag (Note 4)
Non-auto mode, 1-byte address	2nd byte after flag		From 3rd byte after flag
Non-auto mode, 2-byte address	3rd byte after flag		From 4th byte after flag
Transparent mode 1	3rd byte after flag		From 4th byte after flag
Transparent mode 2	–		From 1st byte after flag
Transparent mode 3	–		From 2nd byte after flag

Note 1: S frames are handled automatically and are not transferred to the microprocessor.

Note 2: For U frames (bit 0 of RHCR = 1) the control field is as in the modulo 8 case.

Note 3: For I frames (bit 0 of RHCR = 0) the compressed control field has the same format as in the modulo 8 case, but only the three LSB's of the receive and transmit counters are visible:

7			0
N(R)2-0	P	N(S)2-0	0

Note 4: I field.

4.1.19 TEI2 Register TEI2 Write Address 29_H

7							0
			TEI2				EA

EA Address Field Extension Bit

This bit is to be set to "1" according to HDLC/LAPD.

In all message transfer modes except in transparent modes 2 and 3, TEI2 is used by the ISAC-S TE for address recognition. In the case of a two-byte address field, it contains the value of the second programmable Terminal Endpoint Identifier according of the ISDN LAPD protocol.

In auto and non-auto modes with one-byte address field, TEI2 is a response address, according to X.25 LAPB.

4.1.20 Receive Frame Byte Count High RBCH Read Address $2A_H$

Value after reset: $0XX00000_2$.

7							0
XAC	VN1	VN0	OV	RBC11	RBC10	RBC9	RBC8

XAC Transmitter Active

The HDLC transmitter is active when XAC = 1. This bit may be polled. The XAC bit is active when

– either an XTF/XIF command is issued and the frame has not been completely transmitted

– or the transmission of an S frame is internally initiated and not yet completed.

VN1-0 Version Number of Chip

00 ... V1.1 version

OV Overflow

A "1" in this bit position indicates a message longer than 4095 bytes.

RBC8-11 Receive Byte Count

Four most significant bits of the total number of bytes in a received message.

Note: Normally RBCH and RBCL should be read by the processor after an RME interrupt in order to determine the number of bytes to be read from the RFIFO, and the total message length. The contents of the registers are valid only after an RME interrupt, and remain so until the frame is acknowledged via the RMC bit.

4.1.21 Status Register 2 STAR2 Read/Write Address $2B_H$

Value after reset: 00_H

a) WRITE

7							0
0	0	0	0	0	MULT	0	0

MULT Used to enable or disable the multiframe structure (see chapter 2.4.9)

1: S/T multiframe disabled
0: S/T multiframe enabled

b) READ

7							0
0	0	0	0	WFA	MULT	TREC	SDET

WFA Waiting for Acknowledge
This bit shows, if the last transmitted I frame was acknowledged, i.e. V(A) = V(S) (⇒ WFA = 0) or was not yet acknowledged, i.e. V(A) < V(S) (⇒ WFA = 1).

MULT The value written into the register bit is read.

TREC Timer Recovery Status:

0: The device is not in the timer recovery state.
1: The device is in the timer recovery state.

SDET S Frame Detected:
This bit is set to "1" by the first received correct I frame or S command with p = 1.

It is reset by reading STAR2.

4.2 Special Purpose Registers: IOM®-2 Mode

The following register description is only valid if IOM-2 is selected (ADF2:IMS-1).

4.2.1 Serial Port Control Register SPCR Read/Write Address 30_H

Value after reset: 00_H

7							0
SPU	0	0	TLP	C1C1	C1C0	C2C1	C2C0

Important Note After a hardware reset the pin SDS1 is "low" until the SPCR is written to for the first time. From that moment on, the function taken on by these pins depends on the state of the IOM Mode Select bit IMS (ADF2 register).

SPU **Software Power-Up**.
Used in TE mode only.

If SQXR:CFS = 1, before activating the ISDN S interface in TE mode the SPU and SQXR:IDC bits have to be set to "1" and then cleared again:

After a subsequent CISQ interrupt (C/I code change; ISTA) and reception of the C/I code "PU" (Power-Up indication in TE mode) the reaction of the processor would be:

– to write an activate request command as C/I code in the CIX0 register.

– to reset the SPU and SQXR:IDC bits and wait for the following CISQ interrupt.

TLP **Test Loop**

When set to 1 the IDP1 and IDP0 lines are internally connected together, and the times T1 and T2 are reduced (cf. TIMR).

C1C1, C1C0 Channel 1 Connect

Determines which of the two channels B1 or IC1 is connected to register C1R and/or B1CR, for monitoring, test-looping and switching data to/from the processor.

		C1R		B1CR	
C1C1	**C1C0**	**Read**	**Write**	**Read**	**Application(s)**
0	0	IC1	–	B1	B1 monitoring + IC1 monitoring
0	1	IC1	IC1	B1	B1 monitoring + IC1 looping from/to IOM
1	0	–	B1	B1	B1 access from/to S_0; transmission of a constant value in B1 channel to S_0.
1	1	B1	B1	–	B1 looping from S_0; transmission of a variable pattern in B1 channel to S_0.

C2C1, C2C0 Channel 2 Connect

Determines which of the two channels B2 or IC2 is connected to register C2R and/or B2CR, for monitoring, test-looping and switching data to/from the processor.

		C2R		B2CR	
C2C1	**C2C0**	**Read**	**Write**	**Read**	**Application(s)**
0	0	IC2	–	B2	B2 monitoring + IC2 monitoring
0	1	IC2	IC2	B2	B2 monitoring + IC2 looping from/to IOM
1	0	–	B2	B2	B2 access from/to S_0; transmission of a constant value in B2 channel to S_0.
1	1	B2	B2	–	B2 looping from S_0; transmission of a variable pattern in B2 channel to S_0.

4.2.2 Command/Indication Receive 0 CIR0 Read Address 31_H

Value after reset: $7C_H$

7						0
SQC	BAS		CODR0		CIC0	CIC1

SQC S/Q Channel Change

A change in the received 4-bit S-channel (TE or LT-T mode) has been detected. The new code can be read from the SQRR. This bit is reset by a read of the SQRR.

BAS Bus Access Status

Indicates the state of the TIC bus:

0: the ISAC-S TE itself occupies the D and C/I channel
1: another device occupies the D and C/I channel

CODR0 C/I Code 0 Receive

Value of the received Command/Indication code. A C/I code is loaded in CODR0 only after being the same in two consecutive IOM frames and the previous code has been read from CIR0.
(**refer to chapter 3.3.2**)

CIC0 C/I Code 0 Change

A change in the received Command/Indication code has been recognized. This bit is set only when a new code is detected in two consecutive IOM frames. It is reset by a read of CIR0.

CIC1 C/I Code 1 Change

A change in the received Command/Indication code in IOM channel 1 has been recognized. This bit is set when a new code is detected in one IOM frame. It is reset by a read of CIR0.

CIC1 is only used if terminal mode is selected.

Note: The BAS and CODR0 bits are updated every time a new C/I code is detected in two consecutive IOM frames.

If several consecutive valid new codes are detected and CIR0 is not read, only the first and the last C/I code (and BAS bit) is made available in CIR0 at the first and second read of that register, respectively.

4.2.3 Command/Indication Transmit 0 CIX0 Write Address 31_H

Value after reset: $3F_H$

7						0
RSS	BAC		CODX0		1	1

RSS Reset Source Select

Only valid if the terminal specific functions are activated (STCR:TSF).

0: Subscriber or Exchange Awake

As reset source serves:

– a falling edge on the EAW line (External Subscriber Awake)

– a C/I code change (Exchange Awake).

A logical zero on the EAW line activates also the IOM-interface clock and frame signal, just as the SPU-bit (SPCR) does.

1: Watchdog Timer

The expiration of the watchdog timer generates a reset pulse.

The watchdog timer will be reset and restarted, when two specific bit combinations are written in the ADF1 register within the time period of 128 ms (see also ADF1 register description).

After a reset pulse generated by the ISAC-S TE and the corresponding interrupt (WOV, SAW or CISQ) the actual reset source can be read from the ISTA and EXIR register.

BAC Bus Access Control

Only valid if the TIC-bus feature is enabled (MODE:DIM2-0).

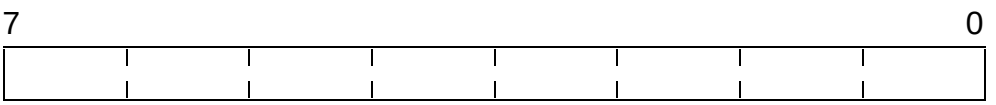
If this bit is set, the ISAC-S TE will try to access the TIC bus to occupy the C/I channel even if no D-channel frame has to be transmitted. It should be reset when the access has been completed to grant a similar access to other devices transmitting in that IOM channel.

Note: Access is always granted by default to the ISAC-S TE/ICC with TIC-bus address (TBA2-0, STCR register) "7", which has the lowest priority in a bus configuration.

CODX0 C/I Code 0 Transmit

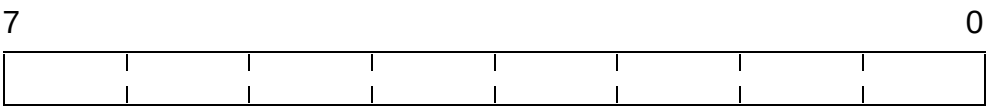
Code to be transmitted in the C/I channel / C/I channel 0.
(**refer to chapter 3.3.2**)

4.2.4 MONITOR Receive Channel 0 MOR0 Read Address 32_H



Contains the MONITOR data received in IOM MONITOR channel/ MONITOR channel 0 according to the MONITOR channel protocol.

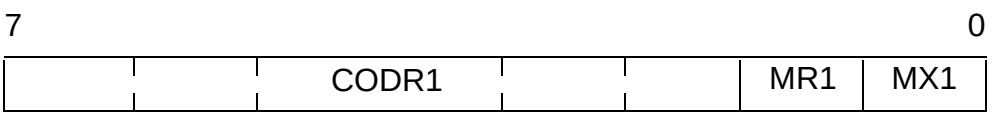
4.2.5 MONITOR Transmit Channel 0 MOX0 Write Address 32_H



Contains the MONITOR data to be transmitted in IOM MONITOR channel/ MONITOR channel 0 according to the MONITOR channel protocol.

4.2.6 Command/Indication Receive 1 CIR1 Read Address 33_H

Value after reset: FF_H



CODR1 C/I Code 1 Receive

Bits 7-2 of C/I channel 1

MR1 MR Bit

Bit 1 of C/I channel 1

MX1 MX Bit

Bit 0 of C/I channel 1

4.2.7 Command/Indication Transmit 1 CIX1 Write Address 33_H

Value after reset: FF_H

7						0
		CODX1			1	1

CODX1 C/I Code 1 Transmit

Bits 7-2 of C/I channel 1

4.2.8 MONITOR Receive Channel 1 MOR1 Read Address 34_H

7							0

Contains the MONITOR data received in IOM channel 1 according to the MONITOR channel protocol.

4.2.9 MONITOR Transmit Channel 1 MOX1 Write Address 34_H

7							0

Contains the MONITOR data to be transmitted in IOM channel 1 according to the MONITOR channel protocol.

4.2.10 Channel Register 1 C1R Read/Write Address 35_H

7							0

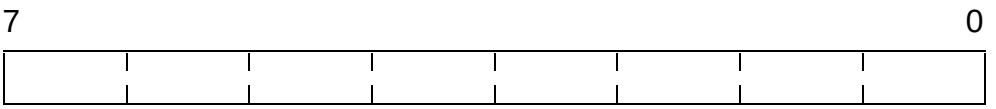
Contains the value received/transmitted in IOM channel B1 or IC1, as the case may be (cf. C1C1, C1C0, SPCR register).

4.2.11 Channel Register 2 C2R Read/Write Address 36_H

7							0

Contains the value received/transmitted in IOM channel B2 or IC2, as the case may be (cf. C2C1, C2C0, SPCR register).

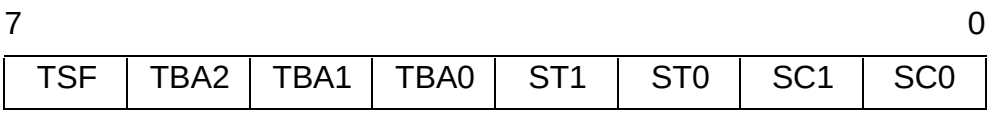
4.2.12 B1-Channel Register B1CR Read Address 37_H



Contains the value received in IOM channel B1, if programmed (see C1C1, C1C0, SPCR register).

4.2.13 Synchronous Transfer Control RegisterSTCR Write Address 37_H

Value after reset: 00_H



TSF Terminal Specific Functions

0: No terminal specific functions

1: The terminal specific functions are activated, such as

– Watchdog Timer

– Subscriber/Exchange Awake (EAW).

In this case the EAW line is always an input signal which can serve as a request signal from the subscriber to initiate the awake function in a terminal.

A falling edge on the EAW line generates an SAW interrupt (EXIR).

When the RSS bit in the CIX0 register is zero, a falling edge on the EAW line (Subscriber Awake) or a C/I code change (Exchange Awake) initiates a reset pulse.

When the RSS bit is set to one a reset pulse is triggered only by the expiration of the watchdog timer (see also CIX0-register description).

Note: The TSF bit will be cleared only by a hardware reset.

TBA2-0 TIC-Bus Address

Defines the individual address for the ISAC-S TE on the IOM TIC bus (**see chapter 2.3.6**).

This address is used to access the C/I- and D-channel on the IOM.

Note: One device liable to transmit in C/I- and D-fields on the IOM should always be given the address value "7".

ST1 Synchronous Transfer 1

When set, causes the ISAC-S TE to generate an SIN-interrupt status (ISTA register) at the beginning of an IOM frame.

ST0 Synchronous Transfer 0

When set, causes the ISAC-S TE to generate an SIN-interrupt status (ISTA register) at the middle of an IOM frame.

SC1 Synchronous Transfer 1 Completed

After an SIN interrupt the processor has to acknowledge the interrupt by setting the SC1 bit before the middle of the IOM frame, if the interrupt was originated from a Synchronous Transfer 1 (ST1). Otherwise an SOV interrupt (EXIR register) will be generated.

SC0 Synchronous Transfer 0 Completed

After an SIN interrupt the processor has to acknowledge the interrupt by setting the SC0 bit before the start of the next IOM frame, if the interrupt was originated from a Synchronous Transfer 0 (ST0).

Otherwise an SOV interrupt (EXIR register) will be generated.

Note: ST0/1 and SC0/1 are useful for synchronizing MP accesses and receive/transmit operations.

4.2.14 B2-Channel Register B2CR Read Address 38_H

Used only in terminal mode (SPCR:SPM = 0).

Contains the value received in the IOM channel B2, if programmed (see C2C1, C2C0, SPCR register).

4.2.15 Additional Feature Register 1 ADF1 Write Address 38_H

Value after reset: 00_H

7							0
WTC1	WTC2	TEM	PFS	IOF	0	0	ITF

WTC1, 2 Watchdog Timer Control 1, 2

After the watchdog timer mode has been selected (STCR:TSF = CIX0:RSS = 1) the watchdog timer is started.

During every time period of 128 ms the processor has to program the WTC1- and WTC2 bit in the following sequence:

	WTC1	**WTC2**
1. 2.	1 0	0 1

to reset and restart the watchdog timer.

If not, the timer expires and a WOV interrupt (EXIR) together with a reset pulse is generated.

TEM Test Mode

In test mode (TEM = 1, PFS = 0) all layer-1 functions are disabled and the ISAC-S TE behaves like an ICC (PEB 2070) device.

PFS Prefilter Select

These bits together determine the pre-filter delay compensation and the test mode (layer 1 disabled) of the ISAC-S TE, as follows:

TEM	**PFS**	**Effect**
0	0	No pre-filter (0 delay)
0	1	Pre-filter delay compensation 520 ns
1	1	Pre-filter delay compensation 910 ns
1	0	Test mode (layer 1 disabled)

IOF **IOM OFF.** Used in terminal mode (SPCR:SPM = 0).

0: IOM interface is operational

1: IOM interface is switched off (DCL, FSC1, IDP0/1, BCL high impedance).

Note: IOF should be set to "1" if external devices connected to the IOM interface should be "disconnected" e.g. for power saving purposes or for not disturbing the internal IOM connection between layer 2 and layer 1. However, the internal operation is independent of the IOF bit.

ITF Inter-Frame Time Fill

Selects the inter-frame time fill signal which is transmitted between HDLC frames.

0: idle (continuous 1 s),

1: flags (sequence of patterns: "0111 1110")

Note: In TE applications with D-channel access handling (collision resolution), the only possible inter-frame time fill signal is idle (continuous 1 s). Otherwise the D channel on the S/T bus cannot be accessed.

4.2.16 Additional Feature Register 2 ADF2 Read/Write Address 39_H

Value after reset: 00_H

7							0
IMS	0	0	0	ODS	D1C2	D1C1	D1C0

IMS IOM Mode Selection

IOM-2 interface mode is selected when IMS = 1.

ODS Output Driver Selection

Tristate drivers (1) or open drain drivers (0) are used for the IOM interface.

D1C2-0 Data Strobe Control

These bits determine the polarity of the two independent strobe signals SDS1 as follows:

D1C2	D1C1	D1C0	SDS1
0	0	0	always low
0	0	1	high during B1
0	1	0	high during B2
0	1	1	high during B1 + B2
1	0	0	always low
1	0	1	high during IC1
1	1	0	high during IC2
1	1	1	high during IC1 + IC2

The strobe signals allow standard combos or data devices to access a programmable channel.

4.2.17 MONITOR Status Register MOSR Read Address $3A_H$

Value after reset: 00_H

7							0
MDR1	MER1	MDA1	MAB1	MDR0	MER0	MDA0	MAB0

MDR1 MONITOR Channel 1 Data Received

MER1 MONITOR Channel 1 End of Reception

MDA1 MONITOR Channel 1 Data Acknowledged

The remote end has acknowledged the MONITOR byte being transmitted.

MAB1 MONITOR Channel 1 Data Abort

MDR0 MONITOR Channel 0 Data Received

MER0 MONITOR Channel 0 End of Reception

MDA0 MONITOR Channel 0 Data Acknowledged

The remote end has acknowledged the MONITOR byte being transmitted.

MAB0 MONITOR Channel 0 Data Abort

4.2.18 MONITOR Control Register MOCR Write Address $3A_H$

Value after reset: 00_H

7							0
MRE1	MRC1	MXE1	MXC1	MRE0	MRC0	MXE0	MXC0

MRE1,0 MONITOR Receive Interrupt Enable (IOM channel 1,0)

MONITOR interrupt status MDR1/MDR0, MER1/0 generation is enabled (1) or masked (0).

MRC1,0 MR Bit Control (IOM Channel 1,0)

Determines the value of the MR bit:

0: MR always "1". In addition, the MDR1/MDR0 interrupt is blocked, except for the first byte of a packet (if MRE1/0 = 1).

1: MR internally controlled by the ISAC-S TE according to MONITOR channel protocol. In addition, the MDR1/MDR0 interrupt is enabled for all received bytes according to the MONITOR channel protocol (if MRE1 0 = 1).

MXE1,0 **MONITOR Transmit Interrupt Enable** (IOM channel 1,0)

MONITOR interrupt status MDA1/0, MAB1/0 generation is enabled (1) or masked (0).

MXC1,0 **MX Bit Control** (IOM channel 1,0)

Determines the value of the MX bit:

0: MX always "1".

1: MX internally controlled by the ISAC-S TE according to MONITOR channel protocol.

4.2.19 S-, Q-Channel Receive Register SQRR Read Address $3B_H$

Value after reset: $0X_H$

7							0
IDC	CFS	CI1E	SYN	SQR1	SQR2	SQR3	SQR4

IDC **Read-Back of Programmed IDC Bit** (see SQXR register)

CFS **Read-Back of Programmed CFS Bit** (see SQXR register)

CI1E **Read-Back of Programmed CI1E Bit** (see SQXR register)

SYN **Synchronization State**

The S/T receiver has synchronized to the received F_A and M bits (1) or has not (0).

SQR1-4 **Received S/Q Bits**

Received S bits in frames 1, 6, 11 and 16, respectively.

4.2.20 S, Q Channel Transmit Register SQXR Write Address $3B_H$

Value after reset: $0F_H$

7							0
IDC	CFS	CI1E	SQIE	SQX1	SQX2	SQX3	SQX4

IDC IOM Direction Control

0: Master (normal) mode
Layer 2 transmits IOM channel 0 and 2 on IDP1, channel 1 on IDP0.

1: Slave (test) mode
Layer 2 transmits IOM channel 0, 1 and 2 on IDP1.

Note: Also refer to chapter 2.3.2

CFS Configuration Select

This bit determines clock relations and recovery on S/T and IOM interfaces.

0: The IOM interface clock and frame signals are always active, "Power-Down" state included.

The states "Power-Down" and "Power-Up" are thus functionally identical except for the indication: PD = 1111 and PU = 0111.

With the C/I-command Timing (TIM) the processor can enforce the "Power-Up" state.

With C/I-command Deactivation Indication (DIU) the "Power-Down" state is reached again.

However, it is also possible to activate the S-interface directly with the C/I-command Activate Request (AR 8/10/L) without the TIM command.

1: The IOM-interface clock and frame signals are normally inactive Power-Down").

For activating the S interface the "Power-Up" state can be induced by software (SPU bit in SPCR register).

After that the S interface can be activated with the C/I-command Activate Request (AR 8/10/L).

The "Power-Down" state can be reached again with the C/I command-Deactivation Indication (DIU).

Note: After reset the IOM interface is always active. To reach the "Power-Down" state the CFS bit has to be set.

CI1E **C/I Channel 1 Interrupt Enable**

Interrupt generation of CIR0:CIC1 is enabled (1) or masked (0).

SQIE **S-, Q-Interrupt Enable**

Generation of CIR0:SQC status (and the accompanying CISQ interrupt is enabled (1) or masked (0).

SQX1-4 **Transmitted Q Bits**

transmitted F_A bits in frames 1, 6, 11 and 16, respectively.

5 Electrical Characteristics

Absolute Maximum Ratings

Parameter	Symbol	Limit Values	Unit
Voltage on any pin with respect to ground	V_S	– 0.4 to V_{DD} + 0.4	V
Ambient temperature under bias	T_A	0 to 70	°C
Storage temperature	T_{stg}	– 65 to 125	°C
Maximum voltage on V_{DD}	V_{DD}	6	V

Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Line Overload Protection

The maximum input current (under overvoltage conditions) is given as a function of the width of a rectangular input current pulse (**figure 54**).

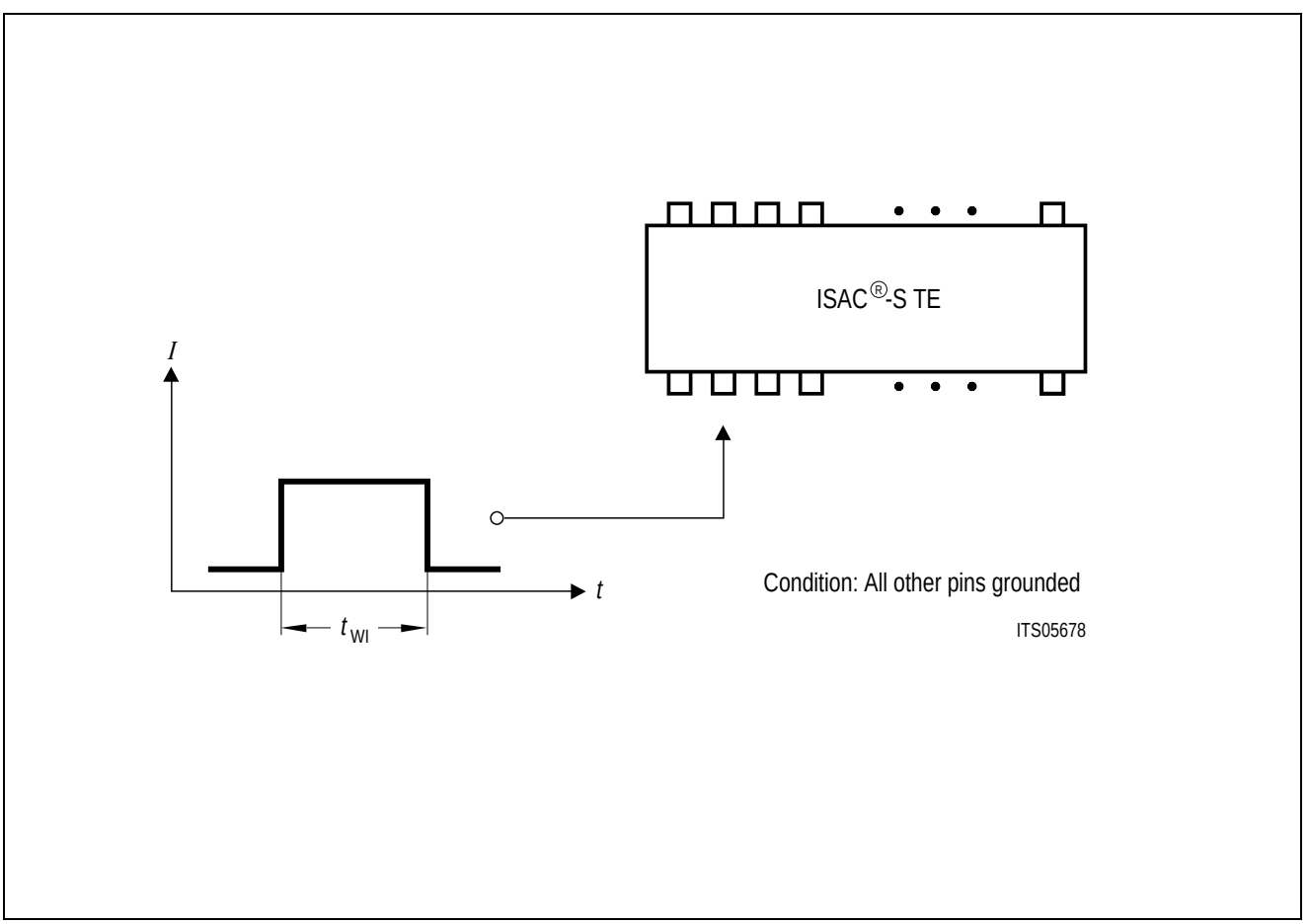


Figure 54
Test Condition for Maximum Input Current

Transmitter Input Current

The destruction limits for negative input signals are given in **figure 55**. $R_i \geq 2\ \Omega$.

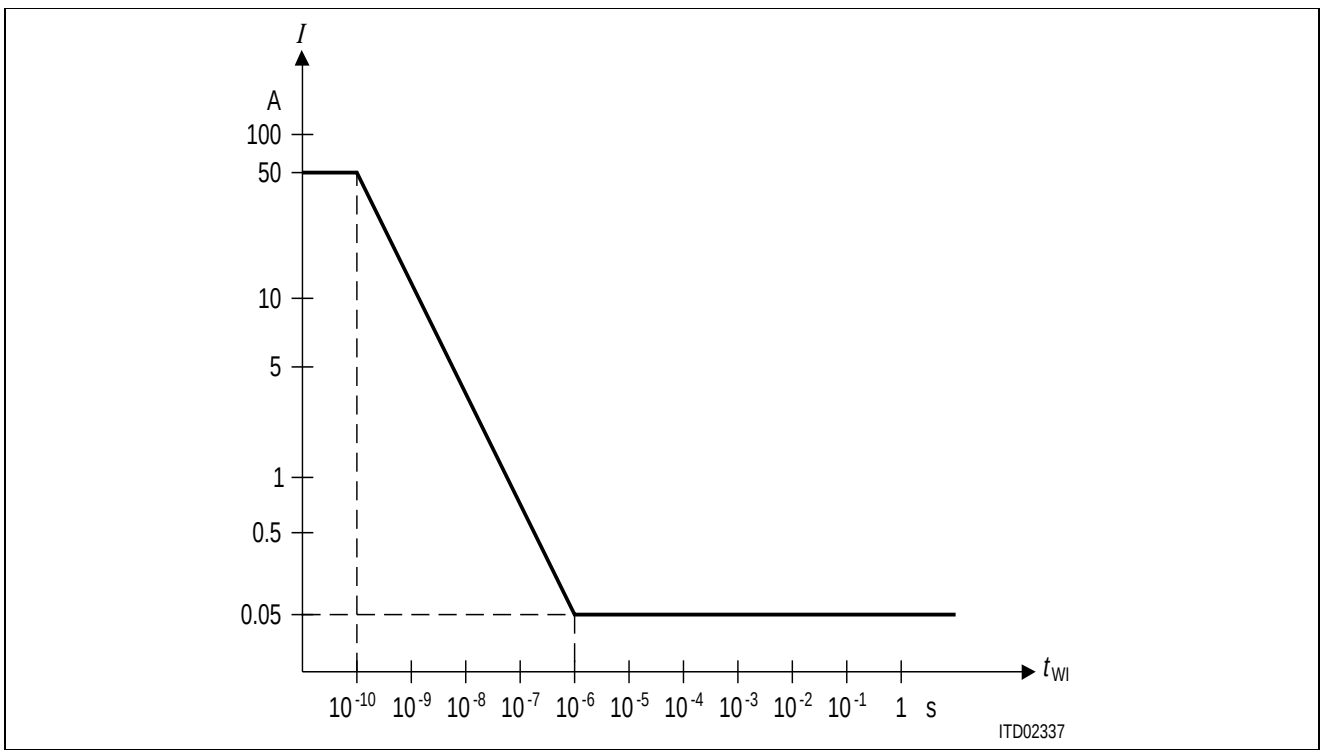


Figure 55

The destruction limits for positive input signals are given in **figure 56**. $R_i \geq 200\ \Omega$.

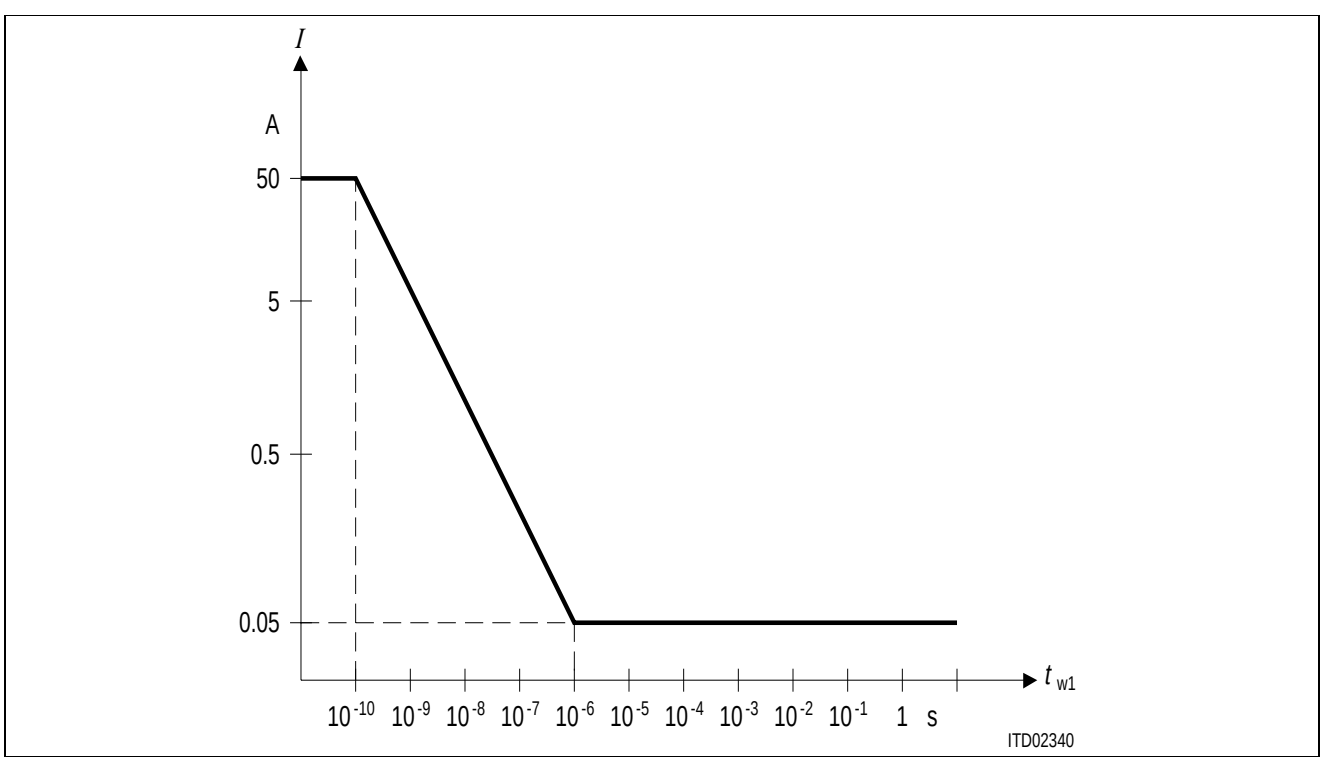
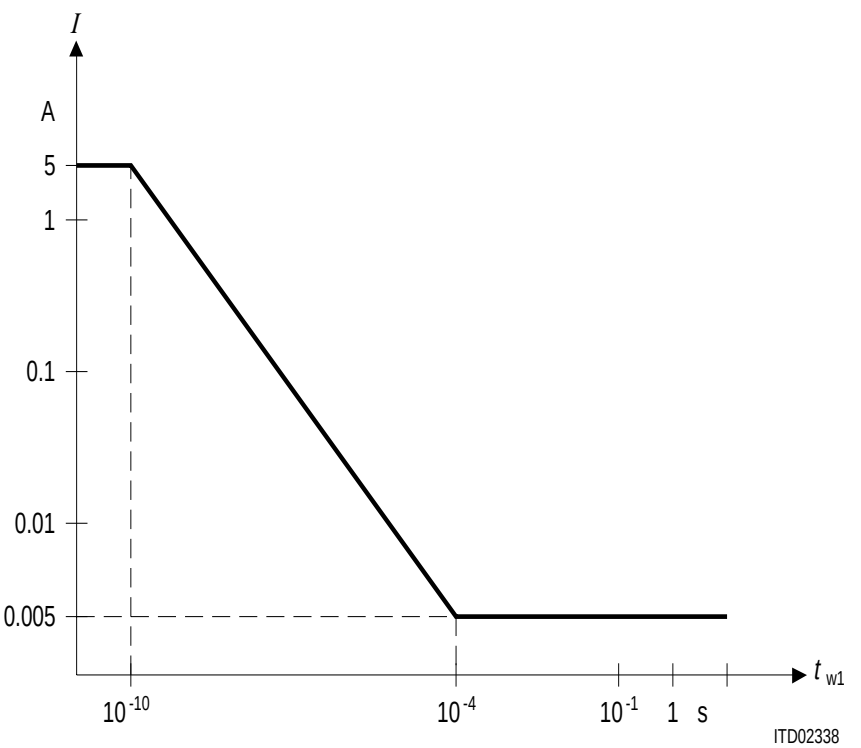


Figure 56

Receiver Input Current

The destruction limits are given in **figure** . $R_i \geq 300\ \Omega$.



DC Characteristics

T_A = 0 to 70 °C; V_{DD} = 5 V ± 5 %, V_{SSA} = 0 V, V_{SSD} = 0 V

Parameter		Symbol	Limit Values		Unit	Test Condition		Remarks
			min	max				
L-input voltage		V_{IL}	– 0.4	0.8	V			All pins except SX1,2, SR1,2
H-input voltage		V_{IH}	2.0	V_{DD} + 0.4	V			
L-output voltage L-output voltage (IDP0)		V_{OL} V_{OL1}		0.45 0.45	V V	I_{OL} = 2 mA I_{OL} = 7 mA		
H-output voltage H-output voltage		V_{OH} V_{OH}	2.4 V_{DD} – 0.5		V V	I_{OH} = – 400 µA I_{OH} = – 100 µA		
Power supply current	power down	I_{CC}		1.5	mA		V_{DD} = 5 V Inputs at V_{SS} / V_{DD} No output loads except SX1,2 (50 Ω load)	
	oper-ational (96 kHz)			17	mA	DCL = 1536 kHz		
	Emergency B1 = FF_H, B2 = FF_H, D = 1			7.7	mA	DCL = 1536 kHz		
	B1 = FF_H, B2 = FF_H, D = Flag			7.95	mA	DCL = 1536 kHz		
	B1 = 55_H, B2 = FF_H, D = Flag			8.75	mA	DCL = 1536 kHz		
	B1 = 00_H, B2 = FF_H, D = Flag			10	mA	DCL = 1536 kHz		
Input leakage current Output leakage current		I_{LI} I_{LO}		10 10	µA µA	0 V < V_{IN} < V_{DD} to 0 V 0 V < V_{OUT} < V_{DD} to 0 V		All pins except CP/BCL, X2, SX1,2, SR1,2, A0, A1, A3, A4

DC Characteristics

T_A = 0 to 70 °C; V_{DD} = 5 V ± 5 %, V_{SSA} = 0 V, V_{SSD} = 0 V (cont'd)

Parameter	Symbol	Limit Values		Unit	Test Condition	Remarks
		min	max			
Input leakage current internal pull-down	I_{LIPD}		120	µA	0 V < V_{IN} < V_{DD} to 0 V	A0, A1, A3, A4, CP/BCL, X2
Absolute value of output pulse amplitude (VSX2 – VSX1)	V_X	2.03 2.10	2.31 2.39	V V	R_L = 50 Ω[1)] R_L = 400 Ω[1)]	SX1,2
Transmitter output current	I_X	7.5	13.4	mA	R_L = 5.6 Ω[1)]	
Transmitter output impedance	R_X	10 0		kΩ Ω	Inactive or during binary one during binary zero R_L = 50 Ω	
Receiver output voltage	V_{SR1}	2.35	2.6	V	I_O < 5 µA	SR1,2
Receiver threshold voltage $V_{SR2} - V_{SR1}$	V_{TR}	225	375	mV	Dependent on peak level	

Note: [1)] Due to the transformer, the load resistance seen by the circuit is four times R_L.

Capacitances

T_A = 25 °C, V_{DD} = 5 V ± 5 %, V_{SSA} = 0 V, V_{SSD} = 0 V, f_c =1 MHz, unmeasuredpins grounded.

Parameter	Symbol	Limit Values		Unit	Remarks
		min.	max.		
Input capacitance I/O capacitance	C_{IN} $C_{I/O}$		7 7	pF pF	All pins except SR1,2
Output capacitance against V_{SSA}	C_{OUT}		10	pF	SX1,2
Input capacitance	C_{IN}		7	pF	SR1,2
Load capacitance	C_L		50	pF	XTAL1,2

Recommended Oscillator Circuits

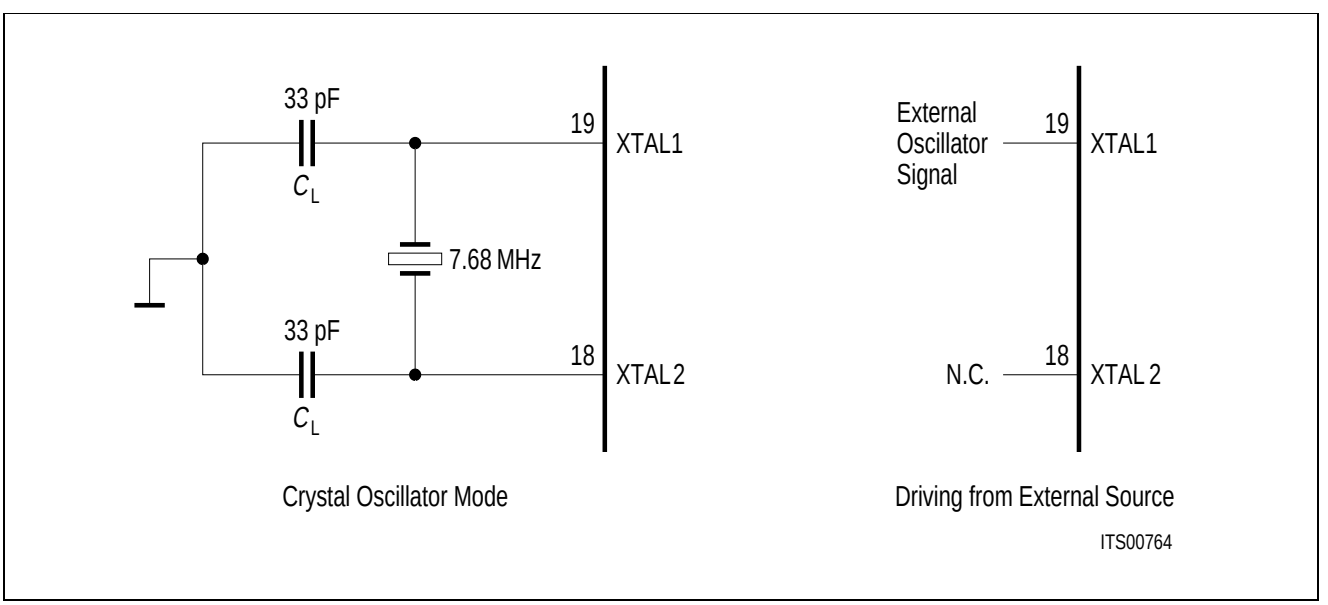


Figure 57
Oscillator Circuits

Crystal Specification

Parameter	Symbol	Limit Values	Unit
Frequency	f	7.680	MHz
Frequency calibration tolerance		max. 100	ppm
Load capacitance	C_L	max. 50	pF
Oscillator mode		fundamental	

Note: The load capacitance C_L depends on the recommendation of the crystal specification. Typical values for C_L are 22 …33 pF.

XTAL1 Clock Characteristics (external oscillator input)

Parameter	Limit Values	
	min.	max.
Duty cycle	1:2	2:1

AC Characteristics

T_A = 0 to 70 °C, V_{DD} = 5 V ± 5%

Inputs are driven to 2.4 V for a logical "1" and to 0.4 V for a logical "0". Timing measurements are made at 2.0 V for a logical "1" and 0.8 V for a logical "0". The AC-testing input/output waveforms are shown in **figure 58.**

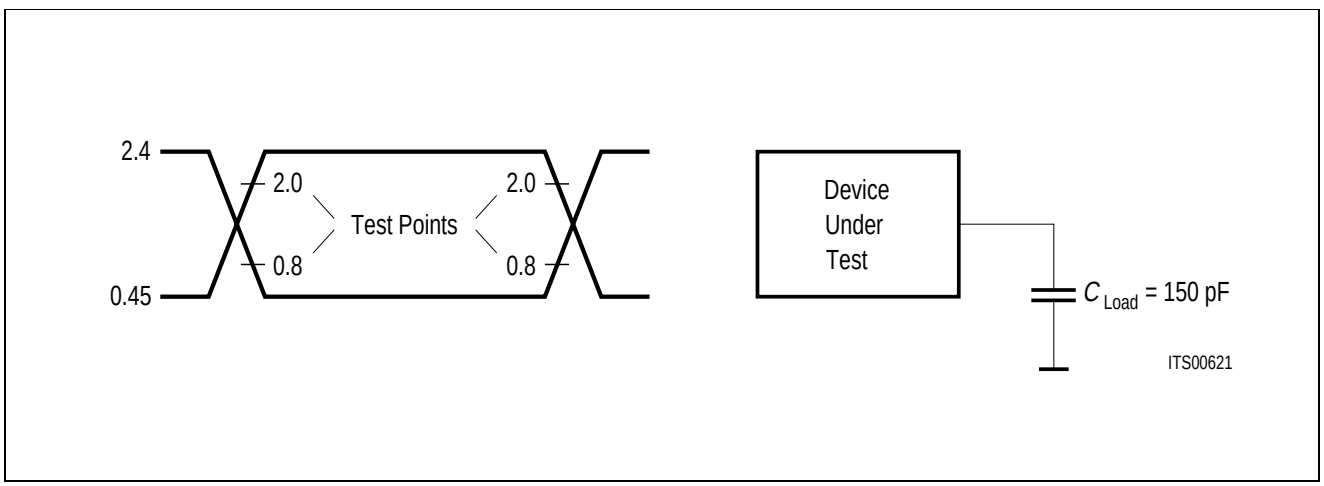


Figure 58
Input/Output Waveform for AC Tests

Microprocessor Interface Timing

Siemens/Intel Bus Mode

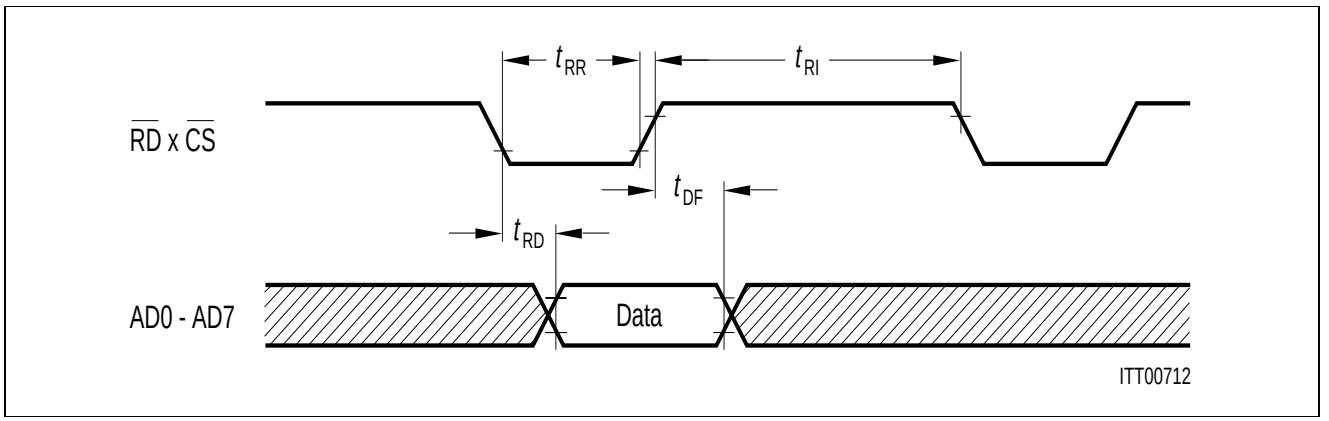


Figure 59
Microprocessor Read Cycle

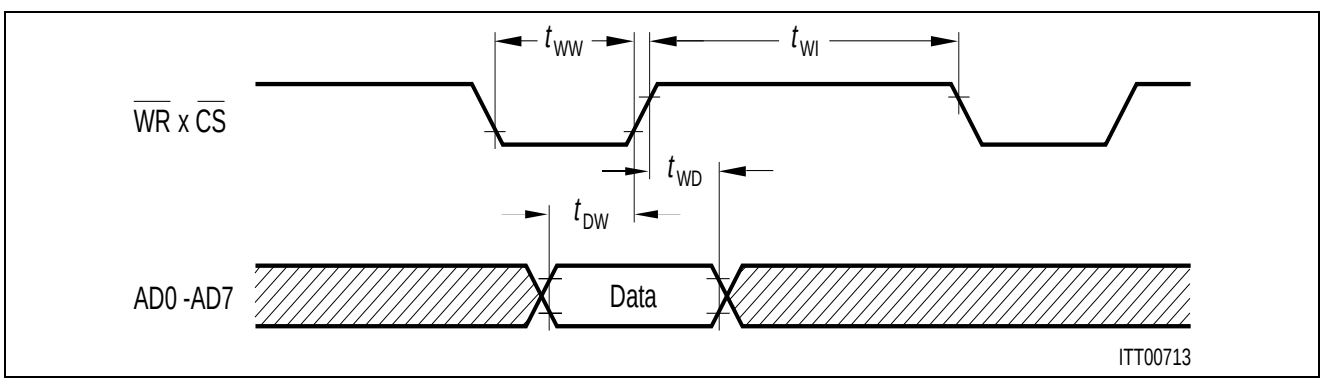


Figure 60
Microprocessor Write Cycle

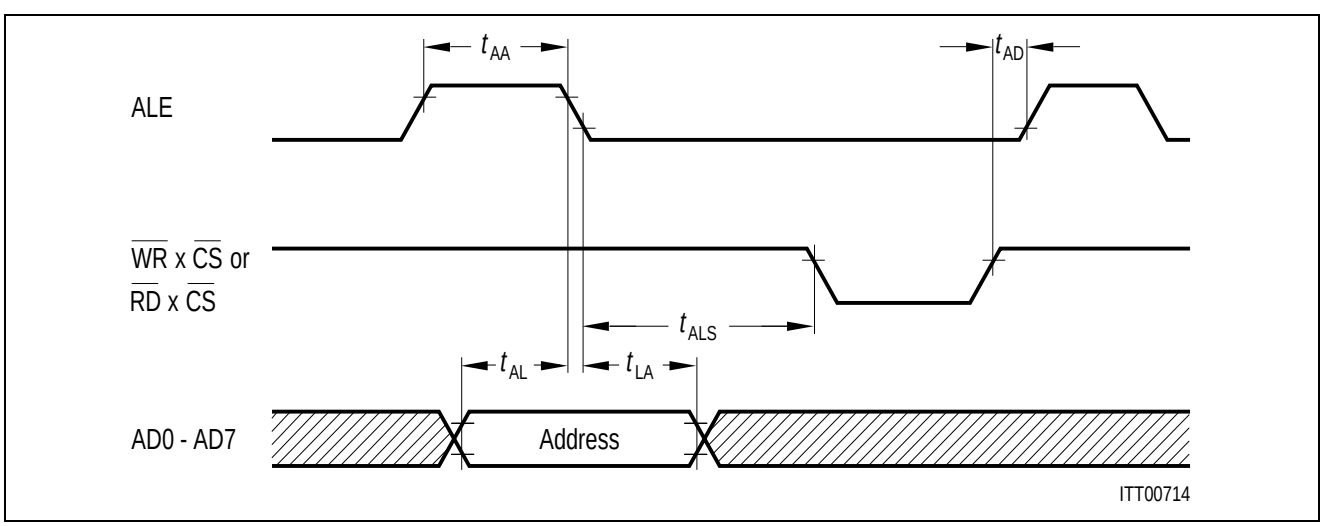


Figure 61
Multiplexed Address Timing

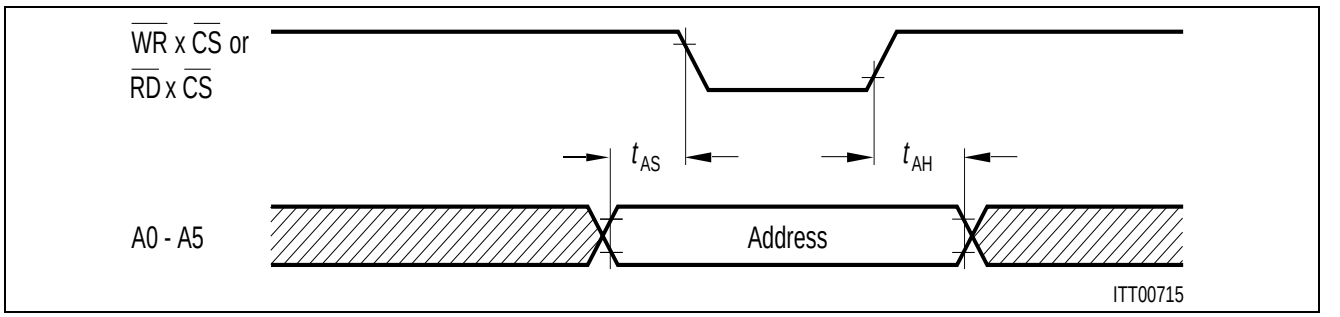


Figure 62
Non-Multiplexed Address Timing

Motorola Bus Mode

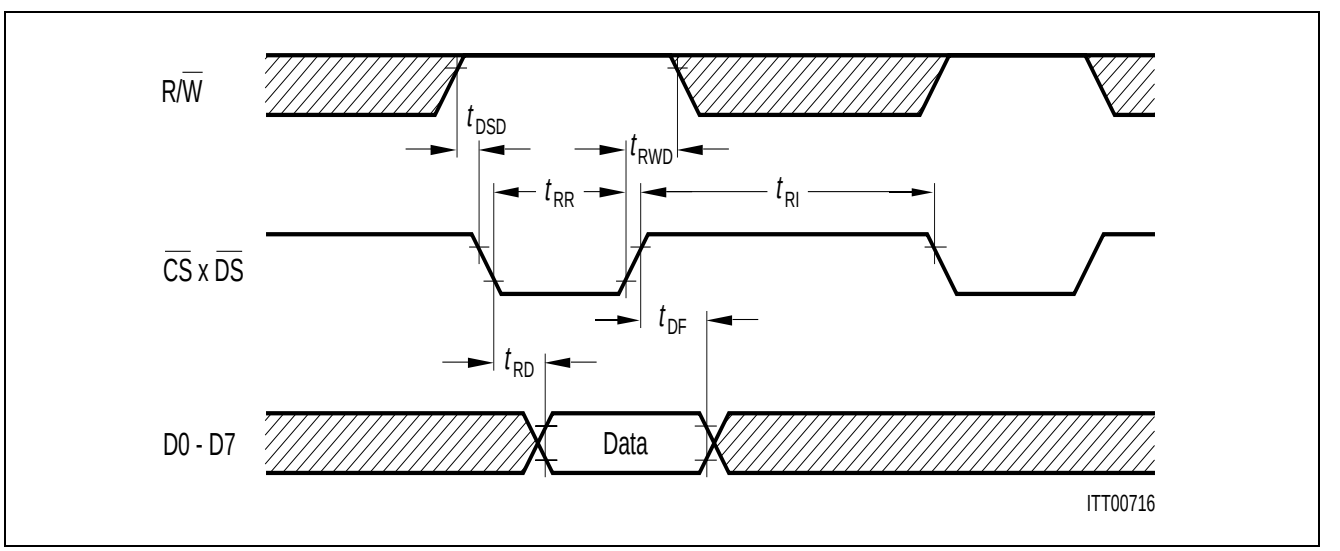


Figure 63
Microprocessor Read Timing

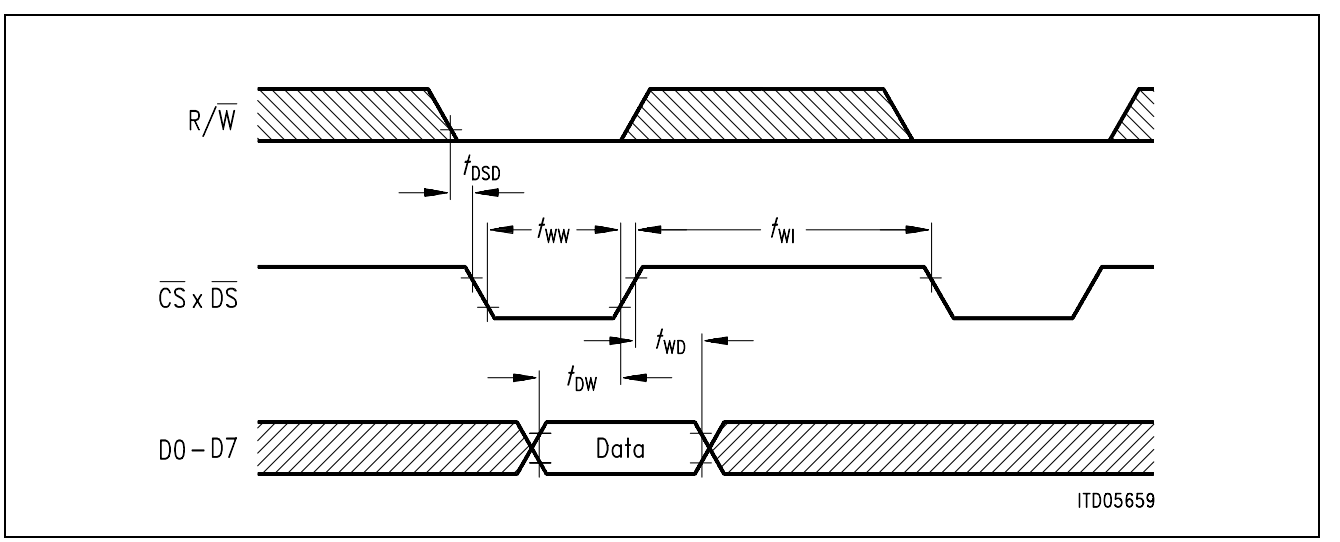


Figure 64
Microprocessor Write Cycle

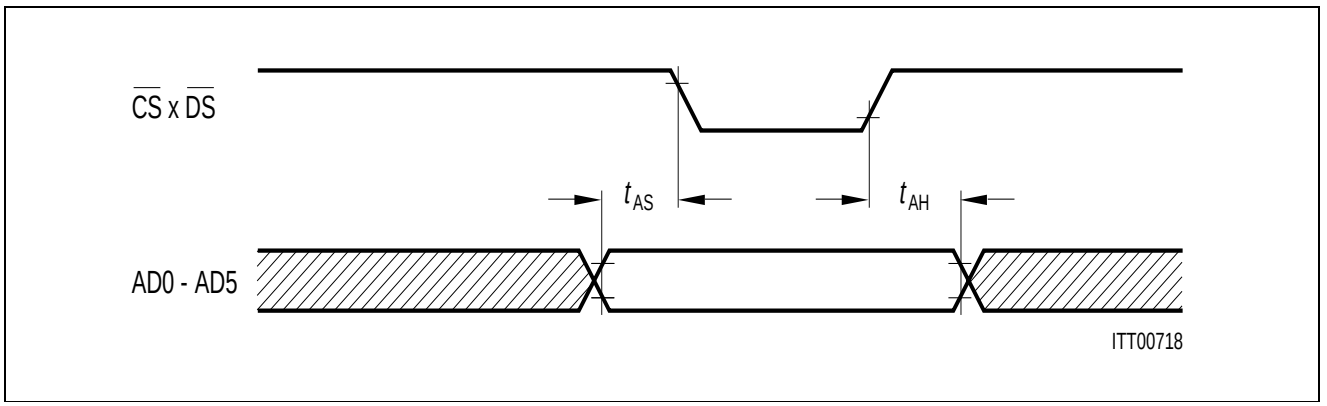


Figure 65
Non-Multiplexed Address Timing

Microprocessor Interface Timing

Parameter	Symbol	Limit Values		Unit
		min.	max.	
ALE pulse width	t_{AA}	50		ns
Address setup time to ALE	t_{AL}	15		ns
Address hold time from ALE	t_{LA}	10		ns
Address latch setup time to $\overline{WR}$, $\overline{RD}$	t_{ALS}	0		ns
Address setup time	t_{AS}	25		ns
Address hold time	t_{AH}	10		ns
ALE guard time	t_{AD}	15		ns
$\overline{DS}$ delay after $\overline{RW}$ setup	t_{DSD}	0		ns
$\overline{RD}$ pulse width	t_{RR}	110		ns
Data output delay from $\overline{RD}$	t_{RD}		110	ns
Data float from $\overline{RD}$	t_{DF}		25	ns
$\overline{RD}$ control interval	t_{RI}	70		ns
$\overline{W}$ pulse width	t_{WW}	60		ns
Data setup time to $\overline{W} \times \overline{CS}$	t_{DW}	35		ns
Data hold time from $\overline{W} \times \overline{CS}$	t_{WD}	10		ns
$\overline{W}$ control interval	t_{WI}	70		ns

Serial Interface Timing

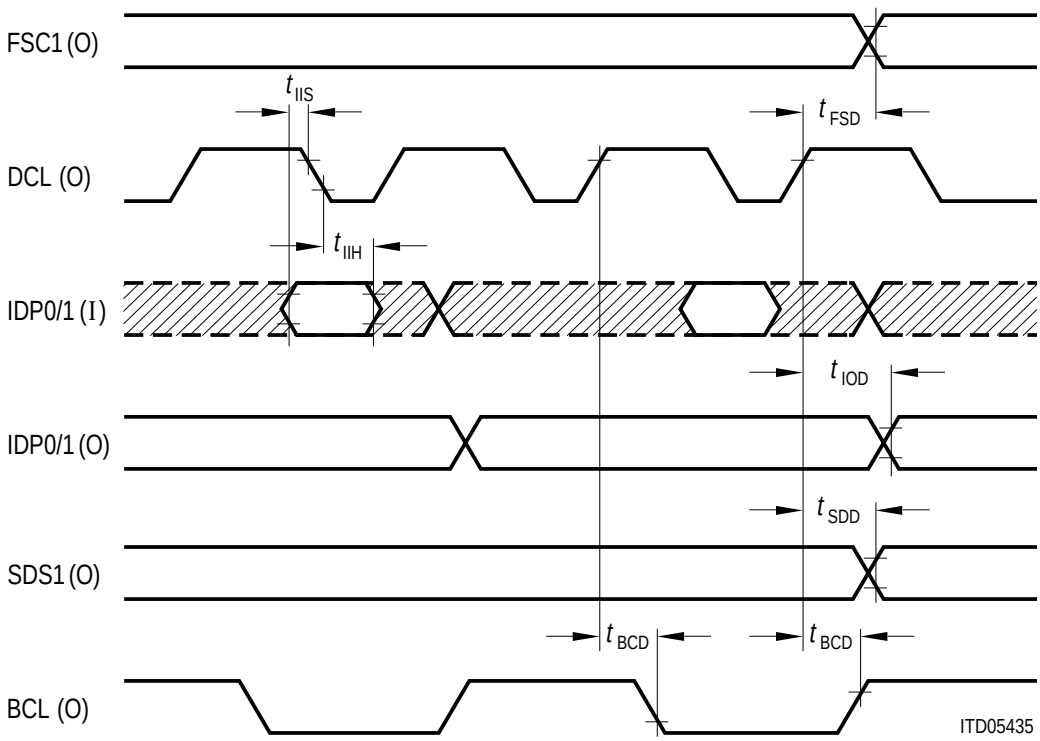


Figure 66
IOM® Timing (TE mode)

IOM® Timing

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
IOM output data delay	t_{IOD}	20	100	ns	IOM-2
IOM input data setup	t_{IIS}	20		ns	IOM-2
IOM input data hold	t_{IIH}	20		ns	
FSC1 strobe delay	t_{FSD}	– 20	20	ns	
Strobe signal delay	t_{SDD}		120	ns	
Bit clock delay	t_{BCD}	– 20	20	ns	

HDLC Mode (ADF2: IMS = 0, ADF1: TEM = 1, MODE: DIM2 – 0 = 101 – 111)

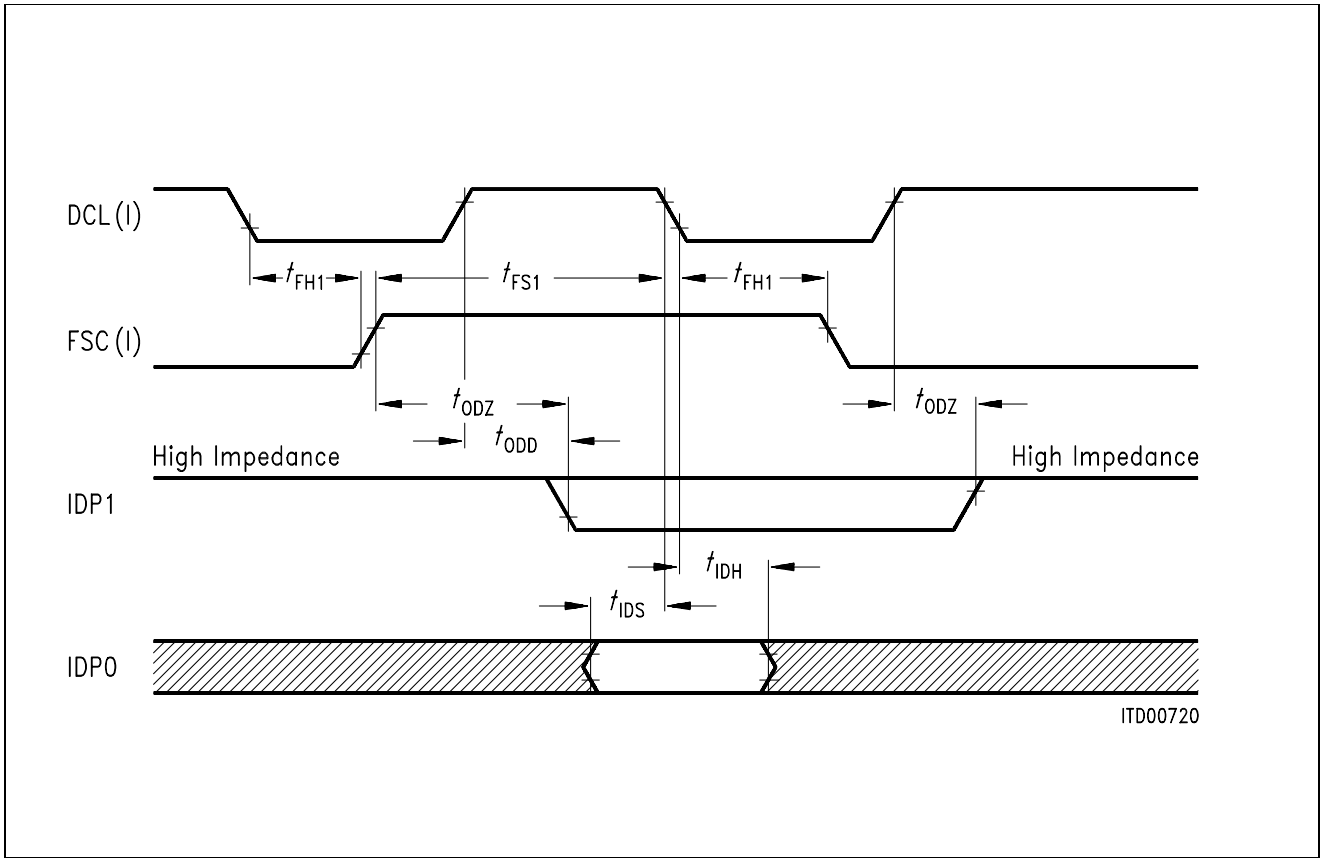


Figure 67
FSC1 (strobe) Characteristics

HDLC Mode Timing

Parameter	Symbol	Limit Values		Unit
		min.	max.	
FSC1 set-up time	t_{FS1}	100		ns
FSC1 hold time	t_{FH1}	30		ns
Output data from high impedance to active	t_{OZD}		80	ns
Output data from active to high impedance	t_{ODZ}		40	ns
Output data delay from DCL	t_{ODD}	20	100	ns
Input data setup	t_{IS}	10		ns
Input data hold	t_{DH}	30		ns

Clock Timing

The clocks are summarized in **table 20**, with the respective duty ratios.

Table 20
ISAC®-S TE Clock Signals (IOM®-2 mode)

Application	DCL	FSC1	BCL	SDS1
TE	o:1536 kHz* 3:2	o:8 kHz* 1:2	o:768 kHz* 1:1	o:8 kHz 1:11 2:10

The 1536-kHz clock is phase-locked to the receive S signal, and derived using the internal DPLL and the 7.68 MHz ± 100 ppm crystal.

A phase tracking with respect to "S" is performed once in 250 µs. As a consequence of this DPLL tracking, the "high" state of the 1536-kHz clock may be either reduced or extended by one 7.68-MHz period (duty ratio 2:2 or 4:2 instead of 3:2) once every 250 µs. Since the other signals are derived from this clock, the "high" or "low" states may likewise be reduced or extended by the same amount once every 250 µs.

The phase relationships of the clocks are shown in **figure 68.**

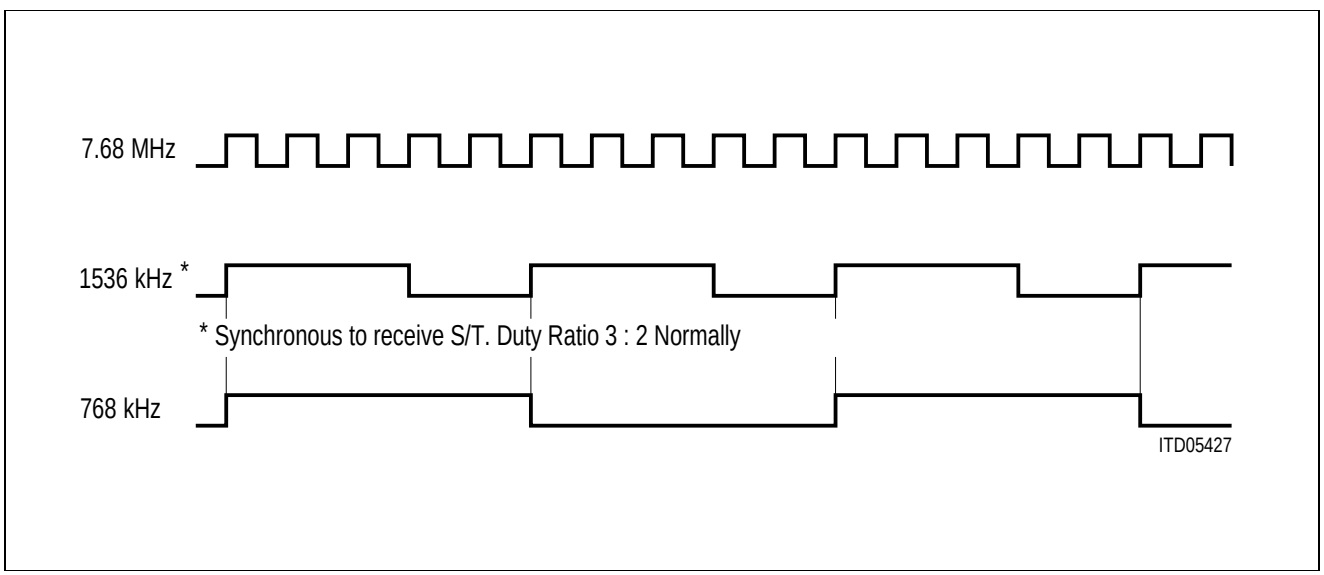


Figure 68
Phase Relationships of ISAC®-S TE Clock Signals

*) Synchronous to receive "S" line

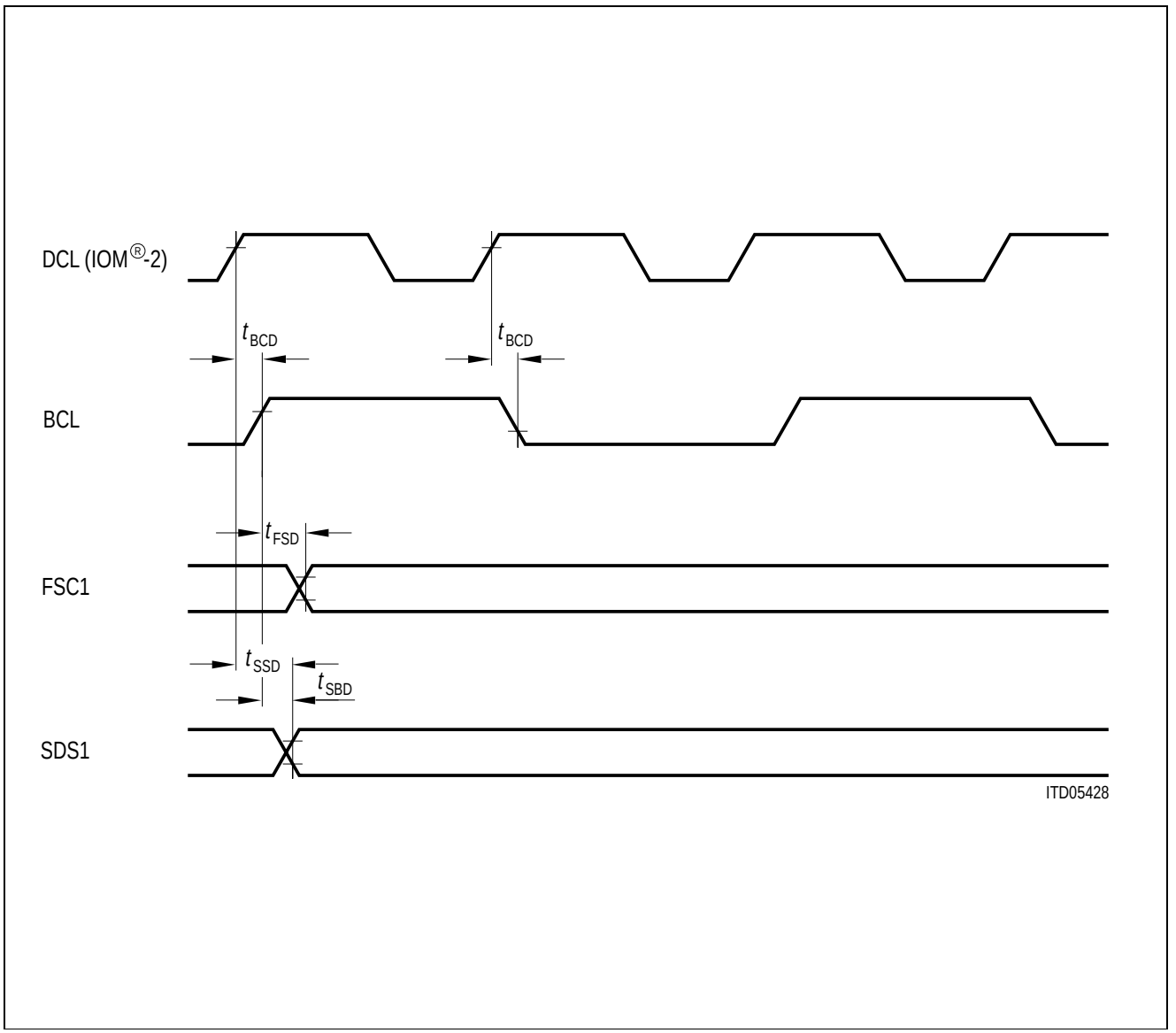


Figure 69
Timing Relationships between ISAC®-S TE Clock Signals

Table 21

Parameter	Symbol	Limit Values		Unit	Condition
		min.	max.		
Bit clock delay	t_{BCD}	– 20	20	ns	IOM-2
SDS1 delay from DCL	t_{SDD}		120	ns	IOM-2
SDS1 delay from BCL	t_{SBD}		120	ns	IOM-2

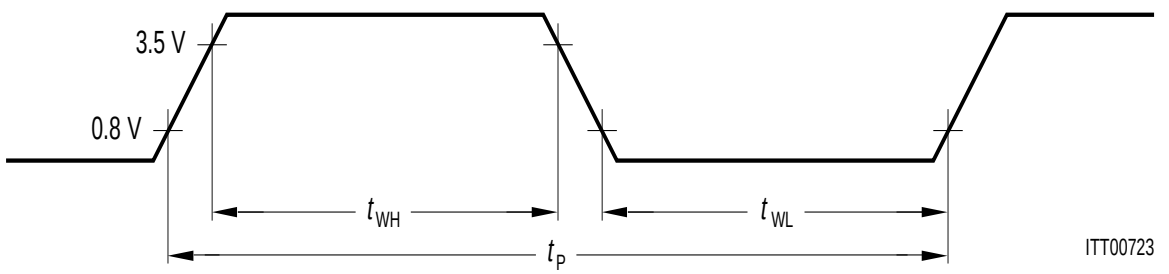


Figure 70
Definition of Clock Period and Width

Table 22
DCL-Clock Characteristics (IOM®-2)

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
(TE) 1536 kHz	t_{PO}	520	651	782	ns	osc ± 100 ppm
	t_{WHO}	240	391	541	ns	osc ± 100 ppm
	t_{WLO}	240	260	281	ns	osc ± 100 ppm

Jitter

In TE mode, the timing extraction jitter of the ISAC-S conforms to CCITT Recommendation I.430 (– 7 % to + 7 % of the S-interface bit period).

Description of the Receive PLL (RPLL) of the ISAC-S TE

The receive PLL performs phase tracking each 250 µs after detecting the phase between the F/L transition of the receive signal and the recovered clock. Phase adjustment is done by adding or subtracting 130 ns to or form a 1.536-MHz clock cycle. The 1.536-MHz clock is than used to generate any other clock synchronized to the line.

During (re)synchronization an internal reset condition may effect the 1.536-MHz clock to have high or low times as short as 130 ns. After the S/T-interface frame has achieved the synchronized state (after three consecutive valid pairs of code violations) the FSC output is set to a specific phase relationship, thus causing once an irregular FSC timing.

Reset

Table 23
Reset Signal Characteristics

Parameter	Symbol	Limit Values	Unit	Test Condition
		min.		
Length of active high state	t_{RST}	4	ms	Power-on/Power-Down to Power-Up (Standby)
		2 x DCL clock cycles		During Power-Up (Standby)

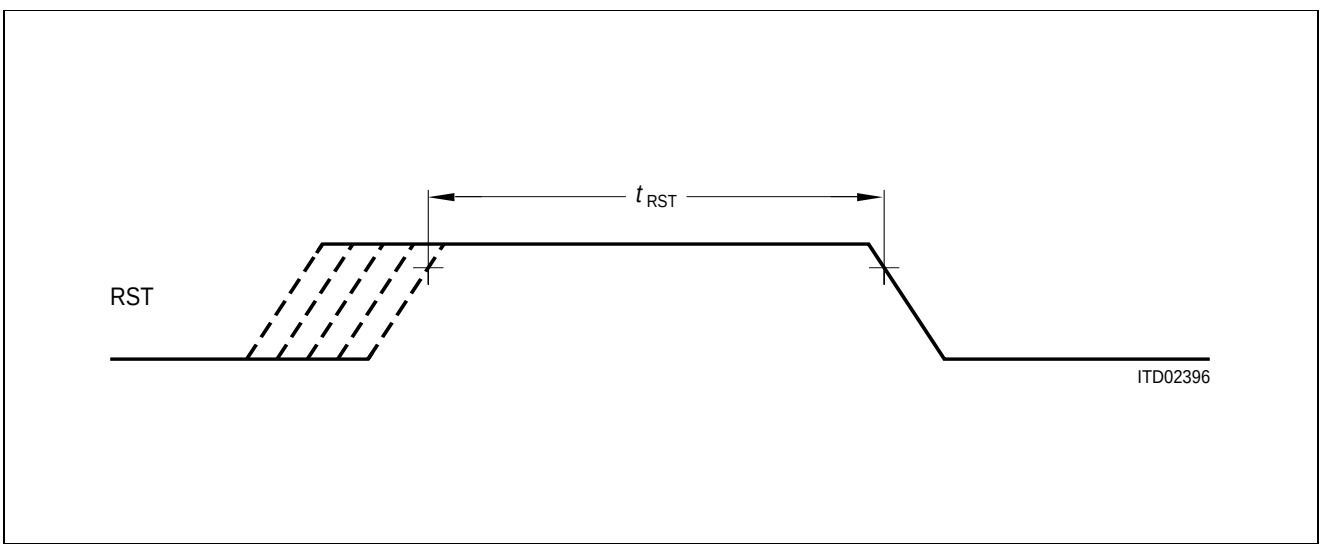


Figure 71
Reset

6 ISAC®-S TE Low Level Controller

The following paragraphs outline the functionality and structure of a software driver example for the ISAC-S TE. This example is based on the Siemens Low Level Controllers (LLC's) for Basic Access IC which are available in C source code. The ISAC-S TE software driver will be also referred to as LLC or ISAC-S LLC.

It should be noted that the ISAC-S LLC does not access the complete palette of device functions but rather a subset of them. For example not all message transfer modes are supported. Please refer to paragraph 'Architecture and Functions' for a more detailed description.

The ISAC-S LLC presented here has been successfully tested in the Siemens ISDN PC development system. Correct operation with a higher layer software has been verified by using the Siemens ISDN-Software Development and Evaluation System (SIDES) and the Siemens ISDN-Operational Software (IOS).

The ISAC-S LLC also apply for the ISAC®-S TE with the limitation of TE functionality only. There is no adaptation in the listing to this limitation.

6.1 Architecture and Functions

The ISAC-S TE LLC may be divided into two major parts, one for layer 1 control, the 'SBC part' and one for directing the HDLC-controller operations, the 'ICC part'. The naming conventions 'SBC part' and 'ICC part' have been introduced for two reasons: The first is that the ISAC-S TE may be viewed as the one-chip integration of the Siemens ISDN Communications Controller, PEB 2070 ICC, and the S-Bus Interface Circuit, PEB 2080 SBC. The second is that the SIPB-mainboard firmware, the basis for this example software, actually uses the same code to control either an ISAC-S TE or an ICC-SBC combination.

The ISAC-S TE LLC consists of **driver functions** and **interrupt server**. The driver functions are implemented as a set of C functions which are responsible for interpreting hardware related commands from the higher layers and carrying out the appropriate actions at the hardware level. Driven by hardware interrupts, the interrupt server analyses the hardware event and informs the higher software layers of that event.

It should be noted that this implementation has attempted to remove as many protocol specific functions as possible from the LLC and to locate them instead in the higher layer protocol itself. This has the advantage of making the LLC- more general and less likely to be in need of re-programming for different protocols.

OPERATING SYSTEM and
Higher Level Protocol Software

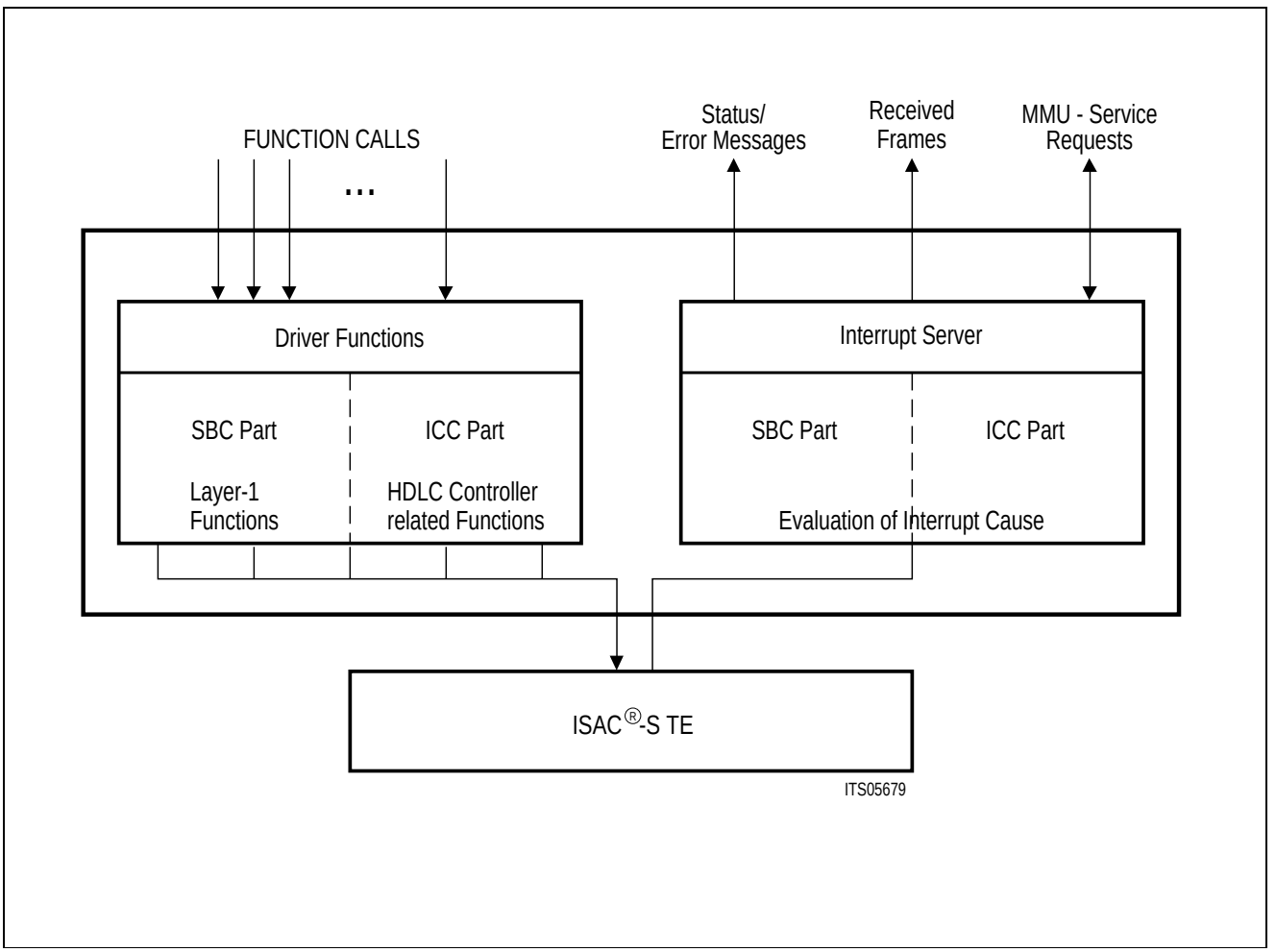


Figure 73
LLC Architecture

The ISAC-S TE LLC supports following standard functions:

– Initialization of the SBC (layer 1) part.
– Activation of layer 1.
– Deactivation of layer 1.
  HDLC-controller initialization.
  The following HDLC-controller message transfer modes are supported:

auto-mode:	full two byte address compare, LAPD support.
non-auto mode:	full two byte address compare.
transparent mode 3:	high byte address compare; called 'TRANSPARENT' mode in the LLC.
transparent mode 2:	no address compare; called 'EXTENDED TRANSPARENT' mode in the LLC.

HDLC framing with two byte address field is assumed.

– HDLC-frame transmission.
– Programming of TEI and SAPI values.
– HDLC-transceiver control.
– Local test loop switching.

The LLC assumes that the ISAC-S TE is operating in an IOM-2 TE configuration.

In addition to the ISAC-S TE standard functions supporting the ISDN-basic access, the ISAC-S TE contains optional, terminal specific functions. These terminal specific functions (watchdog and external awake) are not supported by this LLC.

6.2 Summary of LLC Functions

6.2.1 Layer-1 Related Functions

Mnemonic	Purpose
ActL1_SBC	Layer-1 activation.
DeaL1_SBD	Layer 1 deactivation.
ArlL1_SBC	Activation of a local loop.
EnaClk_SBC	Enable clocking in power down mode.
InitL1_SBC	Layer-1 initialization and reset.
ResL1_SBC	Layer-1 reset.
IntL1_SBC	Handling of CISQ interrupts.

The layer 1 related functions call DECODE_L1_STATUS to report a L1 status change to a higher layer software.

6.2.2 HDLC-Controller Related Functions

Mnemonic	Purpose
InitLay2_ICC	HDLC-controller initialization.
Loop_ICC	Testloop activation at the serial outputs of the IOM interface.
ResetHDLC_ICC	HDLC-transceiver reset.
RecReady_ICC	Setting the HDLC receiver ready or not ready.
SendFrame_ICC	HDLC-frame transmission.
StoreSAPI_ICC	SAPI programming.
StoreTEI_ICC	TEI programming.
Int_ICC	Handling of XPR, RSC, TIN and EXI interrupts.
Rx_ICC	Handling of RPF and RME interrupts.

6.2.3 External Functions

The LLC-program listing shows some references to external functions (indicated by an 'IMPORT' declaration). These functions are used by the LLC but are not part of it. These external functions must be provided by the operating system or a higher layer protocol software.

MMU_req ()

By calling MMU_req the ISAC-S TE LLC requests memory for the temporary storage of a received data frame. The memory management unit (MMU) of the operating system has to provide a memory buffer of the required size (max. 260 bytes).

MMU_free ()

MMU_free is the counterpart to MMU_req. The operating system can release a previously allocated memory buffer.

STRING_IN () and STRING_OUT ()

STRING_IN and STRING_OUT are assembler written functions for fast input and output of data frames from/to the ISAC-S FIFO.

ENTERNOINT () and LEAVENOINT ()

ENTERNOINT and LEAVENOINT are called to disable and enable all system interrupts in time critical sections.

Decode_S_Frame_BASIC ()

Decode_S_Frame_BASIC is called by the LLC-interrupt server to transfer a received HDLC S frame to a higher layer protocol software.

Following information is passed to Decode_S_Frame_BASIC:

'pei': 1-byte value identifying the performed address recognition. The bits 0, 1 and 2 of 'pei' represent the bits TA, SA0 and SA1 of the ISAC-S' RSTA register.

'sapi': 1-byte value representing the received HDLC SAPI address byte. Bit 1 of 'sapi' is the C/R bit value (RSTA:CR). The most significant 6 bits of 'sapi' are 0 in auto-mode, non-auto mode and transparent mode.

'tei': 1-byte value representing the received HDLC TEI address byte. 'tei' is 0 in auto-mode and non-auto mode.

'ctrl': 2-byte value representing the contents of the received HDLC-control field.

'frame_status': 1 byte value

= 0×00: frame is valid.
= 0×80: frame is mutilated (last byte of two byte control field missing).
= 0×82: frame is too long. S frame with I field.

'M128': 1-byte value. 0 in modulo 8 operating mode (1-byte control field), 1 in modulo 128 operating mode (2-byte control field). For correct decoding of 'ctrl' above.

Decode_U_Frame_BASIC ()

Decode_U_Frame_BASIC is called by the LLC-interrupt server to transfer a received HDLC U frame to a higher layer protocol software.

Following information is passed to Decode_U_Frame_BASIC:

'pei': (refer to Decode_S_Frame).

'sapi': (refer to Decode_S_Frame).

'tei': (refer to Decode_S_Frame).

'ctrl': 1-byte value representing the contents of the received HDLC control field.

PassLongFrame_BASIC ()

PassLongFrame_BASIC is called by the LLC-interrupt server to transfer received HDLC I and UI frames to a higher layer protocol software.

The LLC passes a pointer to a structure (FRAME_PASS) containing information about the received frame to PassLongFrame_BASIC. Please refer to the following paragraph for a description of this structure.

6.3 LLC-Code Elements

6.3.1 Structures

The Structure ’ISAC’

As the various routines in the LLC require facilities to store information about the device they control, the global variabel ’pt’ of the type ’ISAC’ has been introduced. The type ’ISAC’ is a structure containing imperative information elements. These information elements are listed below:.

Status Information

pt_op_mode	operating mode of the ISAC-S TE HDLC controller (auto-mode, non-auto mode…)
pt_state	Flags of 'pt_state' indicate the various device states.
pt_ModulMode	hardware configuration (TE or NT-S)

I/O buffer related elements

These elements are used when the HDLC data is transmitted or received. In both the transmit and receive directions additional RAM is required to store data on an intermediate basis. This buffer will be referred to as the data frame. Related information is stored in the following elements:

Transmit buffer pointers

pt_tx_start	pointer to the starting point of the data frame for transmission
pt_tx_curr	pointer to the present byte to be sent

Receive buffer pointers

pt_rx_start	pointer to the starting point of the receive data frame.
pt_rx_curr	pointer to the next free position in the receive buffer.

Data byte counters

pt_tx_cnt	number of bytes yet to be transmitted
pt_rx_cnt	number of bytes currently received

The following elements are used to store the type of frame:

pt_rx_frame	type of received frame.
pt_tx_frame	type of transmitted frame.

The Structure 'FRAME_PASS'

The variable 'fp' of the type FRAME_PASS is used when the LLC-interrupt server has received a valid HDLC I or UI frame. A pointer to 'fp' is passed to PassLongFrame_BASIC. FRAME_PASS contains all information about the received HDLC frame. Following elements are used:

mmu_buff	start of MMU buffer which is used for the temporary storage of that HDLC frame.
start_of_i_data	Start of the I-data field in this MMU buffer.
i_data_cnt	Number of bytes in the I-data field.
Two_byte_cf	0 for a one byte HDLC control field, 1 for a two byte HDLC-control field.
ctrl_field	HDLC-control field.
pei	1-byte value identifying the performed address recognition. The bits 0, 1 and 2 of 'pei' represent the bits TA, SA0 and SA1 of the ISAC-S' RSTA register.
frame	Type of HDLC frame; 0 = I frame, 1 = UI frame.
sapi	Received HDLC SAPI address byte. Bit 1 of 'sapi' is the C/R-bit value (RSTA:CR). The most significant 6 bits of 'sapi' are 0 in auto-mode, non-auto mode and transparent mode.
tei	Received HDLC TEI-address byte. 'tei' is 0 in auto-mode and non-auto mode.

6.3.2 Definitions and Naming Conventions

Public functions are declared with an EXPORT (only for better readability). External functions are imported using an IMPORT which is the redefinition of C's 'extern'. Any function which is only used locally is declared with a LOCAL (= 'static').

6.3.2.1 Type Definitions

For reference here is a list of the type definitions used in the LLC's.

type definitions	meaning
BYTE	one byte value
WORD	word = two byte value
FPTR	far pointer to BYTE

6.3.2.2 Macro Definitions

Error conditions and other states of the ISAC-S TE must be reported to higher layers. This reporting is realized by a few macros which are executed when such conditions are detected. These macros can be mapped to any form of message a higher layer software requires. Any kind of immediately necessary actions may be defined in those macros as well. By using such constructs the code can be kept compact and clearly readable.

Layer 1 Related Status Message

DECODE_L1_STATUS	for L1 status (IC-channel indication) decoding.

HDLC Controller Related Status and Error Messages

CRC_ERROR	CRC error.
MISSING_ACKNOWLEDGE	A 'Missing HDLC I-frame acknowledge' is generated when an acknowledge message for a previously sent I frame is outstanding and the HDLC-message transfer mode is changed from auto-mode to non-auto mode. An outstanding acknowledge is indicated by the ISAC-S TE in register STAR2 ('timer recovery status' and 'waiting for acknowledge' bits).
MMU_ERROR	No memory available to store incoming frame.
N201_ERROR	N201 error, HDLC frame is too long.
PEER_REC_READY	Peer receiver ready.
PEER_REC_BUSY	Peer receive busy.
PROTOCOL_ERROR	Protocol error (PCE interrupt).
REC_FRAME_OVERFLOW	Receive frame overflow.
REC_DATA_OVERFLOW	Receive data overflow (RDO interrupt).
REC_ABORTED	Receive aborted (RAB interrupt).
TX_ACKNOWLEDGE	Transmit frame acknowledge.
TIN_ERROR	TIN interrupt, status enquiry.
TX_DATA_UNDERRUN	Transmit data underrun (XDU interrupt).
XMR_ERROR	Transmit message repeat indication (XMR interrupt).

Following macros are used when a 'timer recovery status' (register STAR2, bit TREC) is recognized.

ENABLE_TREC_STATUS_CHECK	enable 'timer recovery status' check procedure.
DISABLE_TREC_STATUS_CHECK	disable 'timer recovery status' check procedure.

6.4 Interrupts

Int_ICC is to be called in the case of ISAC-S TE interrupts. The following interrupts are handled directly in Int_ICC:

'Transmit pool ready' interrupt (ISTA:XPR)

'Timer' interrupt (ISTA:TIN).

'Receive Status Change' interrupt (ISTA:RSC).

'Extended' interrupt (ISTA:EXI).

The 'Receive Pool Full' (ISTA:RPF) and 'Receive Message End' (ISTA:RME) interrupts are handled by function RX_ICC. The 'CI or SQ channel change' interrupt (ISTA:CISQ) is handled by IntL1_SBC.

Please note that the following interrupts are not handled by the interrupt service routine described here:

ISTA:SIN	(synchronous transfer interrupt)
EXIR:SOV	(synchronous transfer overflow)
EXIR:MOS	(monitor status) is handled by external functions which are not part of this description.
EXIR:SAW	(subscriber awake)
EXIR:WOV	(watchdog timer overflow)

6.5 LLC-Routine Reference

6.5.1 ISAC®-S TE Layer-1 Functions: The SBC Part

ActL1_SBC ()

Initiates layer-1 activation. The appropriate CI code (activate request) is written to the CI channel if the layer 1 is not already activated. ActL1_SBC then returns with ACK_DONE. The subsequent status changes of the SBC will cause CI-channel status change (CISQ) interrupts and these will be evaluated in the layer-1 interrupt service routine IntL1_SBC.

If the layer 1 is already activated nothing is carried out but ActL1_SBC calls DECODE_L1_STATUS to report the activated state.

DeaL1_SBC ()

Initiates layer 1 deactivation. The appropriate CI code is written to the CI channel if the layer 1 is not already deactivated. The subsequent layer 1 status changes cause CI channel status change (CISQ) interrupts and these will be evaluated in the layer 1 interrupt service routine IntL1_SBC.

If the layer 1 is already deactivated nothing is carried out but DeaL1_SBC calls DECODE_L1_STATUS to report the deactivated state.

ArL1_SBC ()

Activates a local loop in the SBC. The appropriate CI code (activate request loop) is written to the SBC. ArL1_SBC returns with ACK_DONE. The subsequent status changes of the SBC will generate CISQ interrupts and these will be evaluated and reported in the layer-1 interrupt service routine IntL1_SBC.

EnaClk_SBC ()

EnaClk_SBC enables clocking in TE configurations when the layer 1 is in power-down state. If first tests if clocks are actually there. If there are clocks the function returns with FALSE. If there are no clocks (power-down state) the power-up procedure is implemented. The SPU bit in register SPCR is set. The TIM code is written to the CI channel. EnaClk_SBC waits until the power-up state (PU) is indicated before the SPU bit is reset to 0. The routine then returns with TRUE.

InitL1_SBC ()

Initializes and resets the layer-1 controller (ResL1_SBC). Timing mode 0 is set and the TIC-bus address is also programmed.

ResL1_SBC ()

This routine resets the layer 1 part of an ISAC-S. It also checks that the layer 1 part is operating correctly.

Reset procedure:

A software reset command (RS) is sent to the layer 1 part via the IOM CI0 channel. ResL1_SBC waits for the expected new state (EI) if no timeout condition occurs and issues a release command (DIU).

If the new state (EI) is not observed the ISAC-S layer 1 part will be deemed to be defective.

IntL1_SBC () Interrupt Handler

Handles the CISQ interrupts which indicate changes in the layer 1 status. The final confirmation of deactivation is carried out here. The actual layer 1 state is evaluated by reading register CIR0. The following is then carried out:

If the CI-channel indication is 'pending deactivation' state (DR), DIU is sent to deactivate the layer 1.

If the indication is an 'activation indication' (AI) the activation must be confirmed from the TE side. IntL1_SBC does it automatically by writing an 'activation request' (AR). In this way this requirement of the ISAC-S TE is transparent to the higher protocol layers.

After every CI-channel status change interrupt (CISQ) DECODE_L1_STATUS is called to report the current layer-1 state.

6.5.2 ISAC®-S TE HDLC-Controller Related Functions: The ICC Part

InitPeitab_ICC ()

Initializes the local variable 'pt'. InitPeitab_ICC is to be called once during the system initialization phase.

InitLay2_ICC ()

Initializes the HDLC controller. The function arguments allow the selection of the HDLC-controller message transfer mode (auto-mode, non-auto mode, ...), one or two byte HDLC-control field operation (modulo 8 or 128) and the setting of the ISAC-S TE internal hardware timer.

After InitLay2_ICC is called the TEI values for a Broadcast Link are programmed (TEI = FF hex). The HDLC controller is not reset.

StoreTEI_ICC ()

StoreTEI_ICC is used to program a TEI value in register TEI1 or TEI2 depending on the function argument value.

StoreSAPI_ICC ()

StoreSAPI_ICC is used to program a SAPI value in register SAP1 or SAP2 depending on the function argument value.

RecReady_ICC ()

Sets HDLC receiver ready or not ready depending on the function argument value.

ResetHDLC_ICC ()

ResetHDLC_ICC resets the HDLC controller. Status flags of the local variable ’pt’ indicating any on-going data transmissions or receptions are reset and memory buffers are released.

SendFrame_ICC ()

SendFrame_ICC initiates the transmission of HDLC frames (S, U, I, UI frames).

A frame can not be sent if the transmit path is still in use, i.e. if the previous transmission is not finished, if the timer recovery state is indicated (only for I frames) or if the XFIFO is blocked (STAR:XFW bit).

If the transmission is begun the interrupt handler (Int_ICC) will handle subsequent tasks, for example shifting remaining data bytes into the XFIFO or calling the MMU to release the memory buffer.

Loop_ICC ()

Switches testloop at the IOM interface on or off, i.e. connects internally the data upstream and data downstream lines. This is achieved through setting/resetting the TLP bit in register SPCR. If the layer-1 part does not deliver clocks while in the deactivated state the clocks will be enabled when the loop is switched on by means of EnableClk_BASIC. In the Siemens Low Level Controllers for BASIC access ICs EnableClk_BASIC is a function pointer which addresses EnaClk_SBC if an ISAC-S or SBC(X) is used. When the loop is switched off the layer 1 part will return to its normal deactivated state.

Int_ICC () Interrupt Handler

Evaluates and handles the ISAC-S TE interrupts.

Interrupt service procedure:

The bits of the interrupt status register ISTA are scanned. XPR, TIN, RSC, and EXI interrupts are handled directly by Int_ICC. For RPF and RME interrupts the function RX_ICC is called, for CISQ interrupts IntL1_SBC is called. The interrupt related actions performed are:

– XPR(transmit pool ready) interrupt, but no TIN and no PCE (EXIR:PCE) interrupt:
  a) HDLC controller reset was given previously.
  b) last transmission is finished. The XFIFO will be loaded if there are more bytes to be sent. If not, a 'transmit frame acknowledge' can be generated (if depends on the message transfer mode and some other conditions).
– TIN interrupt:
  The HDLC controller's internal timer has expired (in auto-mode only).
– RSC (receiver status change of remote station) interrupt:
  A status change of the remote station's receiver has been detected. This is reported to the higher layers.
– EXI (extended) interrupt:
  One of the six non-critical interrupts has been generated. The exact cause is read from register EXIR and reported to the higher layers.

RX_ICC () Interrupt Handler

Handles the receive pool full and receive message end (RPF and RME) interrupts if TIN and PCE (EXIR:PCE) interrupt are not indicated. Received frames are handed over to the higher software levels. Errors detected during the frame reception are reported to the higher layers.

RPF interrupt: 32 data bytes are in the RFIFO. The end of the received frame is yet to be received and the message is not complete.

RME interrupt: The receive message is complete. The RFIFO contains the last bytes of a frame greater than 32 bytes long or a complete frame. In the case of a long frame the beginning of this frame will already have been received using the RPF interrupt. Address and control field information is examined, the type of frame (HDLC U-, UI-, I- or S frame) is determined and the validity of the frame is checked. Finally the frame or a error condition message is sent to the higher layers.

Check_TREC_status_ICC ()

Check_TREC_status_ICC () is called periodically by the operating system, if 'timer recovery status' (STAR2:TREC) was detected during a previous XPR interrupt handling. A 'transmit frame acknowledge' for an HDLC I frame is generated if the TREC status is left and no timer interrupt (ISTA:TIN) is indicated.

6.6 Listing of Driver Routines

```
/*****************************************************************************/
/*                                                                           */
/*    SIEMENS ISDN-Userboard  (c) 1987-1993                                  */
/*    =====================                                                  */
/*                                                                           */
/*    Firmware:   driver functions for ICC/ISAC-S/ISAC-P                     */
/*    File    :   icc.c                                                      */
/*                                                                           */
/*****************************************************************************/

   /* Include Files                                                          */
   /* =============                                                          */


#include "def.h"
#include "basic.h"
#include "message.h"


   /* Import Functions                                                       */
   /* ================                                                       */

         /* from  crt0.asm                                                  */
IMPORT   void          STRING_IN ();
IMPORT   void          STRING_OUT ();

         /* from  basic00.c                                                 */
IMPORT   PEITAB        *GetPeitab_BASIC ();

         /* from  basic_l1.c                                                */
IMPORT   void          IntLay1_BASIC ();
IMPORT   void          ResetLay1_BASIC ();
IMPORT   int           EnableClk_BASIC ();

         /* from  basic_l2.c                                                */
IMPORT   void          PassLongFrame_BASIC ();
IMPORT   void          Decode_S_Frame_BASIC ();
IMPORT   void          Decode_U_Frame_BASIC ();

         /* from  mmu.c                                                     */
IMPORT   int           MMU_free ();
IMPORT   FPTR          MMU_req ();

         /* from  mofc.c                                                    */
IMPORT   int           IntMon_MOFC ();
IMPORT   int           Wr_IntMon_MOFC ();


   /* Export Functions                                                       */
   /* ================                                                       */

EXPORT   int           Assign_ICC ();
EXPORT   void          Check_TREC_status_ICC ();
EXPORT   int           InitLay2_ICC ();
```

```
EXPORT   void           InitPeitab_ICC ();
EXPORT   void           Int_ICC ();
EXPORT   int            Loop_ICC ();
EXPORT   int            SwitchB_ICC ();

EXPORT   int            RecReady_ICC ();
EXPORT   int            ResetHDLC_ICC ();
EXPORT   int            StoreTEI_ICC ();
EXPORT   int            StoreSAPI_ICC ();
EXPORT   int            SendFrame_ICC ();


   /* Local Functions                                                  */
   /* ===============                                                  */

LOCAL    void           RX_ICC ();



   /* Variables                                                        */
   /* =========                                                        */

IMPORT   unsigned int   interrupt_act;

   /* Function Declarations                                            */
   /* =====================                                            */

/************************************************************************/
/*                                                                      */
/*    Function: InitPeitab_ICC ()                                       */
/*    Parms   : '*pt'   pointer to the assigned PEITAB array element    */
/*              'base'  address of detected ICC/ISAC                    */
/*    purpose : initialization of the PEITAB elemtn for an ICC / ISAC-S */
/*                                                                      */
/************************************************************************/
EXPORT   void
InitPeitab_ICC (pt, base)
   register PEITAB   *pt;
   IO_PORT           base;
{
   BYTE        version;
   IO_PORT     reg_rbch = base + ICC_RBCH;

                                     /* read the ICC/ISAC-S (ISAC-P)    */
                                     /* version number                  */
                                     /* 0 for versions A1, A2, ..       */
                                     /* 1 and greater for versions      */
                                     /* 2.x [Bx] (x=1,2,3,4) and later  */
   version = inp (reg_rbch);
                                     /* and set the device identifier   */
   if (version != 0)                 /* accordingly                     */
   {
      if (pt->pt_device == PT_ICC)
         pt->pt_device = PT_ICC_B;

      if (pt->pt_device == PT_ISAC_S)
```

```
        pt->pt_device = PT_ISAC_S_B;
    }


    pt->pt_io_base = base;              /* store the base (IO) address      */

                                        /* the following structure          */
                                        /* elements store the register IO   */
                                        /* addresses (e.g. for FIFOs, ISTA, */
                                        /* MASK, etc.)                      */
    pt->pt_r_fifo  = base + ICC_FIFO;
    pt->pt_r_ista  = base + ICC_ISTA;
    pt->pt_r_mask  = base + ICC_MASK;
    pt->pt_r_star  = base + ICC_STAR;
    pt->pt_r_cmdr  = base + ICC_CMDR;
    pt->pt_r_mode  = base + ICC_MODE;
    pt->pt_r_timr  = base + ICC_TIMR;
    pt->pt_r_exir  = base + ICC_EXIR;
    pt->pt_r_xad1  = base + ICC_XAD1;
    pt->pt_r_xad2  = base + ICC_XAD2;
    pt->pt_r_sap1  = base + ICC_SAP1;
    pt->pt_r_sap2  = base + ICC_SAP2;
    pt->pt_r_rsta  = base + ICC_RSTA;
    pt->pt_r_tei1  = base + ICC_TEI1;
    pt->pt_r_tei2  = base + ICC_TEI2;
    pt->pt_r_rhcr  = base + ICC_RHCR;
    pt->pt_r_spcr  = base + ICC_SPCR;
    pt->pt_r_stcr  = base + ICC_STCR;
    pt->pt_r_cixr  = base + ICC_CIXR;   /* = CIX0/CIR0 in later versions    */
    pt->pt_r_monr  = base + ICC_MONR;   /* = MOX0/MOR0 in later versions    */
    pt->pt_r_adfr  = base + ICC_ADFR;   /* = ADF1 in later versions         */

    pt->pt_r_rbcl  = base + ICC_RFBC;   /* = RBCL in later version          */
    pt->pt_r_rbch  = base + ICC_RBCH;
    pt->pt_r_mox1  = base + ICC_MOX1;
    pt->pt_r_mocr  = base + ICC_MOCR;   /* = MOSR (read access)             */
    pt->pt_r_cix1  = base + ICC_CIX1;   /* CIX1 and CIR1 register           */
    pt->pt_r_adf2  = base + ICC_ADF2;

    pt->pt_r_rfbc  = base + ICC_RFBC;
    pt->pt_r_sfcr  = base + ICC_SFCR;
    pt->pt_r_sscx  = base + ICC_SSGX;

    pt->pt_r_sqxr  = base + ISAC_SQXR;  /* S/Q channel transmit and         */
                                        /* receive register                 */

                                        /* STAR2 register                   */
    pt->pt_r_star2 = base + ICC_STR2;

    DISABLE_TREC_STATUS_CHECK ();
}
/*****************************************************************************/
/*                                                                           */
/*    Function  : InitLay2_ICC ()                                            */
/*    Parameters:                                                            */
/*                                                                           */
```

```
/*     'pei'       0x00 D-channel controller                                */
/*                 0x40 B-channel controller (A)                            */
/*                 0x80 B-channel controller (B)                            */
/*                                                                          */
/*     'modulo'    0    modulo 8 operation                                  */
/*                 1    modulo 128 operation                                */
/*                                                                          */
/*     'mode'      operating mode. (automode, non automode, etc.)          */
/*                                                                          */
/*     'tim_mode'  value for the TIMR register (valid in auto mode only)   */
/*                 refer to the description of that register in the        */
/*                 data sheets.                                             */
/*                                                                          */
/*    Purpose:  Initialization of an ICCs (ISAC-..) HDLC controller part.  */
/*              After execution of InitLay2_ICC, the TEI values for        */
/*              the Broadcast Link are programmed.                          */
/*                                                                          */
/*    Note:  No HDLC controller reset is done.                              */
/*           Only two byte address fields are supported                     */
/*                                                                          */
/*       If the ICC (ISAC) is reprogrammed from AUTOMODE to NON - AUTOMODE */
/*       the successful transmission and acknowledgement of an I-frame     */
/*       currently sent is not assured.                                     */
/*       Switching from AUTOMODE to NON AUTOMODE causes an I frame to be   */
/*       transmitted completely by the ICC. But the transmit acknowledge   */
/*       (XPR interrupt) in NON AUTOMODE only indicates that the ICC has   */
/*       sent the frame out of its XFIFO. It indicates not the successful  */
/*       transmission of the I-frame as it is in AUTOMODE (timer super-    */
/*       vision, polling for acknowledge frames)!                           */
/*       Therefore if an I-frame is outstanding and the mode is changed    */
/*       from AUTOMODE to NON-AUTOMODE  MISSING_ACKNOWLEDGE is called to   */
/*       generate a warning message.                                        */
/*       MISSING_ACKNOWLEDGE is also called if  'timer recovery' status    */
/*       (TREC) or 'waiting for acknowledge (WFA)' is indicated.           */
/*                                                                          */
/****************************************************************************/
EXPORT int
InitLay2_ICC (pei, modulo, mode, tim_mode)
   BYTE     pei, modulo, mode, tim_mode;
{
   BYTE              mode_reg;
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))  /* request pointer to the           */
                                       /* corresponding PEITAB table       */
                                       /* element                          */
      return (ACK_NOT_SUPPORTED);

   if (modulo != 0 && modulo != 1)
      return (ACK_WRONG_PARM);

   outp (pt->pt_r_mask, 0xFF);         /* no interrupts during init.       */

   mode_reg  = inp (pt->pt_r_mode) & (MODE_HMD2 | MODE_HMD1 | MODE_HMD0);

   switch (mode)                       /* select OPERATING MODE            */
```

```
{                                   /* ******************             */

   case PT_MD_AUTO:                 /* HDLC AUTO MODE                  */
                                    /* full address recognition,       */
                                    /* internal timer mode, receiver   */
                                    /* active, 2 bytes address fields  */
                                    /* are selected.                   */
      mode_reg      |= (MODE_TMD | MODE_RAC | MODE_ADM);
      outp (pt->pt_r_timr, tim_mode);
      break;

   case PT_MD_NON_AUTO:             /* HDLC NON AUTO MODE              */
                                    /* full address recognition,       */
                                    /* receiver active, 2 byte address */
                                    /* fields                          */
      mode_reg      |= (MODE_MDS0 | MODE_RAC | MODE_ADM);

      if (((pt->pt_op_mode == PT_MD_AUTO) &&
          (pt->pt_state & PT_TX_ACTIVE) && (pt->pt_tx_frame == PT_FR_I))
          || (inp(pt->pt_r_star2) & (STAR2_TREC | STAR2_WFA)))
      {
         MISSING_ACKNOWLEDGE (pei);
         ResetHDLC_ICC (pei);
      }
      outp (pt->pt_r_timr, 0);
      break;

   case PT_MD_TRANSP:               /* TRANSPARENT MODE                */
                                    /* SAPI-address (high-byte)        */
                                    /* recognition                     */
      mode_reg       |= (MODE_MDS1 | MODE_MDS0 | MODE_RAC | MODE_ADM);
      break;

   case PT_MD_EXT_TRANSP:           /* EXTENDED TRANSPARENT MODE       */
   case PT_MD_CLEAR_EXT:            /* as well as clear mode           */
                                    /* no address recognition          */
      mode_reg      |= (MODE_MDS1 | MODE_MDS0 | MODE_RAC);
      break;

   default:
      outp (pt->pt_r_mask, 0x00);
      return (ACK_WRONG_PARM);
}

pt->pt_op_mode = mode;              /* save MODE register settings     */

                                    /* modulo: 1 (mod 128); 0 (mod 8)  */
outp (pt->pt_r_sap2, (BYTE) (modulo ? 0x02 : 0x00));
outp (pt->pt_r_tei2, 0xFF);

if (modulo)
   pt->pt_state |= PT_M128;
else
   pt->pt_state &= ~PT_M128;

outp (pt->pt_r_mode, mode_reg);
```

```
   outp (pt->pt_r_mask, 0x00);

   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: StoreTEI_ICC ()                                              */
/*    Parms   : 'pei', 'tei' and 'reg2'                                      */
/*    purpose : program TEI in register TEI1 (reg2 = 0) or TEI2 (reg2 = 0)   */
/*                                                                           */
/*****************************************************************************/
EXPORT int
StoreTEI_ICC (pei, tei, reg2)
   BYTE           pei, tei, reg2;
{
   register PEITAB         *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (reg2 == 1)                       /* store TEI in register TEI2       */
      outp (pt->pt_r_tei2, tei);
   else
   {                                    /* store TEI in register TEI1       */
      outp (pt->pt_r_xad2, tei);
      outp (pt->pt_r_tei1, tei);
   }
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: StoreSAPI_ICC ()                                             */
/*    Parms   : pei, sapi, reg2                                              */
/*    purpose : store SAPI in register SAPI1 (reg2 = 0) or SAPI2             */
/*              (reg2 = 0)                                                   */
/*                                                                           */
/*****************************************************************************/
EXPORT int
StoreSAPI_ICC (pei, sapi, reg2)
   BYTE           pei, sapi, reg2;
{
   register PEITAB         *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   sapi &= ~0x03;

   if (reg2 == 1)                       /* store SAPI in SAP2               */
      outp (pt->pt_r_sap2, sapi | ((pt->pt_state & PT_M128) ? 0x02 : 0x00));
   else
   {                                    /* store SAPI in SAP1               */
      outp (pt->pt_r_xad1, sapi);

      if ((pt->pt_ModulMode == PT_MM_NT) || (pt->pt_ModulMode == PT_MM_LT_S))
         sapi |= 0x02;
```

```
      outp (pt->pt_r_sap1, sapi);
   }
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: RecReady_ICC ()                                              */
/*    Parms   : pei, ready                                                   */
/*    purpose : set HDLC receiver ready      ('ready'= 1)                    */
/*                                not ready  ('ready'= 0)                    */
/*              To be used in auto mode only                                 */
/*                                                                           */
/*****************************************************************************/
EXPORT int
RecReady_ICC (pei, ready)
   BYTE     pei, ready;
{
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   outp (pt->pt_r_cmdr, (BYTE) (ready ? 0x00 : CMDR_RNR));
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: ResetHDLC_ICC ()                                             */
/*    Parms   : pei                                                          */
/*    purpose : reset HDLC controller                                        */
/*                                                                           */
/*****************************************************************************/
EXPORT int
ResetHDLC_ICC (pei)
   BYTE     pei;
{
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   outp (pt->pt_r_mask, 0xFF);

                                    /* clear receive and transmit       */
                                    /* paths, i.e. clear the status     */
                                    /* variables indicating any         */
                                    /* transmission or reception of     */
                                    /* frames and release the MMU       */
                                    /* buffers                          */
   FREE_TX_PATH (pt->pt_pei);

   if (pt->pt_rx_start)
   {
      MMU_free (pt->pt_rx_start);
      pt->pt_rx_start    = NULL_PTR;
```

```
      pt->pt_state      &= ~PT_REC_ACTIVE;
      pt->pt_rx_frame    = 0x00;
      pt->pt_rx_cnt      = 0;
   }

   pt->pt_state    &= ~PT_REC_ACTIVE;

                                       /* set the reset flag in the state  */
                                       /* variable. This allows the        */
                                       /* interrupt service routine to     */
                                       /* react correctly on the following */
                                       /* XPR interrupt                    */
   pt->pt_state |= PT_HDLC_RESET;
                                       /* the reset commands:              */
                                       /* - receive message complete (RME) */
                                       /* - reset hdlc receiver      (RHR) */
                                       /* - transmitter reset        (XRES)*/
   outp (pt->pt_r_cmdr, CMDR_RMC | CMDR_RHR | CMDR_XRES);

   if (pt->pt_op_mode == PT_MD_AUTO)   /* write TIMR register to stop the  */
                                       /* internal timer in automode       */
      outp (pt->pt_r_timr, inp(pt->pt_r_timr));

   outp (pt->pt_r_mask, 0);            /* now allow all interrupts again   */
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function:   SendFrame_ICC ()                                           */
/*    Parms   :   'pei'                                                      */
/*                'frame_type' specifying the frame                          */
/*                'cnt'        number of bytes to send                       */
/*                'frame_ptr'  pointer to the data bytes                     */
/*                                                                           */
/*    purpose :   Initiate transmission of HDLC frames ( S, U, I, UI )       */
/*                                                                           */
/*****************************************************************************/
EXPORT int
SendFrame_ICC (pei, frame_type, cnt, frame_ptr)
   BYTE        pei, frame_type;
   WORD        cnt;
   FPTR        frame_ptr;
{
   register PEITAB   *pt;
   BYTE              cmd;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

                                       /* return if XFIFO is not write     */
                                       /* enable                           */
   if (!(inp (pt->pt_r_star) & 0x40))
      return (ACK_ACCESS_FAULT);

                                       /* return if transmit path still    */
                                       /* blocked and not in automode      */
```

```
if (pt->pt_state & PT_TX_ACTIVE && pt->pt_op_mode != PT_MD_AUTO)
   return (ACK_ACCESS_FAULT);

if (pt->pt_op_mode == PT_MD_AUTO)
{
                                    /* it is not allowed to send an I   */
                                    /* frame in the timer recovery      */
                                    /* or in waiting_for_acknowledge    */
                                    /* status                           */
   if (inp(pt->pt_r_star2) & (STAR2_TREC | STAR2_WFA))
      if (frame_type == PT_FR_I)
         return (ACK_ACCESS_FAULT);

   if (inp(pt->pt_r_star2) & STAR2_WFA)
      if (pt->pt_state & PT_TX_MMU_FREE)
         {
            MMU_free (pt->pt_tx_start);
            pt->pt_state &= ~PT_TX_MMU_FREE;
         }
}

pt->pt_state   |= PT_TX_ACTIVE;     /* transmitter is active            */
pt->pt_tx_start = frame_ptr;        /* store data frame pointer         */
pt->pt_tx_frame = frame_type;       /* and frame type                   */

if (cnt <= 32)
{
                                    /* if the number of bytes is <=32   */
                                    /* the frame can be shifted         */
                                    /* completely into the XFIFO        */
   STRING_OUT (frame_ptr, pt->pt_r_fifo, cnt);
   pt->pt_tx_cnt = 0;
}
else
{
                                    /* if the number of bytes is        */
                                    /* greater 32 the first 32 are      */
                                    /* shifted into the XFIFO, the      */
                                    /* remaining are sent later         */
                                    /* (interrupt service routine)      */
   STRING_OUT (frame_ptr, pt->pt_r_fifo, 32);
   pt->pt_tx_cnt  = cnt - 32;
   pt->pt_tx_curr = frame_ptr + 32;
}

                                    /* compute the command byte for     */
                                    /* the CMDR register:               */
                                    /* in automode the 'transmit I      */
                                    /* frame' command must be used      */
                                    /* when it is an HDLC I frame.      */
                                    /* The 'transmit transparent        */
                                    /* frame' command must be used in   */
                                    /* all other cases                  */
if (pt->pt_op_mode == PT_MD_AUTO)
{
   cmd = (pt->pt_tx_frame == PT_FR_I) ? CMDR_XIF : CMDR_XTF;
```

```
      if (inp (pt->pt_r_star) & CMDR_RNR)
         cmd |= CMDR_RNR;
   }
   else
      cmd = CMDR_XTF;
                                       /* When the frame fits completely   */
                                       /* into the XFIFO the XME command   */
                                       /* must be given                    */
   if (!pt->pt_tx_cnt)
      cmd |= CMDR_XME;

   outp (pt->pt_r_cmdr, cmd);          /* now output the command byte to   */
                                       /* the CMDR register                */

                                       /* UI frame sent while waiting for  */
                                       /* ackowledge in automode (an ID    */
                                       /* check response UI frame)         */
                                       /* The flag is checked by the       */
                                       /* interrupt service routine when   */
                                       /* handling the next XPR interrupt. */
   if (inp(pt->pt_r_star2) & STAR2_WFA && pt->pt_op_mode == PT_MD_AUTO
                                    && frame_type == PT_FR_UI)
      pt->pt_state |= UI_SENT_WHILE_WAITING_FOR_ACK;

   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: Loop_ICC ()                                                  */
/*    Parms   : 'pei'                                                        */
/*              'on'    1  -> test-loop on                                   */
/*                      0  -> test-loop off                                  */
/*    purpose: switch testloop at the IOM interface on/off                   */
/*                                                                           */
/*****************************************************************************/
EXPORT int
Loop_ICC (pei, on)
   BYTE        pei;
   BOOLEAN     on;
{
   PEITAB   *pt_dch;
   BYTE     r_spcr;
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   pt_dch = GetPeitab_BASIC (0);

   if (on)                             /* Loop ON                          */
   {
      pt->pt_state |= PT_LOOP;

                                       /* enable clocks in TE mode         */
      if (pt->pt_ModulMode == PT_MM_TE)
```

```
      {
                                        /* dummy value in the cixr register */
                                       /* prevents a false interpretation of*/
                                       /* the incoming (looped) C/I channel */
         if (EnableClk_BASIC (pt_dch))
            outp (pt_dch->pt_r_cixr, 0x6F);
      }

      r_spcr = inp (pt->pt_r_spcr);
      outp (pt->pt_r_spcr, r_spcr | SPCR_TPL);
   }
   else                                 /* Loop OFF                         */
   {
      r_spcr = inp (pt->pt_r_spcr) & ~SPCR_TPL;
      outp (pt->pt_r_spcr, r_spcr);

      pt->pt_state &= ~PT_LOOP;
   }

   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: SwitchB_ICC ()                                               */
/*    purpose : switch the B-channels in IOM1 configurations                 */
/*              to the SSI or SLD interface or back to network               */
/*                                                                           */
/*****************************************************************************/
EXPORT int
SwitchB_ICC (pei, chan_ctrl, sip_act)
   BYTE        pei, chan_ctrl;
   BOOLEAN     sip_act;
{
   register PEITAB   *pt;
   BYTE              r_spcr;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (chan_ctrl > 0x0F)
      return (ACK_WRONG_PARM);

   if (!(pt->pt_state & PT_IOM2))
   {
      r_spcr = inp (pt->pt_r_spcr) & 0xF0;

      if (sip_act)                      /* activate SIP ?                   */
         r_spcr |= SPCR_SAC;            /*  yes: set SAC bit                */
      else
         r_spcr &= ~SPCR_SAC;           /*  no: clear SAC bit               */

      outp (pt->pt_r_spcr, r_spcr | chan_ctrl);
   }

   return (ACK_DONE);
}
```

```
   /* ***   The interrupt service routines   ***                          */
   /*************************************************************************/


/*************************************************************************/
/*                                                                       */
/*    Function: Int_ICC ()                                               */
/*    Parms   :'pt'   pointer to the corresponding PEITAB-table element  */
/*    purpose : handle ICC (ISAC-S, ISAC-P) interrupts                   */
/*              Int_ICC is called from IntServ_BASIC in basic_l2.c which */
/*              is SIPB system specific.                                 */
/*                                                                       */
/*************************************************************************/
EXPORT void
Int_ICC (pt)
   register PEITAB   *pt;
{
   WORD           cnt;
   BYTE           exir, cmd;
   register BYTE  ista;

   if (!(ista = inp (pt->pt_r_ista)))
      return;

   exir = inp (pt->pt_r_exir);

                                       /* XPR interrupt                 */
                                       /* =============                 */
                                       /* the XPR interrupt indicates   */
                                       /* that the XFIFO is ready for new  */
                                       /* data bytes.                   */
                                       /* Reasons:                      */
                                       /* - HDLC controller reset       */
                                       /*                    (CMDR:XRES)    */
                                       /* - data transmission finished  */
   if ((ista & ISTA_XPR) && !(ista & ISTA_TIN) && !(exir & EXIR_PCE))
   {
                                       /* transmit byte count is 0      */
                                       /* ------------------------      */
      if ((cnt = pt->pt_tx_cnt) == 0)
      {
                                       /* HDLC controller reset command    */
                                       /* given previously ?            */
                                       /* ------------------------------   */
                                       /* do nothing when it was a HDLC    */
                                       /* controller reset only the     */
                                       /* indicating flag must be cleared  */
         if (pt->pt_state & PT_HDLC_RESET)
            pt->pt_state &= ~PT_HDLC_RESET;
         else
         {
                                       /* XPR was generated because the    */
                                       /* last transmission is finished    */
                                       /* ------------------------------   */
                                       /* AUTOMODE operation ?          */
```

```
if (pt->pt_op_mode == PT_MD_AUTO)
{
                           /* UI frame sent while waiting for  */
                           /* I frame acknowledge ?            */
   if (pt->pt_state & UI_SENT_WHILE_WAITING_FOR_ACK)
   {
                           /* the UI frame was sent out if the */
                           /* XFIFO is empty (write enable)    */
      if (inp(pt->pt_r_star) & STAR_XFW)
         TX_ACKNOWLEDGE (pt->pt_pei, pt->pt_tx_frame);

      pt->pt_state &= ~UI_SENT_WHILE_WAITING_FOR_ACK;

                           /* if we are in timer recovery      */
                           /* status the TREC status check     */
                           /* procedure is activated. The      */
                           /* transmit acknowledge for the I   */
                           /* frame must not be generated !!!  */
      if (inp (pt->pt_r_star2) & STAR2_TREC)
         ENABLE_TREC_STATUS_CHECK ();
      else
         TX_ACKNOWLEDGE (pt->pt_pei, (BYTE) PT_FR_I);
   }
   else
   {
                           /* if we are in timer recovery      */
                           /* status and the last frame was an */
                           /* I frame the TREC status check    */
                           /* procedure is activated.          */
                           /* If not an transmit acknowledge   */
                           /* is generated                     */
      if (pt->pt_tx_frame == PT_FR_I &&
          (inp (pt->pt_r_star2) & STAR2_TREC))
         ENABLE_TREC_STATUS_CHECK ();
      else
         TX_ACKNOWLEDGE (pt->pt_pei, pt->pt_tx_frame);
   }
}
else
                           /* In all other operating modes     */
                           /* (non automode, transparent mode, */
                           /* ...) the transmit acknowledge    */
                           /* can be generated at once.        */
   TX_ACKNOWLEDGE (pt->pt_pei, pt->pt_tx_frame);

                           /* transmit byte count and status   */
                           /* flag are reset and any           */
                           /* MMU buffer used for temporary    */
                           /* transmit data storage is         */
                           /* released if necessary            */
pt->pt_tx_cnt = 0;
pt->pt_state &= ~PT_TX_ACTIVE;

if (pt->pt_state & PT_TX_MMU_FREE)
{
   MMU_free (pt->pt_tx_start);
```

```
            pt->pt_state &= ~PT_TX_MMU_FREE;
         }
      }
   }
   else
   {                                /* transmit count is not 0          */
                                    /* more data to be sent !           */
                                    /* -------------------------        */
      if (pt->pt_op_mode == PT_MD_AUTO)
         cmd = (pt->pt_tx_frame ? CMDR_XTF : CMDR_XIF) |
                                    (inp(pt->pt_r_star) & CMDR_RNR);
      else
         cmd = CMDR_XTF;

                                    /* less than 32 bytes left ?        */
      if (pt->pt_tx_cnt <= 32)
      {
                                    /* shift all bytes into the XFIFO   */
                                    /* and give XME command             */
         STRING_OUT (pt->pt_tx_curr, pt->pt_r_fifo, cnt);
         pt->pt_tx_cnt  = 0;
         outp (pt->pt_r_cmdr, cmd | CMDR_XME);
      }
      else
      {                             /* more than 32 bytes are left to   */
                                    /* be sent; write 32 into the XFIFO */
         STRING_OUT (pt->pt_tx_curr, pt->pt_r_fifo, 32);
         outp (pt->pt_r_cmdr, cmd); /* give the transmit command,       */
         pt->pt_tx_curr += 32;      /* update current buffer pointer    */
         pt->pt_tx_cnt  -= 32;      /* and counter of remaining bytes   */
      }
   }
}

if (ista & ISTA_TIN)                /* TIN interrupt                    */
{                                   /* =============                    */
                                    /* ResetHDLC_ICC (pt->pt_pei);      */
   DISABLE_TREC_STATUS_CHECK ();
   TIN_ERROR (pt->pt_pei);
}
                                    /* HDLC receiver interrupt ?        */
                                    /* =========================        */
                                    /* (receive pool full or receive    */
                                    /* message end and not PCE and not  */
                                    /* TIN)                             */
if ((ista & (ISTA_RPF | ISTA_RME))
                               && !(exir & EXIR_PCE) && !(ista & ISTA_TIN))
   RX_ICC (ista & ISTA_RPF, pt);


                                    /* status change of the remote      */
                                    /* station's receiver               */
                                    /* (i.e. RR or RNR received).       */
                                    /* The status can be determined by  */
                                    /* reading the RRNR bit of          */
                                    /* register STAR                    */
```

```
if (ista & ISTA_RSC)
{
   if (inp (pt->pt_r_star) & 0x10)
      PEER_REC_BUSY (pt->pt_pei);   /* peer receiver busy              */
   else
      PEER_REC_READY (pt->pt_pei);  /* peer receiver ready             */
}


                                    /* B (2.x) versions of L1 device    */
                                    /* controllers can't prevent CIC bit*/
                                    /* being set even when masked.      */
                                    /* CIC interrupt ? (layer 1 device  */
                                    /*                    status change) */
if ((ista & ISTA_CIC) && !interrupt_act)
   IntLay1_BASIC (pt);

if (ista & ISTA_EXI)                /* Extended interrupt ?             */
{                                   /* ==================               */
                                    /* transmit message repeat int. ?   */
   if ((exir & EXIR_XMR) && !(exir & EXIR_PCE) && !(ista & ISTA_TIN))
   {
      XMR_ERROR (pt->pt_pei);
      FREE_TX_PATH (pt->pt_pei);
   }

   if (exir & EXIR_XDU)             /* transmit data underrun ?         */
   {
      TX_DATA_UNDERRUN (pt->pt_pei);
      FREE_TX_PATH (pt->pt_pei);
   }

   if (exir & EXIR_PCE)             /* protocol error interrupt ?       */
   {
                                    /* ResetHDLC_ICC (pt->pt_pei);      */
      PROTOCOL_ERROR (pt->pt_pei);
   }


   if (exir & EXIR_RFO)             /* receive frame overflow int. ?    */
   {
      MMU_free (pt->pt_rx_start);

      pt->pt_rx_start    = NULL_PTR;
      pt->pt_state      &= ~PT_REC_ACTIVE;
      pt->pt_rx_frame    = 0;
      pt->pt_rx_cnt      = 0;

      REC_FRAME_OVERFLOW (pt->pt_pei);
   }

   if (exir & EXIR_MOR)             /* MON channel interrupt ?          */
      if (interrupt_act)
         IntMon_MOFC ();
      else
         Wr_IntMon_MOFC ();
```

```
   }
}
/*****************************************************************************/
/*                                                                           */
/*    Function: RX_ICC ()                                                    */
/*    Parms   : 'pt' pointer to the assigned PEITAB array element            */
/*              'rpf' = 1 if RPF interrupt                                   */
/*    purpose : handle interrupts generated by the receiver of an            */
/*              ICC (ISAC-S, ISAC-P)                                         */
/*                                                                           */
/*****************************************************************************/
LOCAL void
RX_ICC (rpf, pt)
   BOOLEAN           rpf;
   register PEITAB   *pt;
{
   WORD           RecCnt, ctrl;
   FPTR           ptr;
   BYTE           pei = pt->pt_pei;
   BYTE           rsta, tei, sapi, frame_status = VALID;
   BOOLEAN        Two, AutoM, CR_of_I_valid = TRUE;

                                       /* RPF interrupt:                    */
                                       /* 32 bytes of a frame longer than   */
                                       /* 32 bytes have been received       */
                                       /* and are now available in the      */
                                       /* RFIFO.                            */
                                       /* The message is not complete.      */
   if (rpf)
      RecCnt = 32;
   else
   {
                                       /* RME interrupt:                    */
                                       /* Receive message end. The RFIFO    */
                                       /* contains a complete frame         */
                                       /* (length <= 32 byte) or the last   */
                                       /* bytes or a frame (length > 32)    */
                                       /* ================================= */

                                       /* read byte count register(s) to    */
                                       /* get the number of currently       */
                                       /* received bytes                    */
                                       /* please note that ICC / ISAC-S     */
                                       /* version Axx had only one byte     */
                                       /* count register !!!                */
      if (pt->pt_device == PT_ICC || pt->pt_device == PT_ISAC_S)
         RecCnt = (BYTE) inp (pt->pt_r_rfbc);
      else
         RecCnt = (WORD) inp (pt->pt_r_rbcl) |
                  (WORD) (inp (pt->pt_r_rbch) & 0x0F) << 8;

      if (RecCnt && !(RecCnt &= 0x1F))
         RecCnt = 32;
   }
                                       /* 'RecCnt' now contains the number  */
                                       /* of bytes actually received        */
```

```
                                    /* was receiver active before or is */
                                    /* the RPF/RME for a new incoming   */
                                    /* frame ?                          */
if (!(pt->pt_state & PT_REC_ACTIVE))
{
   if (RecCnt > 0)
   {
      if (rpf)
         pt->pt_rx_curr = pt->pt_rx_start = MMU_req (266);
      else
         pt->pt_rx_curr = pt->pt_rx_start = MMU_req (38);

      if (pt->pt_rx_start == NULL_PTR)
      {
         MMU_ERROR (pei);
         pt->pt_rx_frame = PT_FR_NO_MEMORY;
      }
   }

   pt->pt_state |= PT_REC_ACTIVE;
   pt->pt_rx_cnt = RecCnt;
}
else
                                    /* if data has been already         */
                                    /* received only the receive byte   */
                                    /* counter must be updated          */
   pt->pt_rx_cnt += RecCnt;


                                    /* automode and frame greater       */
                                    /* 260 byte and automode link ?     */
if (pt->pt_op_mode == PT_MD_AUTO && pt->pt_rx_cnt > 260 &&
   ((inp (pt->pt_r_rsta) & 0x0D) == 9))
{
   pt->pt_rx_frame = PT_FR_OVERFLOW;

                                    /* ICC B4, ISAC-S B3                */
                                    /* reset the receiver if incoming   */
                                    /* frame exceeds 528 byte I field   */
                                    /* length ->                        */
                                    /* unbounded frame                  */
   if (rpf && pt->pt_rx_cnt > 528)
   {
      outp (pt->pt_r_cmdr, CMDR_RHR);

      MMU_free (pt->pt_rx_start);

      pt->pt_rx_start    = NULL_PTR;
      pt->pt_state      &= ~PT_REC_ACTIVE;
      pt->pt_rx_frame    = 0x00;
      pt->pt_rx_cnt      = 0;

      N201_ERROR (pei);
      return;
   }
```

```
}
else
   if (pt->pt_rx_cnt > 266)
      pt->pt_rx_frame = PT_FR_OVERFLOW;

                                    /* read the bytes from the RFIFO    */
                                    /* if no error was detected         */
if (pt->pt_rx_frame < PT_FR_ERROR)
{
   if (RecCnt)
   {
      STRING_IN (pt->pt_rx_curr, pt->pt_r_fifo, RecCnt);
      pt->pt_rx_curr += RecCnt;     /* update buffer pointer            */
   }                                /* it points to the next free       */
                                    /* location in the buffer           */
}

if (rpf)                            /* return when it was a RPF int.    */
{
   outp (pt->pt_r_cmdr, CMDR_RMC | (inp (pt->pt_r_star) & CMDR_RNR));
   return;
}

                                    /* RME interrupt handling !!!       */
                                    /* =========================        */

                                    /* the receive status byte is in    */
                                    /* register RSTA                    */
rsta = inp (pt->pt_r_rsta);


/*********************************************************************/
/* It follows a scanning section to get some information about the   */
/* received data:                                                    */
/* - Performed address recognition                                   */
/* - SAPI ('sapi'), TEI ('tei') and control field byte(s) ('ctrl')   */
/*    as well as the type of frame (HDLC U, UI, S or I frame) are    */
/*    determined.                                                    */
/* In addition the length of a frame is checked.                     */
/*********************************************************************/

                                    /* set 'pei' according to performed */
                                    /* address recognition              */
pei      |= ((rsta & 0x0C) >> 1) | (rsta & 0x01);
AutoM    = FALSE;
tei      = 0;
sapi     = rsta & 0x02;             /* get the C/R bit value            */
ptr      = pt->pt_rx_start;

                                    /* now get additional information   */
                                    /* (TEI, SAPI, control field)       */
switch (pt->pt_op_mode)             /* It depends on the selected       */
                                    /* operating mode                   */
{
   case PT_MD_CLEAR_EXT:            /* no address recognition,          */
                                    /* no firmware interaction          */
```

```
      pt->pt_rx_frame = PT_FR_TR;
      ctrl  = 0x00L;
      break;

   case PT_MD_EXT_TRANSP:          /* no address recognition, SAPI     */
                                   /* and TEI are the first two bytes  */
                                   /* of data                          */
      if (pt->pt_rx_cnt > 0)
         pt->pt_rx_cnt--;

      sapi = *ptr++;

   case PT_MD_TRANSP:              /* high byte address recognition,   */
                                   /* TEI is the first byte read       */
      if (pt->pt_rx_cnt < 2)
         frame_status = MUTILATED;
      else
         pt->pt_rx_cnt -= 2;
                                   /* read TEI and control field       */
      tei   = *ptr++;
      ctrl  = (WORD) *ptr++;

      if (pt->pt_op_mode == PT_MD_TRANSP)
         pei |= 0x20;
      else
         pei |= 0x30;

      break;

   case PT_MD_AUTO:                /* full address recognition in      */
   case PT_MD_NON_AUTO:            /* AUTO/nonAUTOMODE read only the   */
                                   /* HDLC control field information   */


      if (pt->pt_op_mode == PT_MD_AUTO)
                                   /* AUTOMODE link ???                */
         AutoM = ((rsta & 0x0D) == 0x09) ? TRUE : FALSE;

         if (!AutoM)
            pei |= 0x10;
                                   /* the (first byte of the) control  */
                                   /* field is in register RHCR        */
      ctrl = (WORD) inp (pt->pt_r_rhcr);
      break;
}

switch (ctrl & 0x03)               /* determine the frame type         */
{                                  /* =======================          */
   case 0x3:                       /* *** HDLC U frame **              */

      Two = FALSE;                 /* one byte control field !         */

      if (pt->pt_rx_cnt == 0)
      {
         pt->pt_rx_frame = PT_FR_U;
         break;
```

```
   }
   else                          /* as can be seen here U frames     */
      pt->pt_rx_frame = PT_FR_UI;/* with I field are always treated  */
                                 /* as UI frames regardless whether  */
                                 /* it's an real UI frame or an      */
                                 /* erroneous (= too long) U frame   */
   break;

case 0x1:                        /* *** HDLC S-Frame **              */

                                 /* two byte control field ?         */
   if ((Two = (pt->pt_state & PT_M128)))
   {
      ctrl <<= 8;
      ctrl  |= (WORD) *ptr++;

      if (pt->pt_rx_cnt > 0)
         pt->pt_rx_cnt--;
      else                       /* Second byte of the two byte      */
                                 /* control field is missing !       */
         frame_status = MUTILATED;
   }

   if (pt->pt_rx_cnt > 0)        /* S frame with I-field !           */
      frame_status = TOO_LONG;

   pt->pt_rx_frame = PT_FR_S;
   break;

case 0x2:                        /* *** HDLC I frame **              */
case 0x0:
                                 /* no address recognition           */
   if (pt->pt_op_mode == PT_MD_CLEAR_EXT)
      {
      pt->pt_rx_frame = PT_FR_TR;
      break;
      }

   Two = (pt->pt_state & PT_M128);
   pt->pt_rx_frame = PT_FR_I;

                                 /* C/R bit of received I frame      */
                                 /* valid (=1) in TE configuration ? */
                                 /* If 'CR_of_I_valid' is FALSE the  */
                                 /* automatic acknowledge of an      */
                                 /* I frame in Automode is           */
                                 /* prevented! A protocol software   */
                                 /* will receive the PROTOCOL_ERROR  */
                                 /* message and re-establish the     */
                                 /* link.                            */
   if (AutoM && !(sapi & 0x02) && (pt->pt_ModulMode == PT_MM_TE))
      CR_of_I_valid = FALSE;

   if (AutoM)
      break;
```

```
      if (Two)                      /* two byte control field ?         */
      {
         if (pt->pt_rx_cnt == 0)
            frame_status = MUTILATED;

         if (pt->pt_rx_cnt > 0)
            pt->pt_rx_cnt--;

         ctrl <<= 8;
         ctrl  |= (WORD) *ptr++;
      }
      break;
}


if (pt->pt_rx_cnt > 260)            /* I part greater than 260 ?        */
{
   pt->pt_rx_frame = PT_FR_OVERFLOW;
   N201_ERROR(pei);
                                    /* must reset the controller        */
   outp (pt->pt_r_cmdr, CMDR_RMC | CMDR_RHR | CMDR_XRES);
   outp (pt->pt_r_timr, inp(pt->pt_r_timr));
   pt->pt_state |= PT_HDLC_RESET;
   FREE_TX_PATH (pt->pt_pei);
}
else
   if (!CR_of_I_valid)              /* C/R of I frame invalid in TE ?   */
   {                                /* prevent acknowledging S-frame    */
                                    /* beeing sent and create           */
                                    /* PROTOCOL_ERROR message.          */
      pt->pt_rx_frame = PT_FR_FAULT;
      PROTOCOL_ERROR (pt->pt_pei);
                                    /* must reset the controller        */
      outp (pt->pt_r_cmdr, CMDR_RMC | CMDR_RHR | CMDR_XRES);
      outp (pt->pt_r_timr, inp(pt->pt_r_timr));
      pt->pt_state |= PT_HDLC_RESET;
      FREE_TX_PATH (pt->pt_pei);
   }
   else                             /* enter 'RMC' command if not       */
      outp (pt->pt_r_cmdr, CMDR_RMC | (inp (pt->pt_r_star) & CMDR_RNR));


/*********************************************************************/
/*                                                                   */
/* Now all information about the received frame is available:        */
/*    - performed address recognition or TEI and SAPI values.        */
/*    - HDLC control field                                           */
/*    - type of frame (HDLC U, UI, S, I frame).                      */
/*    - info about the validity of the frame                         */
/*                                                                   */
/*********************************************************************/

if (rsta = (rsta & (RSTA_RDO | RSTA_CRC | RSTA_RAB)) ^RSTA_CRC)
   pt->pt_rx_frame = PT_FR_FAULT;

switch (pt->pt_rx_frame)
```

```
{
   case PT_FR_FAULT:
      if (rsta & RSTA_RDO)
         REC_DATA_OVERFLOW (pei);

      if (rsta & RSTA_RAB)
         REC_ABORTED (pei);

      if (rsta & RSTA_CRC)          /* CRC has already been inverted    */
         CRC_ERROR (pei);

      break;

   case PT_FR_S:                    /* HDLC S frame ?                   */
                                    /* ==============                  */
                                    /* extra parameter for 1 byte       */
                                    /* address field set to FALSE       */
      Decode_S_Frame_BASIC (pei, sapi, tei, ctrl, frame_status,
                        ((pt->pt_state & PT_M128) ? 0x01 : 0x00), FALSE);
      MMU_free (pt->pt_rx_start);
      break;

   case PT_FR_U:                    /* HDLC U frame ?                   */
                                    /* ==============                  */
                                    /* extra parameter for 1 byte       */
                                    /* address field set to FALSE       */
      Decode_U_Frame_BASIC (pei, sapi, tei, (BYTE) ctrl, FALSE);
      MMU_free (pt->pt_rx_start);
      break;

   case PT_FR_UI:                   /* HDLC UI or I frame ?             */
   case PT_FR_I:                    /* ====================             */
   case PT_FR_TR:                   /* ====================             */

      if (pt->pt_rx_frame < PT_FR_ERROR)
      {
         FRAME_PASS     fp;

         fp.mmu_buff          = pt->pt_rx_start;
         fp.start_of_i_data   = ptr;
         fp.i_data_cnt        = pt->pt_rx_cnt;
         fp.Two_byte_cf       = Two;
         fp.ctrl_field        = ctrl;
         fp.pei               = pei;
         fp.frame             = pt->pt_rx_frame | frame_status;
         fp.sapi              = sapi;
         fp.tei               = tei;

                                    /* transfer the frame to the 'long  */
                                    /* frame queue'                     */
         PassLongFrame_BASIC (&fp);
      }
      break;

}  /* end of 'switch (pt->pt_rx_frame)' ------------------------------ */
```

```
                                        /* release the data buffer if the   */
                                        /* frame reception or the frame     */
                                        /* were erroneous                   */
   if (pt->pt_rx_frame >= PT_FR_ERROR)
      MMU_free (pt->pt_rx_start);

   pt->pt_rx_start    = NULL_PTR;
   pt->pt_state      &= ~PT_REC_ACTIVE;
   pt->pt_rx_frame    = 0x00;
   pt->pt_rx_cnt      = 0;
}
/*****************************************************************************/
/*                                                                           */
/*    Function: Check_TREC_status_ICC ()                                     */
/*    Parms   :                                                              */
/*    purpose : called periodically if timer recovery status was detected    */
/*              during previous XPR interrupt handing. A                     */
/*              transmit-acknowledge for I frame is generated if the TREC    */
/*              status is left.                                              */
/*                                                                           */
/*****************************************************************************/
EXPORT void
Check_TREC_status_ICC ()
{
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (0)))
      return;

   outp (pt->pt_r_mask, ~MASK_TIN);    /* allow only TIN interrupts        */

                                       /* timer recovery status left ?     */
   if (!(inp(pt->pt_r_star2) & STAR2_TREC))
   {
      if (inp(pt->pt_r_ista) & ISTA_TIN)
      {
         ResetHDLC_ICC (pt->pt_pei);
         TIN_ERROR (pt->pt_pei);
      }
      else
                                       /* generate a transmit acknowledge  */
                                       /* I frame if there was no TIN      */
                                       /* interrupt                        */
         TX_ACKNOWLEDGE (pt->pt_pei, (BYTE) PT_FR_I);

      DISABLE_TREC_STATUS_CHECK ();
   }

   outp (pt->pt_r_mask, 0x00);
}
```

```
/*****************************************************************************/
/*                                                                           */
/*    SIEMENS ISDN-Userboard  (c) 1987-1993                                  */
/*    =====================                                                  */
/*                                                                           */
/*    Firmware:   driver functions for SBC / L1 part of ISAC-S               */
/*    File    :   sbc.c                                                      */
/*                                                                           */
/*****************************************************************************/

   /* Include Files                                                          */
   /* =============                                                          */


#include "def.h"
#include "basic.h"
#include "message.h"

                /* CI codes for SBC and ISAC-S                        PM  */
                /*****************************************************/

#define  CI_PU          (BYTE)0x1C     /* 0111  PU indication             */
#define  CI_TIM         (BYTE)0x00     /* 0000  timing requested          */
#define  CI_AI          (BYTE)0x30     /* 1100  activation indication     */
#define  CI_AR          (BYTE)0x20     /* 1000  activation request        */
#define  CI_DIU         (BYTE)0x3C     /* 1111  deactivation ind. upstream */
#define  CI_DID         (BYTE)0x3C     /* 1111  deactivation ind. downst.  */
#define  CI_DR          (BYTE)0x00     /* 0000  deactivation request      */
#define  CI_RS          (BYTE)0x04     /* 0001  Reset                     */
#define  CI_EI          (BYTE)0x18     /* 0110  Error indicate downstream  */


   /* Imported Functions                                                     */
   /* ==================                                                     */

      /* from  crt0.asm                                                     */
IMPORT   WORD           ENTERNOINT ();
IMPORT   void           LEAVENOINT ();

      /* from basic00.c                                                     */
IMPORT   PEITAB         *GetPeitab_BASIC ();


   /* Export Functions                                                       */
   /* ================                                                       */

EXPORT   int            InitL1_SBC ();
EXPORT   int            ActL1_SBC ();
EXPORT   int            ArlL1_SBC ();
EXPORT   int            DeaL1_SBC ();
EXPORT   void           IntL1_SBC ();

EXPORT   int            ResL1_SBC ();
EXPORT   int            EnaClk_SBC ();
```

```
   /* Variables                                                          */
   /* =========                                                          */

   /* Function Declaration                                               */
   /* ====================                                               */

/*************************************************************************/
/*                                                                       */
/*    Function: EnaClk_SBC ()                                            */
/*    Parms   : pointer to PEITAB table element                          */
/*    purpose : enable clocks for TE configurations                      */
/*                                                                       */
/*************************************************************************/
EXPORT int
EnaClk_SBC (pt)
   register PEITAB   *pt;
{
   unsigned int   count, i = 0;
   BYTE           BitSet, spcr;
                                    /* Test to see if clocks are        */
                                    /* actually there. Because the SBC  */
                                    /* after reset does not deactivate  */
                                    /* its clocks immediately we will   */
                                    /* make pretty sure that the clocks */
                                    /* are there before we leave this   */
                                    /* routine                          */
   BitSet = inp (pt->pt_r_star) & STAR_BVS;
   count = 0;
                                    /* we test to see if 6 changes in   */
                                    /* the STAR:BVS bit indicating the  */
                                    /* reception of at least 3 frames   */
                                    /* (6 B channels). If at any time   */
                                    /* we fail to find a bit change     */
                                    /* and the counter i reaches its    */
                                    /* maximum then we assume that      */
                                    /* clocks are no longer present     */
   for (i = 0; i < 500; i++)
      if ((inp(pt->pt_r_star) & STAR_BVS) != BitSet)
         {                          /* Of course we have to reset our   */
                                    /* counter every time a bit change  */
         if (++count > 6)           /* is observed to give the next     */
            return (FALSE);         /* bit change the same amount of    */
                                    /* time in which to occur !!!       */
         i = 0;
         BitSet = inp (pt->pt_r_star) & STAR_BVS;
         }

                                    /* the Bx versions reqire one edge  */
                                    /* at FSC.                          */
                                    /* Otherwise the setting of the SPU */
                                    /* has no effect (result: no clock) */
                                    /* The IOM direction control bit    */
                                    /* IDC in the ADF1 (SQXR) register  */
                                    /* is set before and reset after    */
                                    /* the system is clocking           */
```

```
                                        /* ICC Bx: IDC is in reg. ADF1      */
   if (pt->pt_device == PT_ICC_B)
      outp (pt->pt_r_adfr, 0x10);

                                        /* ISAC-S Bx: IDC is in reg. SQXR   */
   if (pt->pt_device == PT_ISAC_S_B)
      outp (pt->pt_r_sqxr, 0x80);

   spcr = inp(pt->pt_r_spcr);
   outp (pt->pt_r_spcr, spcr | SPCR_SPU);

   if (pt->pt_state & PT_IOM2)
     outp (pt->pt_r_cixr, CIXR_TBC | CI_TIM | 0x03);
   else
      outp (pt->pt_r_cixr, CIXR_TBC | CI_TIM);

                                        /* wait for power up indication     */
   while ((inp(pt->pt_r_cixr) & CIR_MASK) != CI_PU)
      if (++i > 1000)
         break;                         /* time out                         */

   outp (pt->pt_r_spcr, spcr);

                                        /* now reset the IDC bit            */

                                        /* ICC Bx: IDC is in reg. ADF1      */
   if (pt->pt_device == PT_ICC_B)
      outp (pt->pt_r_adfr, 0x00);
                                        /* ISAC-S Bx: IDC is in reg. SQXR   */
   if (pt->pt_device == PT_ISAC_S_B)
      outp (pt->pt_r_sqxr, 0x00);

   return (TRUE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: InitL1_SBC ()                                                */
/*    Parms   : PEI value, mode of operation                                 */
/*    purpose : initialize an SBC controlling ICC / L1 part of an ISAC-S     */
/*              reset L1 to come to default state                            */
/*                                                                           */
/*****************************************************************************/
EXPORT int
InitL1_SBC (pei, mode_type)
   BYTE     pei, mode_type;
{
   register PEITAB   *pt;
   BYTE              r_mode;

                                        /* return if the addressed device   */
                                        /* is not operational or not used   */
                                        /* for LAYER 1  control             */
   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (!(pt->pt_state & PT_L1_CTRL))
```

```
   return (ACK_NOT_SUPPORTED);


outp (pt->pt_r_mask, 0xFF);
                                  /* compare the requested            */
                                  /* initialization mode with         */
                                  /* detected hardware configuration  */
                                  /* (’pt_ModulMode’)                 */
if (pt->pt_ModulMode != mode_type)
{
   outp (pt->pt_r_mask, 0x00);
   return (ACK_WRONG_MODUL_MODE);
}

                                  /* timing mode 0 is used on the     */
                                  /* SIPB for TE and NTS configu-     */
                                  /* ration                           */

r_mode = inp (pt->pt_r_mode);

if (mode_type == PT_MM_TE)
   outp (pt->pt_r_mode, (r_mode & ~(MODE_HMD2 | MODE_HMD1)) | MODE_HMD0);
else
   outp (pt->pt_r_mode, r_mode & ~(MODE_HMD2 | MODE_HMD1 | MODE_HMD0));

if (pt->pt_state & PT_IOM2)       /* IOM 2 mode ?                     */
{
   outp (pt->pt_r_adf2, 0x80);    /* program IOM2 mode in ICC/ISAC-S  */

   switch (mode_type)
   {
      case PT_MM_NT:
                                  /* Changed to be terminal mode      */
                                  /* timing rather than SPCR_SPM      */
         outp (pt->pt_r_spcr, 0x00);
                                  /* no terminal specific functions   */
         outp (pt->pt_r_stcr, 0x00);

         outp (pt->pt_r_mode, (r_mode & ~(MODE_HMD2 | MODE_HMD0))
                                                  | MODE_HMD1);
         break;

      case PT_MM_TE:
         outp (pt->pt_r_spcr, 0x00);   /* terminal mode               */
         outp (pt->pt_r_stcr, 0x70);   /* TIC bus address ’7’         */
                                       /* no watchdog timer           */
         break;
   }
}
else
{
   outp (pt->pt_r_adf2, 0x00);    /* program IOM2 mode in ICC/ISAC-S  */
   outp (pt->pt_r_stcr, 0x70);    /* program TIC bus address          */
}

outp (pt->pt_r_mask, 0x00);
```

```
   if (!ResL1_SBC (pt))
      return (ACK_ACCESS_FAULT);

   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: ActL1_SBC ()                                                 */
/*    Parms   : PEI value                                                    */
/*    purpose : establish L1 link   (= activation)                           */
/*                                                                           */
/*****************************************************************************/
EXPORT int
ActL1_SBC (pei)
   BYTE     pei;
{
   register PEITAB   *pt;
                                       /* return if the addressed device   */
                                       /* is not operational or not used   */
                                       /* for LAYER 1  control             */
   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (!(pt->pt_state & PT_L1_CTRL))
      return (ACK_NOT_SUPPORTED);

                                       /* the activation procedure is not  */
                                       /* done if the layer 1 link is      */
                                       /* already established. In that     */
                                       /* case only an activation          */
                                       /* indication message is generated  */
   if (((pt->pt_CI_rec = inp(pt->pt_r_cixr)) & CIR_MASK) != CI_AI)
   {
      if (pt->pt_ModulMode == PT_MM_TE)
         EnaClk_SBC (pt);

      if (pt->pt_state & PT_IOM2)
         outp (pt->pt_r_cixr, CIXR_TBC | CI_AR | 0x03);
      else
         outp (pt->pt_r_cixr, CIXR_TBC | CI_AR);

      return (ACK_DONE);
   }

   DECODE_L1_STATUS (pei, pt->pt_CI_rec);
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: ArlL1_SBC ()                                                 */
/*    Parms   : PEI value                                                    */
/*    purpose : activate local loop                                          */
/*                                                                           */
/*****************************************************************************/
EXPORT int
```

```
ArlL1_SBC (pei)
   BYTE     pei;
{
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (pt->pt_ModulMode == PT_MM_TE)
      EnaClk_SBC (pt);

   if (pt->pt_state & PT_IOM2)
      outp (pt->pt_r_cixr, 0x6B);
   else
      outp (pt->pt_r_cixr, 0x68);

   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: DeaL1_SBC                                                    */
/*    Parms   : PEI                                                          */
/*    purpose : release L1 link                                              */
/*                                                                           */
/*****************************************************************************/
EXPORT int
DeaL1_SBC (pei)
   BYTE     pei;
{
   register PEITAB   *pt;

   if (!(pt = GetPeitab_BASIC (pei)))
      return (ACK_NOT_SUPPORTED);

   if (!(pt->pt_state & PT_L1_CTRL))
      return (ACK_NOT_SUPPORTED);

   if (pt->pt_ModulMode != PT_MM_NT && pt->pt_ModulMode != PT_MM_LT_S)
      return (ACK_WRONG_MODUL_MODE);

   if (((pt->pt_CI_rec = inp (pt->pt_r_cixr)) & CIR_MASK) != CI_DIU)
   {
      if (pt->pt_state & PT_IOM2)
         outp (pt->pt_r_cixr, CIXR_TBC | CI_DR | 0x03);
      else
         outp (pt->pt_r_cixr, CIXR_TBC | CI_DR);

      return (ACK_DONE);
   }

   DECODE_L1_STATUS (pei, pt->pt_CI_rec);
   return (ACK_DONE);
}
/*****************************************************************************/
/*                                                                           */
/*    Function: IntL1_SBC ()                                                 */
```

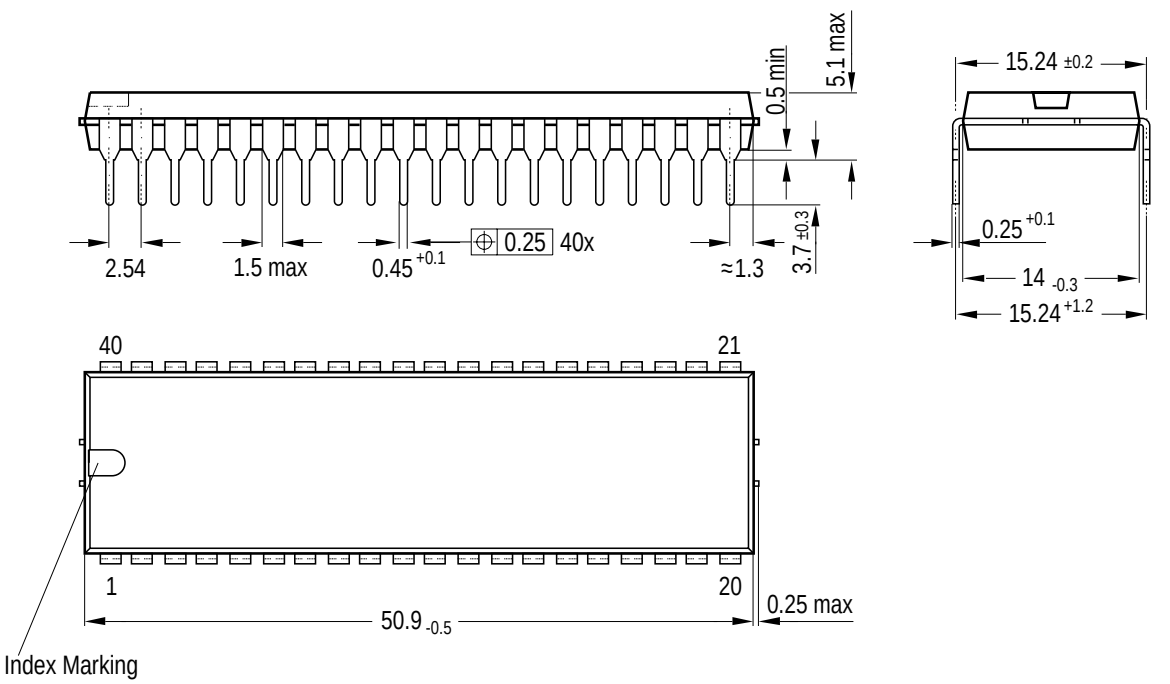
```
/*    Parms   : pointer to PEITAB table element of ICC / ISAC-S           */
/*    purpose : handle C/I interrupts                                     */
/*                                                                        */
/**************************************************************************/
EXPORT void
IntL1_SBC (pt)
   register PEITAB   *pt;
{
   pt->pt_CI_rec = inp (pt->pt_r_cixr);/* read CIRR (CIR0) register       */

   if (pt->pt_ModulMode == PT_MM_NT)
   {
                                       /* in NT / LT-S configuration:     */
                                       /* send DID if SBC/ISAC-S is in the */
                                       /* DIU state                       */
                                       /* -> deactivation                 */
      if ((pt->pt_CI_rec & CIR_MASK) == CI_DIU)
      {
         if (pt->pt_state & PT_IOM2)
            outp (pt->pt_r_cixr, CIXR_TBC | CI_DID | 0x03);
         else
            outp (pt->pt_r_cixr, CIXR_TBC | CI_DID);

      }
   }
   else                                /* TE configuration:               */
   {                                   /* power down SBC/ISAC-S if it has  */
                                       /* changed from activated to       */
                                       /* pending mode                    */
      if ((pt->pt_CI_rec & CIR_MASK) == CI_DR)
      {
         if (pt->pt_state & PT_IOM2)
            outp (pt->pt_r_cixr, CIXR_TBC | CI_DIU | 0x03);
         else
            outp (pt->pt_r_cixr, CIXR_TBC | CI_DIU);

      }
                                       /* activation confirmation in IOM2  */
                                       /* configurations. The SBC         */
                                       /* (ISAC-S) must confirm an        */
                                       /* activation from network side.   */
                                       /* Only then it will be transparent */
                                       /* for upstream B channel data     */
      if ((pt->pt_state & PT_IOM2) &&
         ((pt->pt_CI_rec & CIR_MASK) == CI_AI))
         outp (pt->pt_r_cixr, CIXR_TBC | CI_AR | 0x03);
   }

   DECODE_L1_STATUS (pt->pt_pei, pt->pt_CI_rec);
}
/**************************************************************************/
/*                                                                        */
/*    Function: ResL1_SBC ()                                              */
/*    Parms   : pointer to PEITAB table element of ICC / ISAC-S           */
/*    purpose : Reset SBC / L1 part of ISAC-S                             */
/*              (also used for device test)                               */
```

```
/*                                                                          */
/****************************************************************************/
EXPORT int
ResL1_SBC (pt)
   register PEITAB   *pt;
{
   int      i, state, failed = FALSE;
   BYTE     ForceCommand, NewState, ReleaseCommand, Loop, r_spcr;

   switch (pt->pt_ModulMode)
   {
      case PT_MM_TE:
         ForceCommand = CI_RS;         /* send the RES (reset) code        */
         NewState     = CI_EI;         /* and wait for a change to the EI  */
                                       /* state,                           */
         ReleaseCommand = CI_DIU;      /* then send DIU                    */
         break;

      case PT_MM_NT:
         ForceCommand = CI_DR;         /* send the deactivation request    */
                                       /* code                             */
         NewState     = CI_DIU;        /* and wait for DIU                 */
         ReleaseCommand = CI_DID;      /* then send DID to deactivate the  */
                                       /* SBC                              */
         break;

      default:
         if (pt->pt_Lay1id == SBC_LAY1)
            pt->pt_Lay1id = UNK_LAY1;

         return (FALSE);
   }

   if (pt->pt_state & PT_IOM2)
   {
      ReleaseCommand |= 0x03;
      ForceCommand   |= 0x03;
   }

   state = ENTERNOINT ();              /* disable all system interrupts    */

                                       /* if testloop mode was programmed  */
                                       /* switch it off to enable L1       */
                                       /* status recognition               */
   r_spcr = inp (pt->pt_r_spcr);

   if (Loop = (r_spcr & SPCR_TPL))
      outp (pt->pt_r_spcr, (r_spcr & ~SPCR_TPL));

   outp (pt->pt_r_mask, ~ISTA_CIC);    /* allow only C/I interrupts        */

   if (pt->pt_ModulMode == PT_MM_TE)
      EnaClk_SBC (pt);

                                       /* output the command code          */
   outp (pt->pt_r_cixr, (BYTE) (CIXR_TBC | ForceCommand));
```

```
   i = 0;
                                    /* wait for the expected state      */
   while ((inp(pt->pt_r_cixr) & CIR_MASK) != NewState)
      if (i++ > 20000)
      {                             /* break if timeout                 */
         failed = TRUE;
         break;
      }

                                    /* output the release command       */
   outp (pt->pt_r_cixr, (BYTE)(CIXR_TBC | ReleaseCommand));

   if (pt->pt_ModulMode == PT_MM_TE)   /* TE mode ?                    */
   {                                /* Wait for DIU or AIU because      */
                                    /* it can cause problems for the    */
                                    /* enable clock routine if the      */
                                    /* clocks disappear mid routine     */
                                    /* due to an earlier reset          */

      for (i = 0; i < 20000; i++)
      {
         pt->pt_CI_rec = inp (pt->pt_r_cixr) & CIR_MASK;

         if ((pt->pt_CI_rec == CI_DIU) || (pt->pt_CI_rec == CI_AI))
            break;
      }

      if ((pt->pt_state & PT_IOM2) && (pt->pt_CI_rec == CI_AI))
         outp (pt->pt_r_cixr, CIXR_TBC | CI_AR | 0x03);
   }

   if (Loop)                        /* restore original value of SPCR   */
      outp (pt->pt_r_spcr, r_spcr);

   outp (pt->pt_r_mask, 0x00);      /* enable interrupts again          */
   LEAVENOINT (state);

   if (failed)
   {
      if (pt->pt_Lay1id == SBC_LAY1)
         pt->pt_Lay1id = UNK_LAY1;

      return (FALSE);
   }
   else
      return (TRUE);
}
```

7 Package Outlines

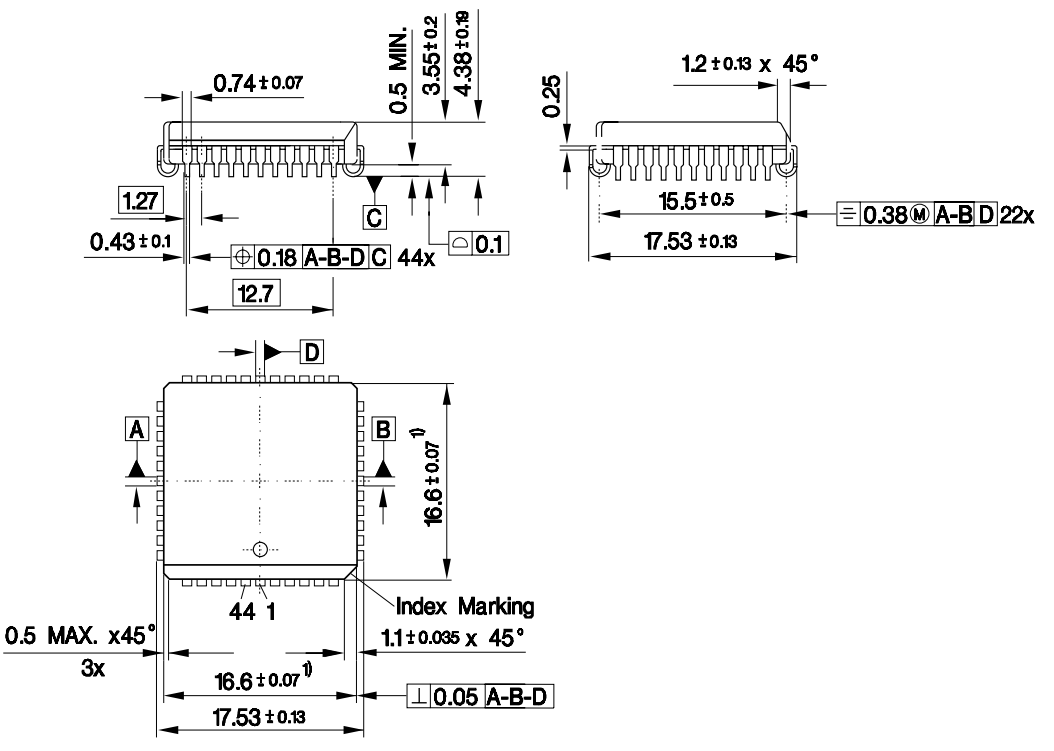
P-DIP-40-2
(Plastic Dual-In-Line Package)



GPD05055

Dimensions in mm

Plastic Package, P-LCC-44-1 (SMD)
(Plastic-Leaded Chip Carrier)

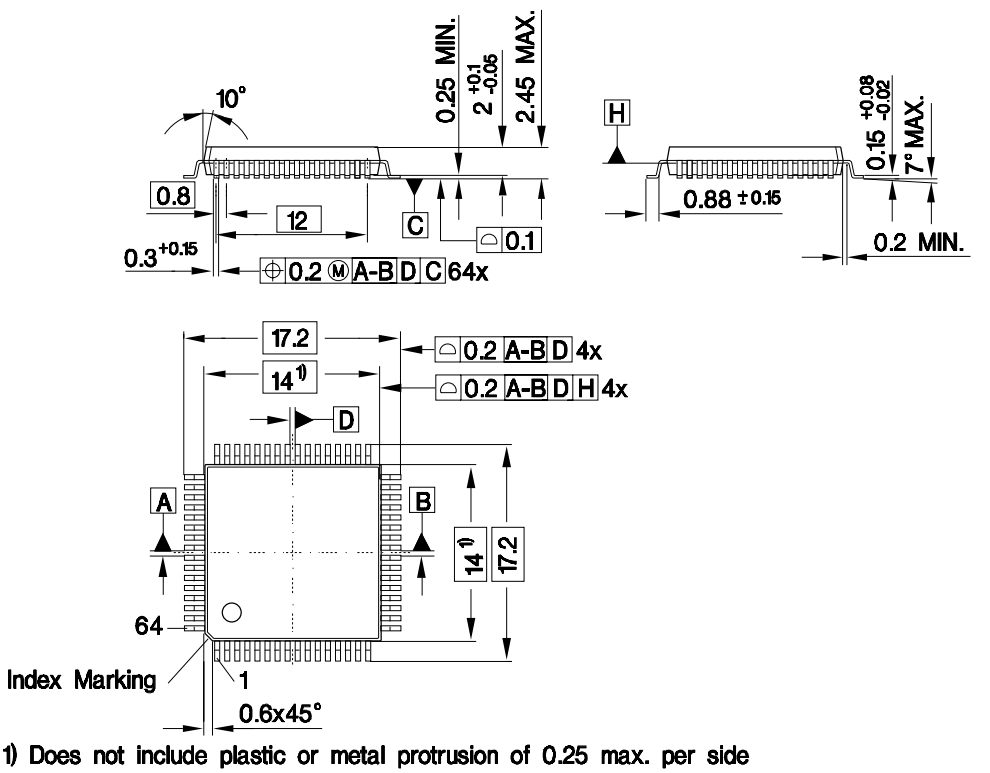


1) Does not include plastic or metal protrusion of 0.15 max. per side

GPL05102

Dimensions in mm

Plastic Package, P-MQFP-64-1 (SMD)
(Plastic Metric Quad Flat Package)



GPM05250

Sorts of Packing
Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”

SMD = Surface Mounted Device

Dimensions in mm

Transformers and Crystals Vendor List

Crystals:

Frischer Electronic
Schleifmühlstraße 2
D-91054 Erlangen, Germany

KVG
Waibstadter Straße 2-4
D-74924 Neckarbischofsheim 2, Germany
Tel.: (…7263) 648-0

NDK
2-21-1 Chome Nishihara Shibuya-Ku
Tokyo 151, Japan
Tel.: (03)-460-2111
or
Cupertino, CA, USA
Tel.: (408) 255-0831

Saronix
4010 Transport at San Antonio
Palo Alto, CA 94303, USA
Tel.: (415) 856-6900
or
via Arthur Behrens KG
Schrammelweg 3
D-82544 Egling-Neufahrn, Germany

Tele Quarz
Landstraße 13
D-74924 Neckarbischofsheim 2, Germany

Transformers:

Advanced Power Components (APC)
47 Riverside
Medway City Estate Strood
County of Kent, GB
Tel.: (044) 634-290 588

Pulse Engineering
P.O. Box 12235
San Diego, CA 92112, USA
Tel.: (619) 268-2454
or
4, avenue du Quebéc
F-91940 Les Ulis, France
or
Dunmore Road
Tuam County Galway, Ireland
Tel.: (093) 24107

S+M Components
Balanstraße 73
P.O. Box 801709
D-81617 Munich, Germany
Tel.: (…89) 4144-8041
Fax.: (…89) 4144-8483

Siemens Oostcamp
Belgium

Schott Corporation
Suite 108
1838 Elm Hill Pike, Nashville, TN 37210, USA
Tel.: (615) 889-8800

TDK
Christinenstraße 25
D-40880 Ratingen 1, Germany
Tel.: (…2192) 487-0

Universal Microelectronics

Vacuumschmelze (VAC)
Grüner Weg 37
Postfach 2253
D-63412 Hanau 1, Germany
Tel.: (…6181) 380
or
186 Wood Avenue South
Iselin, NJ OB830, USA
Tel.: (908) 603 5905

Valor
Steinstraße 68
D-81667 München, Germany
Tel.: (…89) 480 2823
Fax.: (…89) 484 743

Vogt electronic AG
Postfach 1001
D-94128 Obernzell, Germany
Tel.: (…8591) 17-0
Fax.: (…8591) 17-240