



T-Link Building Block (TBB) Specification

--- Issue 2 ---

Document # TL86-0099-02,
October 1986.

(c) Copyright: Northern Telecom, 1986

Datapath Technology
Licensing
Program

T-Link
Building
Block

Northern Telecom makes no representations in respect to, and does not warrant, any of the information in this document, but furnishes such in good faith and to the best of its knowledge and ability. Without restricting the generality of the foregoing, Northern Telecom makes no representations or warranties as to the fitness for a particular purpose, or as to whether or not the use of the information in this document may infringe any patent or other rights of any other person or corporation. The recipient waives any claims it may have against Northern Telecom in respect of any use which the recipient makes of the information or products derived therefrom.

Northern Telecom reserves the right to alter or modify this document, or the equipment or products to which it relates, at any time without notice and without liability.

(c) Copyright : Northern Telecom, 1986.

Table of Contents

Table of Contents	i
List of Figures.....	iii
List of Tables.....	iv
Acronyms.....	v
1. Introduction	1
2. T-Link and CPI.....	3
2.1 T-Link	3
2.2. Computer to PBX Interface (CPI)	4
3. T-Link Building Block.....	5
4. Internal Structure	7
4.1. 8031A Port Use	8
4.2. Address Map.....	9
5. Network Data Channel.....	11
5.1. Operation.....	11
6. Serial Terminal Interface.....	12
7. Command/Status Channel	14
7.1. Hardware Design.....	14
7.1.1. Timing	15
7.2. Register design.....	16
7.3. Operation.....	17
7.3.1. Write to the TBB.....	17
7.3.2. Read from TBB.....	18
7.3.3. Ready Flag	18
8. MISC.....	20
8.1. Hard Reset.....	20
8.2. TBB CLock Circuits.....	20
8.3. Power/Grounding.....	22
9. Data Register Bit Interpretation	23
9.1. Command/Status Register - 0000	23
9.2. Loopback Control - 0001	25
9.2.1. Write (by TAD).....	25
9.2.2. Read (by TAD).....	25
9.3. Config 0 - General - 0100.....	26
9.4. Config 1 - General - Register 0101	27
9.5. Config 2 - General - Register 0110.....	28
9.5.1. Asynchronous Parameters.....	28
9.5.2. Synchronous.....	28
9.6. Config 3 Rate selection/indication - Register 0111.....	29
10. TBB Operation	30
10.1. Sample State Sequence.....	30
11. Applications.....	32
11.1. Datapath TAD.....	34
11.2. ISDN TAD	35

List of Figures

Figure 1-1.	ISDN and Today's Digital Network.....	1
Figure 1-2.	Terminal Adaption Device (TAD).....	2
Figure 3-1.	T-Link Building Block (TBB).....	5
Figure 4-1.	TBB Block Diagram.....	7
Figure 5-1.	Network Data Channel.....	11
Figure 5-2.	Channel Timing.....	11
Figure 6-1.	Serial Terminal Interface.....	12
Figure 7-1.	Command / Status Channel.....	14
Figure 7-2.	Write Cycle Timing.....	15
Figure 7-3.	Read Cycle Timing.....	16
Figure 8-1.	TBB Clock Circuits.....	21
Figure 10-1.	State Diagram.....	30
Figure 11-1.	TAD Blocks.....	32
Figure 11-2.	Protocol Independence.....	33
Figure 11-3.	Network Independence.....	33
Figure 11-4.	Datapath TAD.....	34
Figure 11-5.	ISDN TAD.....	35

List of Tables

Table 4-1. Address Map10
Table 6-1. TBB Terminal Interface Circuits13
Table 7-1. TBB Control Register Codes..... 17
Table 9-1. State Table 23
Table 9-2. Command/Status Register 24
Table 9-3. Config 0 Register26
Table 9-4. Config 1 Register27
Table 9-5. Config 2 Register28
Table 9-6. Rate Table.....29

Acronyms

AUC	- Asynchronous USART Clock
BNR	- Bell Northern Research
CPE	- Customer Premise Equipment
CPI	- Computer to PBX Interface
CTS	- Clear To Send
DCD	- Data Carrier Detect
DCE	- Data Communication Equipment
DSR	- Data Set Ready
DTE	- Data Termination Equipment
DTR	- Data Terminal Ready
DMS100	- Digital Multiplexed Telephone Switch family
EIA	- Electronic Industries Association
EPROM	- Electrically Programmable Read Only Memory
I/F	- Interface
I/O	- Input/Output
IC	- Integrated Circuit
IPE	- Inband Parameter Exchange
INTEL	- INTEL Corporation, Santa Clara California
ISDN	- Integrated Services Digital Network
Kbit/s	- Kilobits per second
Mbit/s	- Megabits per second
NDC	- Network Data Clock
PBX	- Private Branch Exchange
RA	- Rate Adaption
RAM	- Random Access Memory
RLB	- Remote LoopBack
RST	- Reset
RTS	- Request To Send
RXC	- Receive Clock
RXD	- Receive Data
SL100	- Northern Telecoms Stored Logic family of PBXs
STI	- Serial Terminal Interface
SDC	- Synchronous Data Clocks
T1	- Digital Trunk (1.544Mbit/s)
T-Link	- Northern Telecoms T1 RA protocol
TA	- Terminal Adaptor
TAD	- Terminal Adaption Device
TBB	- T-Link Building Block
TI	- Test Indicator
TTL	- Transistor Transistor Logic
TXC	- Transmit Clock
TXD	- Transmit Data
USART	- Universal Synchronous Asynchronous Receiver Transmitter
WDC	- WatchDog Counter
WTC	- Watchdog Timer Clock

1. Introduction

Digital networks that use 64Kbit/s, octet form, digitally encoded transmission facilities are ideal for high speed data transfer. Most DTEs today cannot interface directly with the digital network. To attach present DTEs to the new digital networks Terminal Adaption Devices (TADs) are used. A TAD must be capable of interfacing to the many different types of DTE interfaces available today and adapting the DTE data at non-64Kbit/s to the 64Kbit/s transmission speed.

The Integrated Services Digital Network (ISDN) will use a device called a Terminal Adaptor (TA) to accomodate non-64Kbit/s DTEs at the R Interface as shown in the diagram. The TA must provide rate adaption (RA) as part of its function. The form of RA to be used in the TA is left up to CPE suppliers. Some RA schemes (I.463) have been proposed but these have never been tested in use.

In order to allow ISDN and non-ISDN TADs to communicate during the ISDN transition period it is desirable to use the same RA method for ISDN Terminal Adaptors (TAs) as for today's digital networks.

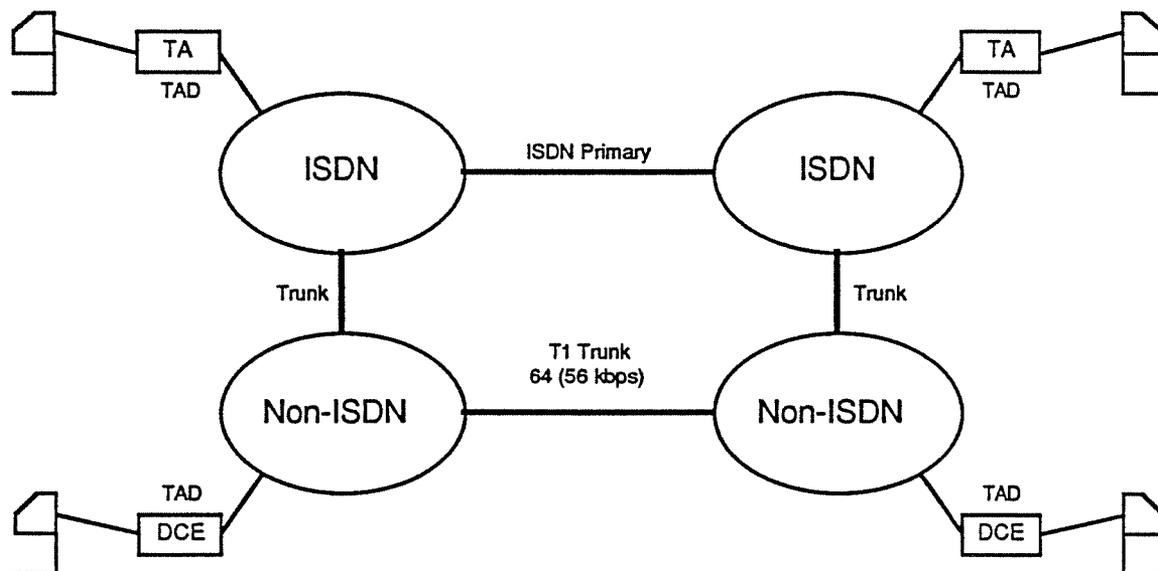


Figure 1-1. ISDN and Today's Digital Networks

T-Link is one method of rate adaption that is in use in today's digital networks. This document is intended to specify the internal design and external interface for a T-Link Rate Adaption Building Block (TBB) in functional and operational terms. Detailed H/W and S/W descriptions will be available at a later date for bona-fide users.

Northern Telecom reserves the right to make changes to the specification. Any changes will result in a new issue of the specification.

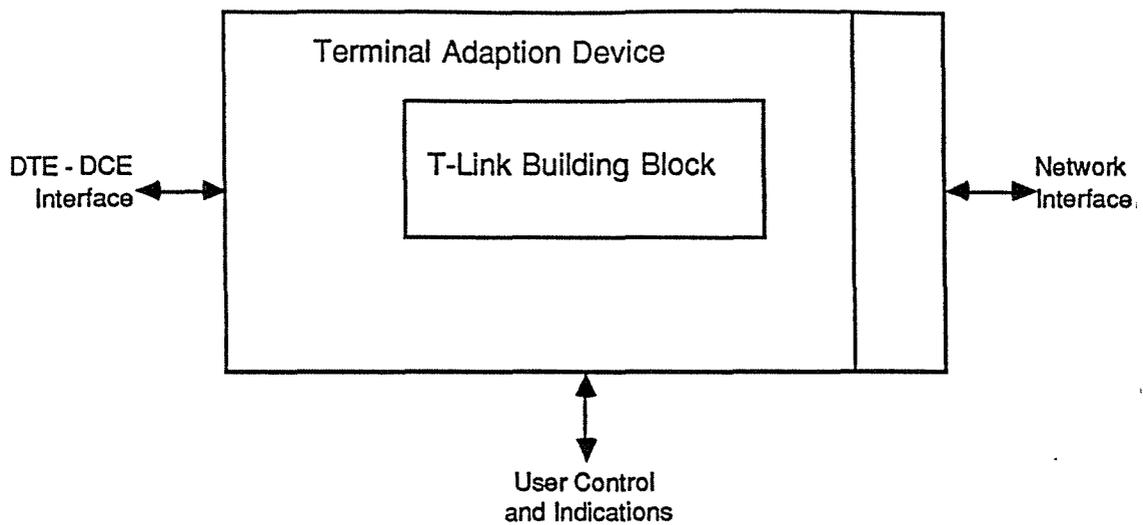


Figure 1-2. Terminal Adaption Device (TAD)

2. T-Link and CPI

Northern Telecom (NT) has developed protocols for passing data across digital networks. The Computer to Private Branch Exchange Interface (CPI) is a trunk type interface for many (24) circuits and the T-Link protocol is a single channel specification.

2.1 T-Link

This protocol allows synchronous or asynchronous terminals from 50bit/s to 64Kbit/s to be adapted to a 64Kbit/s, octet boundaried, digital network. It also accommodates the many parameters associated with asynchronous terminals.

A T-Link communication line is similar to a modem link. Both ends of the link will converse using the T-Link protocol. This allows the ends to:

- 1) Synchronise the end to end link
- 2) Exchange setup parameters to establish communication
- 3) Exchange terminal status to allow the ends to act as DCEs
- 4) Perform the Rate Adaption and Deadaption of data between DTEs

To perform these functions T-Link defines special octets that carry parameters, status and data.

When a link is brought into operation both ends send sync octets and then exchange parameter octets. This is similar to modems that use answer tones and training sequences to establish the end to end synchronization.

When both ends are synchronized they enter the data mode. The protocol handles data differently at different speeds. Up to and including 9.6Kbit/s the data is copied and passed multiple times in separate octets across the link. A voting scheme picks the most commonly repeated octet as the correct data. This means that for these data rates T-Link is capable of correcting transmission errors.

At data speeds above 9.6Kbit/s but below 48Kbit/s the data is not copied and is only sent once. Non-data octets are used to fill in the extra bandwidth.

At 48Kbit/sec and 56Kbit/s the data is not copied and no extra time exists. Rate adaption is done by using fractions of the transmission octet to equate to fractions of the 64Kbit/sec transmission rate.

Example 1) a. If 48Kbit/s rate is to be adapted then only 6 of the 8 data bits of the octet are used. The equivalent data rate is $\frac{6}{8} \times 64 = 48\text{Kbit/s}$

Example 2) For 56Kbit/s only 7 of the 8 bits of the octet are used. The equivalent data rate is $\frac{7}{8} \times 64 = 56\text{Kbit/s}$

For 64Kbit/s data the complete 8 bit transmission octet is used.

To make up the speed difference between low speed DTE data and the 64Kbit/s stream the protocol sends status octets containing the DTE/DCE interface lead information. This also allows the DTEs to control the link for half duplex operation and to enable and disable the link.

Northern Telecom uses T-Link in TADs called Data Units that are used for the Datapath digital data service on Northern Telecoms DMS100/SL100 family of digital telephone switches.

2.2. Computer to PBX Interface (CPI)

CPI was developed in order to allow Digital data to be exchanged between a PBX and a Host Computer using T1 Digital transmission links. These links contain 24 individual octet structured links. CPI specifies a method to use the signalling bits in the T1 link to establish the the end to end layer 1 link through the telephone network and T-Link on each of the 24 channels to establish end to end synchronization as described above. CPI is well established and in use in North America with Northern Telecom DMS100/SL100 Digital Telephone Switches and other major computer manufacturing companies.

3. T-Link Building Block

To assist CPE vendors in quickly establishing a standardized RA scheme, Northern Telecom will offer a series of T-Link Building Blocks (TBB) to CPE vendors for use in their TAs. The TBB is intended to be integrated into a TAD that may be used for an ISDN TA, a Datapath Data Unit, or any other digital TAD application.

One form of building block is pseudo-code software modules that allow the T-Link Protocol to be quickly ported to any micro-processor design.

Another form of building block will consist of a firmware program packaged in an EPROM that contains the complete code necessary to perform the T-link protocol. This code will be in the INTEL 8031A single chip microprocessor assembly language. This firmware package will be a complete implementation of an 8031A based system and the hardware built around it must be specially tailored to support its memory and I/O mapping.

Another form of the building block will be a hardware module that can be integrated into a design as a peripheral IC. This block is intended to be used like a communication controller IC that uses:

- 1) A command and a data register to pass command, status and option information.
- 2) A serial channel to handle the DTE/DCE interface.
- 3) A serial channel to handle the Network interface

In both cases the complete Rate Adaption capability will be built into the TBB. The user will not have the capability of modifying the internal program that handles the Rate Adaption protocol. This is Northern Telecoms version of the T-Link protocol that is used in the DMS Datapath products.

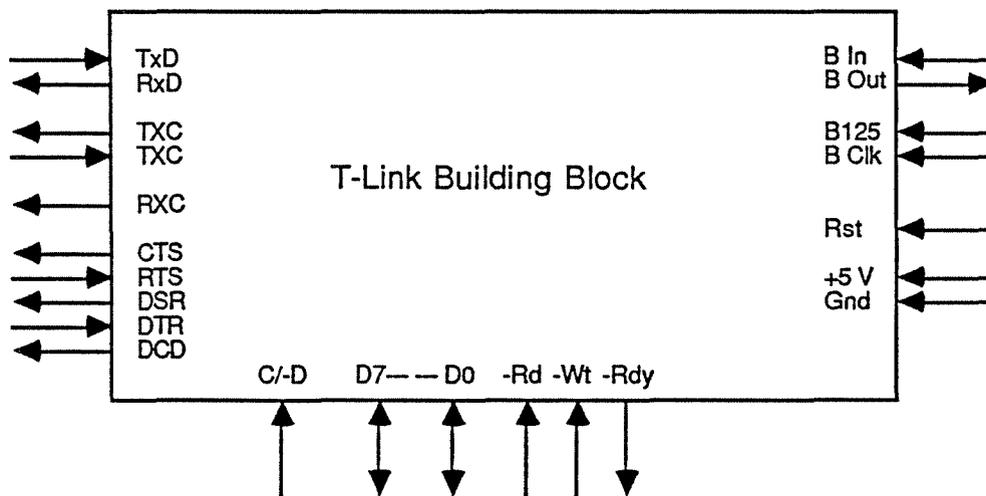


Figure 3-1. T-Link Building Block (TBB)

The document will:

- 1) Specify the general internal structure of an 8031A system required to support the firmware.-- Section 7.
- 2) Specify the structure of the 3 interfaces for the outside world. -- Sections 8,9 and 10.
- 3) Specify any Miscellaneous support circuits required for the 8031A system.-
- Section 11.
- 4) Describe the bit use for the command and option registers.-- section 12.
- 5) Describe the operation of the TBB.-- Section 13.
- 6) Discuss the applications for the TBB.-- Section 14.

4. Internal Structure

The TBB is based on a 12MHz INTEL 8031A single chip microprocessor. The firmware is written to utilize the 8031A in a particular manner. The following block diagram shows the structure that is required to support the firmware.

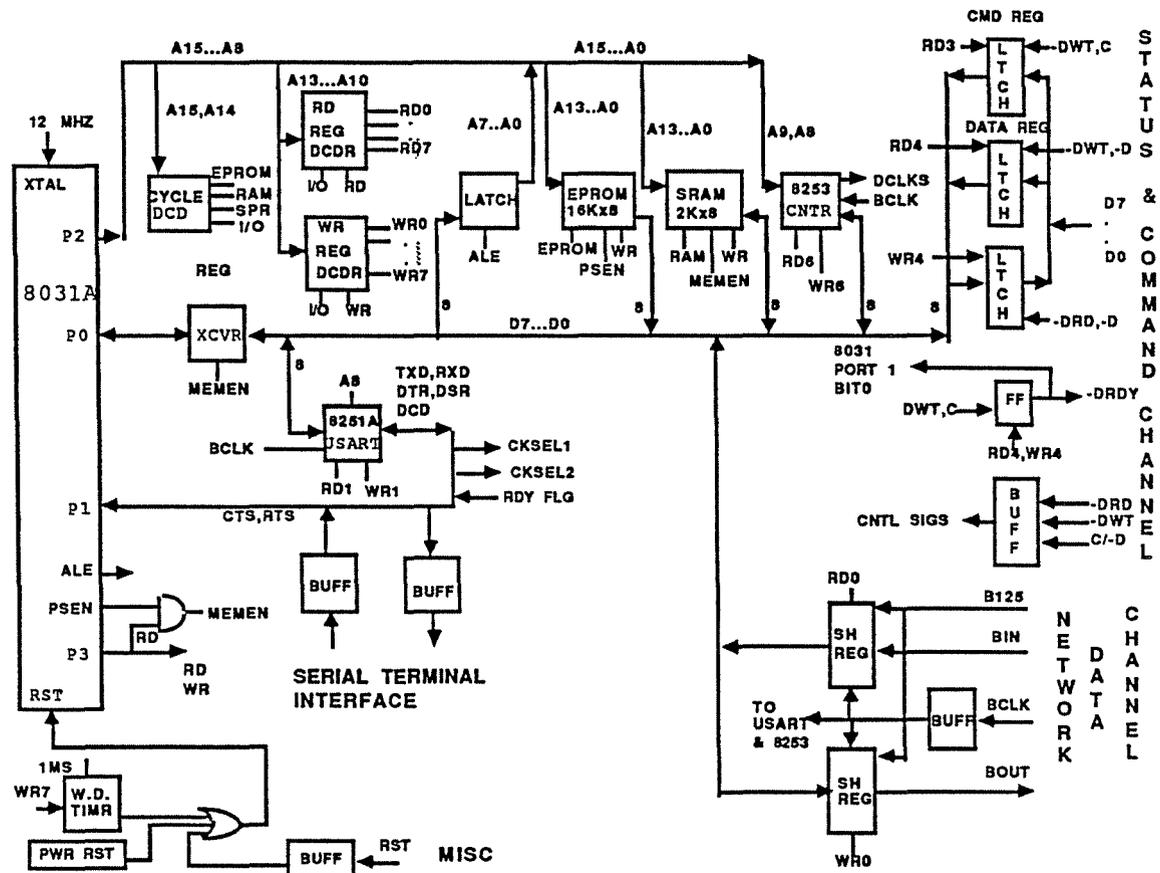


Figure 4-1. TBB Block Diagram

4.1. 8031A Port Use

The 8031A has 4 ports that are used as follows:

- 1) An 8 bit data bus is provided on port 0. Port 0 bits 7..0 = Data bus bits DB7...DB0
- 2) A 16 bit address bus is provided by using port 2 for the upper 8 bits and by multiplexing the lower 8 address bits with the data bits on port 0.

Port 2 bits 7..0 = Add bus bits AD15..AD8

Port 0 bits 7..0 = Add bus bits AD7..AD0

The 8031A is designed to operate in this fashion and the lower 8 bits are latched using the Address Latch Enable (ALE) provided by the 8031A.

- 3) Port 3 is used to provide the control functions to/from the 8031A. These include the read and the write signals along with the Interrupts as follows:

Port 3 bit 0,1 - not used

bit 2 - Txdy Interrupt from USART

bit 3 - Rxdy Interrupt from USART

bit 4 - 125 usec Interrupt from Network Data Channel

bit 5 - Reserved Interrupt from Status/Command Channel

bit 6 - 8031A WR signal

bit 7 - 8031A RD signal

- 4) Port 1 is used as a general I/O register. Some of the DTE/DCE interface leads are handled by this port as the USART does not have enough I/O to handle all the leads that are supported by the TBB.

Port 1 bit 0 - Rdy flag(see section 10)

bit 1..3 - not used

bit 4 - RTS input

bit 5 - CTS output

bit 6 - Clksel1 to select internal or external clk source.

bit 7 - Clksel2 to select network synchronised clocking

See an INTEL data book for the timing specifics for these ports.

4.2. Address Map

As can be seen from table 7-1, bit 15 and 14 are used to decode whether the bus activity will be to memory or general I/O. The lower 14 bits during a memory activity indicate the memory address giving a possible memory space of 16K Bytes.

Program store memory is contained in a 16K x 8bit EPROM (27128) with an access time of 250ns max.

Data store memory is contained in a 2K x 8bit static RAM (9128) with an access time of 200ns max.

The general I/O uses bit 13 as an enable and bits 12,11, and 10 are decoded to create register select lines for up to 8 registers. All non-memory activity is decoded in this manner. Control is provided for:

- 1) Network Data Channel used to pass 64Kbit/s data.-- Register 0.
- 2) USART used to control the DTE/DCE Serial Terminal Interface.-- Register 1.
- 3) Control register to pass control information to the TBB and poll the interface ready flag.-- Register 3.
- 4) Data register used to pass commands and options between the TBB and the TAD.-- Register 4.
- 5) Counter/Timer used for clock manipulation and timers.-- Register 6.
- 6) Watchdog timer used to detect problems in the TBB.-- Register 7.

Some of the registers use additional address bits to further decode the address space within the register.

The USART used is the INTEL 8251 and an INTEL data book should be consulted to explain its operation.

The Counter/Timer used is the INTEL 8253 and the INTEL data book will provide complete information on its use.

The firmware is set up to use the 16 bit address space provided by port 0 and port 2 as follows:

USE	ADDRESS BITS(code)																DESCRIPTION
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
EPROM	0	0	A	A	A	A	AA	AAAAAAAA	AAAA								Memory Addr
RAM	0	1	A	A	A	A	AA	AAAAAAAA	AAAA								Memory Addr
SPARE	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Not Used
I/O																	
(Spare)	1	1	0	x	x	x	-	-	-	-	-	-	-	-	-	-	Spare
(Net Ch)	1	1	1	0	0	0	-	-	-	-	-	-	-	-	-	-	Network Chnl
(USART)	1	1	1	0	0	1	-	0	-	-	-	-	-	-	-	-	Data Reg
	1	1	1	0	0	1	-	1	-	-	-	-	-	-	-	-	Cntrl Reg
(Spare)	1	1	1	0	1	0	-	-	-	-	-	-	-	-	-	-	Spare
(Ext I/F)	1	1	1	0	1	1	-	-	-	-	-	-	-	-	-	-	Cntrl Reg
	1	1	1	1	0	0	-	-	-	-	-	-	-	-	-	-	Data Reg
(Spr)	1	1	1	1	0	1	-	-	-	-	-	-	-	-	-	-	Spare
(Ctr/Tmr)	1	1	1	1	1	0	0	0	-	-	-	-	-	-	-	-	Reg 0
	1	1	1	1	1	0	0	1	-	-	-	-	-	-	-	-	Reg 1
	1	1	1	1	1	0	1	0	-	-	-	-	-	-	-	-	Reg 2
	1	1	1	1	1	0	1	1	-	-	-	-	-	-	-	-	Reg 3
(Wchdg Tmr)	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	Clr Tmr

Table 4-1. Address Map

The Network Data Channel, the Serial Terminal Interface and the and the Command/Status Channel operation will be described in detail in the next 3 sections.

5. Network Data Channel

This channel is designed to pass serial data between the TBB and the network interface of the TAD. The external interface for the channel consists of a serial input (Bin), a serial output (Bout), a 125us clock line (B125), and a bit clock operating at 2.56MHz (Bclk). Internally this is the Network Data Channel Register (0). This is composed of 2 shift registers that are read and written by the 8031A.

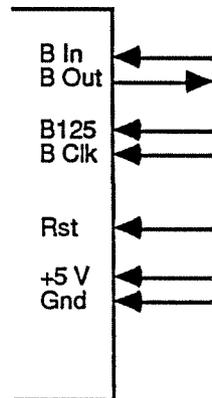


Figure 5-1. Network Data Channel

5.1. Operation

This channel is meant to be attached directly to the line interface circuits of the network interface. Every 125us the line I/F and the TBB pass data between each other. This is triggered by the falling edge of the B125 line. When this line is activated the Bclk is applied to the shift registers and data is passed serially in both directions simultaneously. The B125 line also causes a high priority interrupt to the 8031A. The 8031A will jump to an interrupt routine that reads and writes register 0. In a 64Kbit/s network link the channel transfer must occur every 125 micro-seconds to ensure synchronous operation.

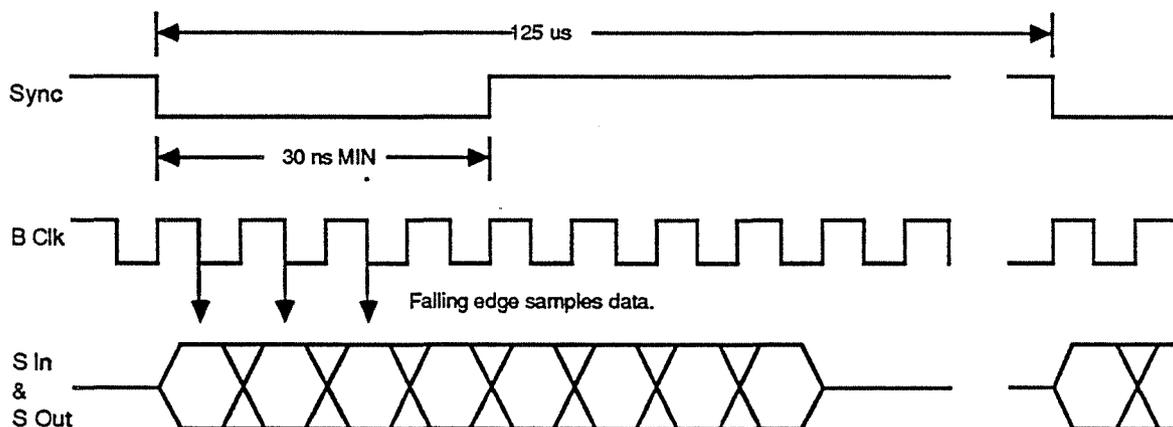


Figure 5-2. Channel Timing

6. Serial Terminal Interface

The Serial Terminal Interface (STI) is setup to provide the I/O necessary to handle the standard V-series/EIA interface. The TBB will contain a USART and clock generation and therefore serial data and clock information must be passed to/from TBB as required for sync or async protocols. The RA protocol must carry terminal lead status across the link, therefore the standard handshake leads are connected directly to the TBB. Note that because the STI is emulating a DCE the signal names are reversed to the normal direction. For example, TXD is an input.

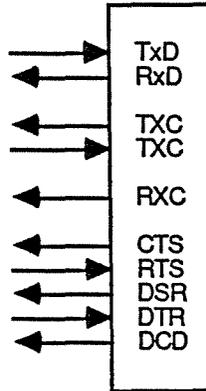


Figure 6-1. Serial Terminal Interface

Clocking for data synchronization can be sourced by the attached device or by the TBB. The following table shows the interface leads that are handled by the TBB. Any other interface leads must be handled by the TAD that controls the TBB.

Signal Name	CCITT / ISO			EIA				Direction
	Ckt #	V.24 Pinout	V.35 Pinout	RS232C		RS449		
				Abbrv	Pinout	Abbrv	Pinout	
Transmit Data	103	2	P, S	BA	2	SD	4, 22	to TBB
Receive Data	104	3	R, T	BB	3	RD	6, 24	from TBB
Request to Send	105	4	C	CA	4	RS	7, 25	to TBB
Ready for Sending	106	5	D	CB	5	CS	9, 27	from TBB
Data Set Ready	107	6	E	CC	6	DM	11, 29	from TBB
Data Terminal Rdy	108/2	20	H	CD	20	TR	12, 30	to TBB
Carrier Detect	109	8	F	CF	8	RR	13, 31	from TBB
Transmit Clock (DCE source)	114	15	Y, AA	DB	15	ST	5, 23	from TBB
Receive Clock (DCE source)	115	17	V, X	DD	17	RT	8, 26	from TBB
Transmit Clock (DTE source)	113	24	U, W	DA	24	TT	17, 35	to TBB

Table 6-1. TBB Terminal Interface Circuits

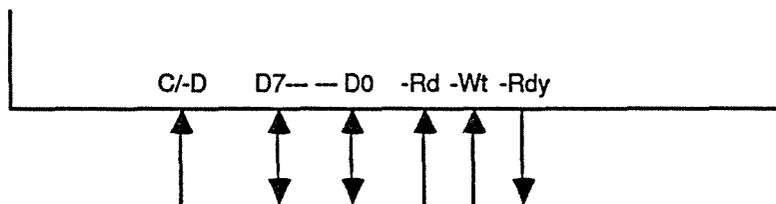
Internally these leads will be buffered and fed either to/from the USART (TXD,DTR,RXD,DSR,DCD), to/from the 8031A port 1 (RTS,CTS), or to/from the CNTR/timer (TXCde, TXC, RXC).

7. Command/Status Channel

This parallel I/F performs all functions required to allow the TAD to pass the following information:

- 1) Options for T-Link protocol
- 2) Commands/status info

The channel is used to allow the TAD to write commands and read status to/from the TBB to set up options such as baud rate and parity.



7-1. Command/Status Channel

7.1. Hardware Design

Refer to the block diagram (fig 7-1). The channel consists of:

- 1) A control register that is used to tell the TBB what activity is taking place. -- Register 3 of the I/O space.
- 2) A data register that is used to pass the actual command or option data. -- Register 4 of the I/O space.
- 3) A ready flag that is used by the TBB to determine when a transfer request has occurred. -- 8031A Port 1 bit 0.

Externally the channel consists of:

- 1) An 8 bit data bus. (D7....D0)
- 2) One address bit that selects either the control register or the data register. (C/-D)
- 3) A read signal (DRD-)
- 4) A write signal (DWT-)
- 5) A flag (RDY-) that can be used to interrupt the TAD when information is present to read or when a write transaction is complete. The status of this flag is also available by reading the control register bit 0 if interrupts are not desired.

7.1.1. Timing

The physical cycles for accessing the TBB registers are shown as follows.

For a write the C/-D line must be present 30ns before the falling edge of the write and the data must be present at least 20ns before the rising edge of the write. The rising edge of the write clocks an edge triggered 8 bit D latch and therefore no hold time is required after the edge.

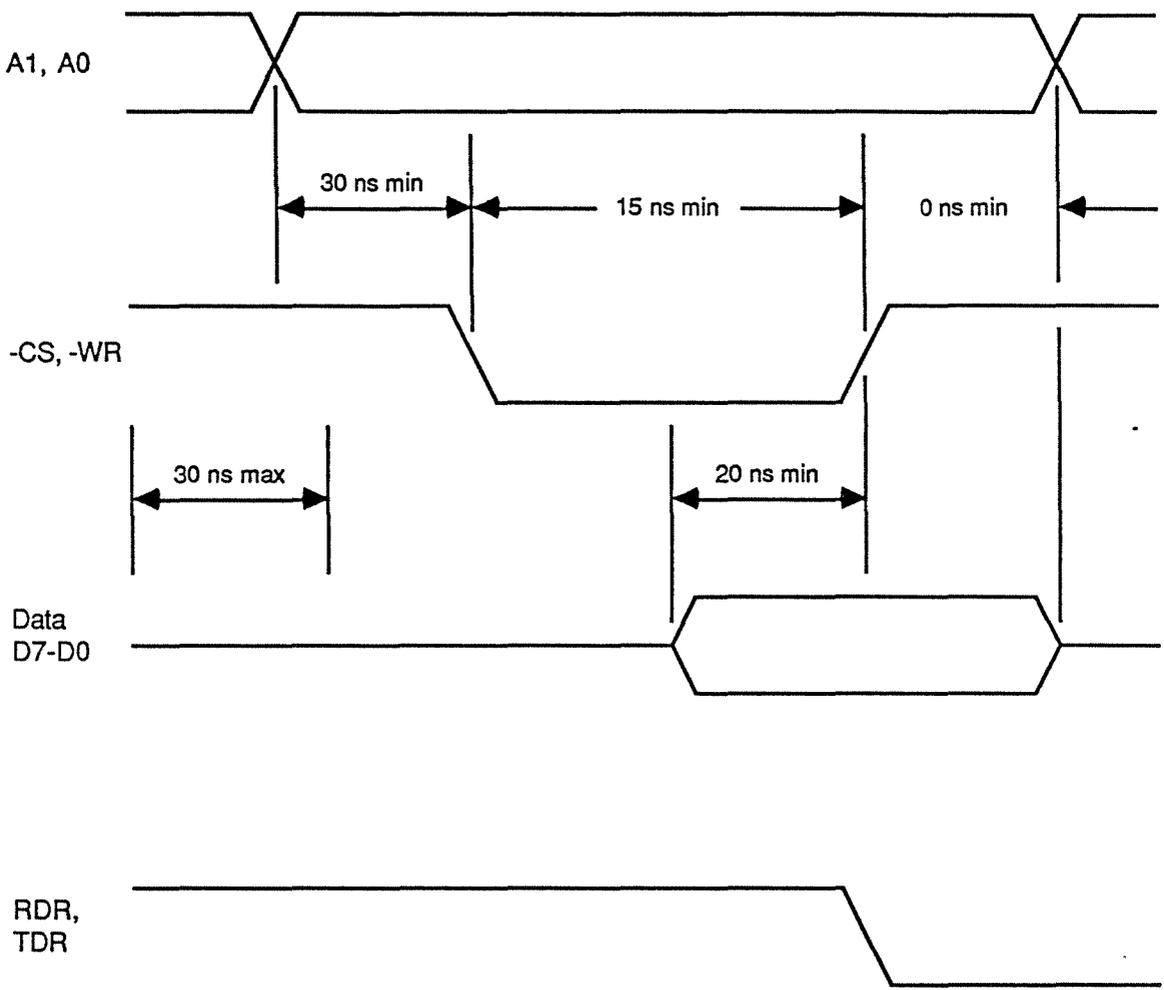


Figure 7-2. Write Cycle Timing

For a read the C/-D line must be valid at least 30 before the falling edge of the read. The data will be valid on the bus within 30ns. After the read is removed it takes 30ns for the bus to return to high impedance state.

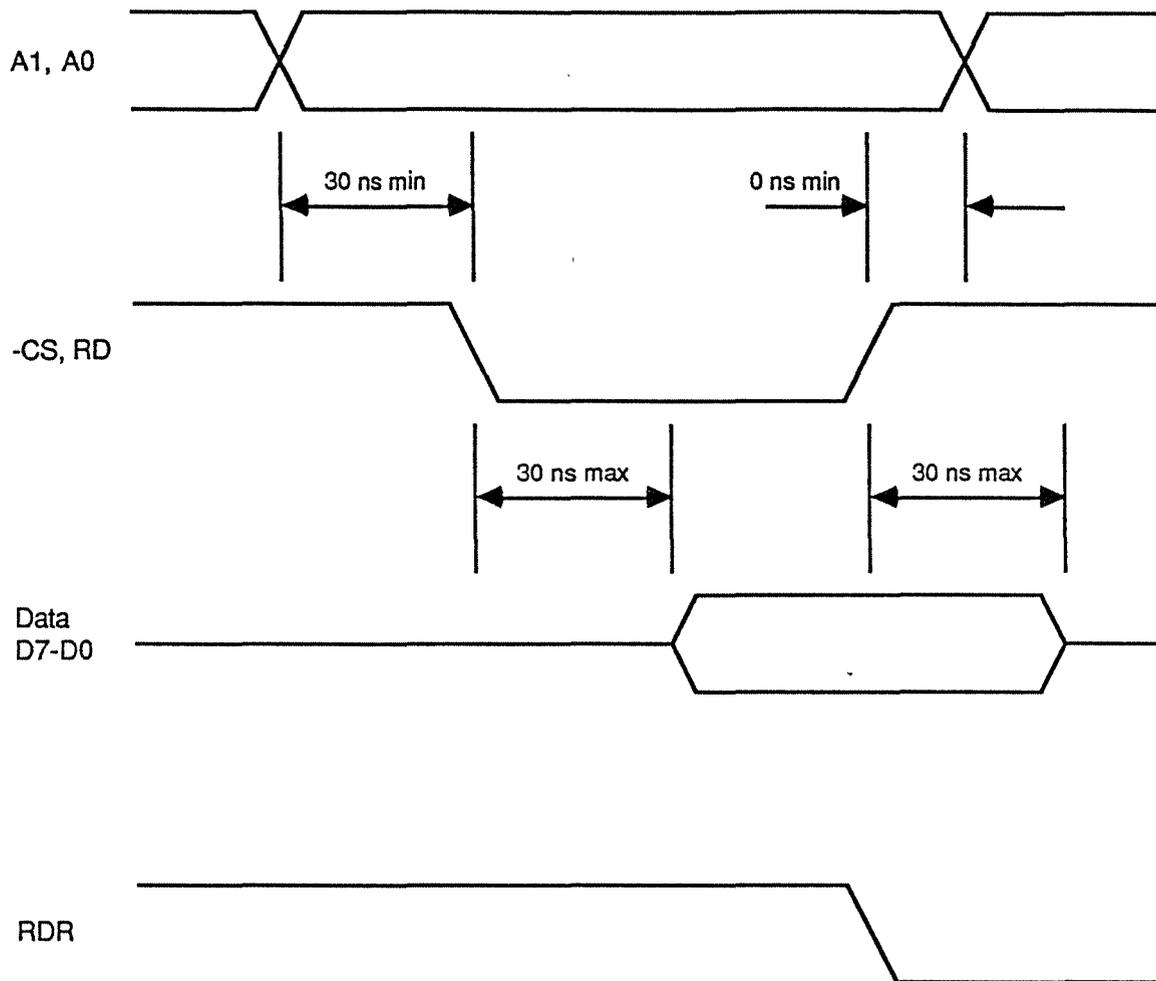


Figure 7-3. Read Cycle Timing

7.2. Register design.

The control register is the key to passing information between the TBB and the TAD. When written, the internal 8 bits used as follows:

- 1) Bit 0 -- R/-W indicates whether the TAD requests a read or a write.
- 2) Bit 1,2,3 -- Spare
- 3) Bits 7 6 5 4 -- Control Code as follows;

Note that bit 7 is reserved for expansion capability. Also note that not all codes are used and that the code space is not in sequence.

When this byte is written to the Command Register								TBB will read from the Data Register
D7	D6	D5	D4	D3	D2	D1	D0	
	0	0	0				0	Command Message
	0	0	1				0	Loopback Message
	1	0	0				0	Configuration 0 Message
	1	0	1				0	Configuration 1 Message
	1	1	0				0	Configuration 2 Message
	1	1	1				0	Configuration 3 Message
								TBB will write to Data Register
	0	0	0				1	Status Message
	0	0	1				1	Loopback Message
	1	0	0				1	Configuration 0 Message
	1	0	1				1	Configuration 1 Message
	1	1	0				1	Configuration 2 Message
	1	1	1				1	Configuration 3 Message
X				X	X	X		not used

Table 7-1. TBB Control Register Codes

When the control register is read only bit 0 is valid. It gives the status of the ready flag.

The data register contains the actual command or option information which are discussed in section 12.

7.3. Operation

The control codes tell the TBB whether the access is to pass command/status information to/from the TBB, or option data. The actual command/status or option data will be passed via the data register. This design requires a low level state machine to pass the information and a high level state machine to control the commands and states. The low level state machine operations will be described here and the high level operation is described in section 13. The low level state machine must allow writes and reads to/from the TBB.

7.3.1. Write to the TBB

The following describes the sequence of events that take place when the TAD wishes to pass either commands or option data to the TBB.

- 1) TAD writes the data register (R/-W=0,C/-D=0) with the information to pass to the TBB.

- 2) TAD writes the control register (R/-W=0,C/-D=1) with the internal read/write bit (B0) set low (write) and the correct code for the operation required.
- 3) This causes the RDY Flag to be deactivated to indicate that the TBB is not ready as it is processing the access.
- 4) The RDY Flag is read by the 8031A on its next background cycle.
- 5) The 8031A will then read register 3 (control register) and see the R/-W bit set to 0 (Write).
- 6) The 8031A reads register 4 (data register) and stores the data in the internal RAM. A background routine will inspect the command register and act on the command. See section 13 for a description of how the commands are used to control the TBB.
- 7) When the TBB reads the data register it activates the RDY line/bit. This indicates that the TBB is ready to accept the next control register write.

7.3.2. Read from TBB

The following describes the sequence of events that occur when the TAD reads the TBB to obtain the TBB status or read the option registers.

- 1) The TAD writes the control register indicating a read (R/-W bit set high) and the register requested.
- 2) This causes the RDY flag to be deactivated to both the 8031A port 1 bit 0 and to the TAD via the RDY line or the RDY bit.
- 3) The 8031A detects the RDY flag low during the background cycle.
- 4) The 8031A reads the control register and uses the register code to get the data from internal RAM.
- 5) The 8031A writes the data to the data register.
- 6) This causes the RDY flag to be set high indicating that the data register contains valid data.
- 7) The TAD must have been polling or be waiting for the RDY interrupt and now reads the data register.

7.3.3. Ready Flag

The ready flag is used as a handshake line that allows the TAD and the TBB to operate without wait states on the bus. This is necessary due to the fact that the 8031A may take up to 10ms to respond to requests when it is busy.

The TAD may inspect the status of the RDY flag by:

- 1) Using the Ready line to interrupt the TAD processor and jumping to an interrupt routine that sets the status of the TBB RDY flag to clear.

- 2) Connecting the RDY line to a register or I/O port to poll it.
- 3) Reading the control register (C/-D set high, R/-W set high) and inspecting bit 0 on the data bus.

8. MISC

The operation of the Reset circuitry and the Clocks in the TBB are discussed in this section.

8.1. Hard Reset

The Hard Reset line (RST) on the 8031A is controlled by 3 sources:

- 1) The external TBB reset line (RST).
- 2) The internal Watchdog Counter(WDC).
- 3) The Power-up Reset circuit.

If the TAD wishes to force the TBB to its initial state (for power-up or in case of problems) it activates the external RST line.

The Watchdog Counter is used to reset the 8031A (and the TBB) in the event that the TBB malfunctions. The WDC is a free running counter that if left alone will reach a final count which causes the 8031A RST line to be activated. In normal operation the firmware performs a write to the WDC (Write Reg 7) that clears the counter before it can reach its timeout count. The timeout is preset at 125ms. If a problem occurs and the firmware cannot write the register it resets the entire TBB which would enter the selftest mode.

The Power-up Reset circuit consists of a slowly charging RC network that forces a reset of the 8031A .5 seconds after the power-up. This ensures that all hardware has stabilised before the processor is reset.

8.2. TBB Clock Circuits

There are 4 clock circuits that are of importance in the TBB. They are:

- 1) The Network Synchronization Clock (NSC).
- 2) The Synchronous Data Clocks circuits (SDC).
- 3) The Asynchronous USART Clock (AUC).
- 4) The Watchdog Timer Clock (WTC)

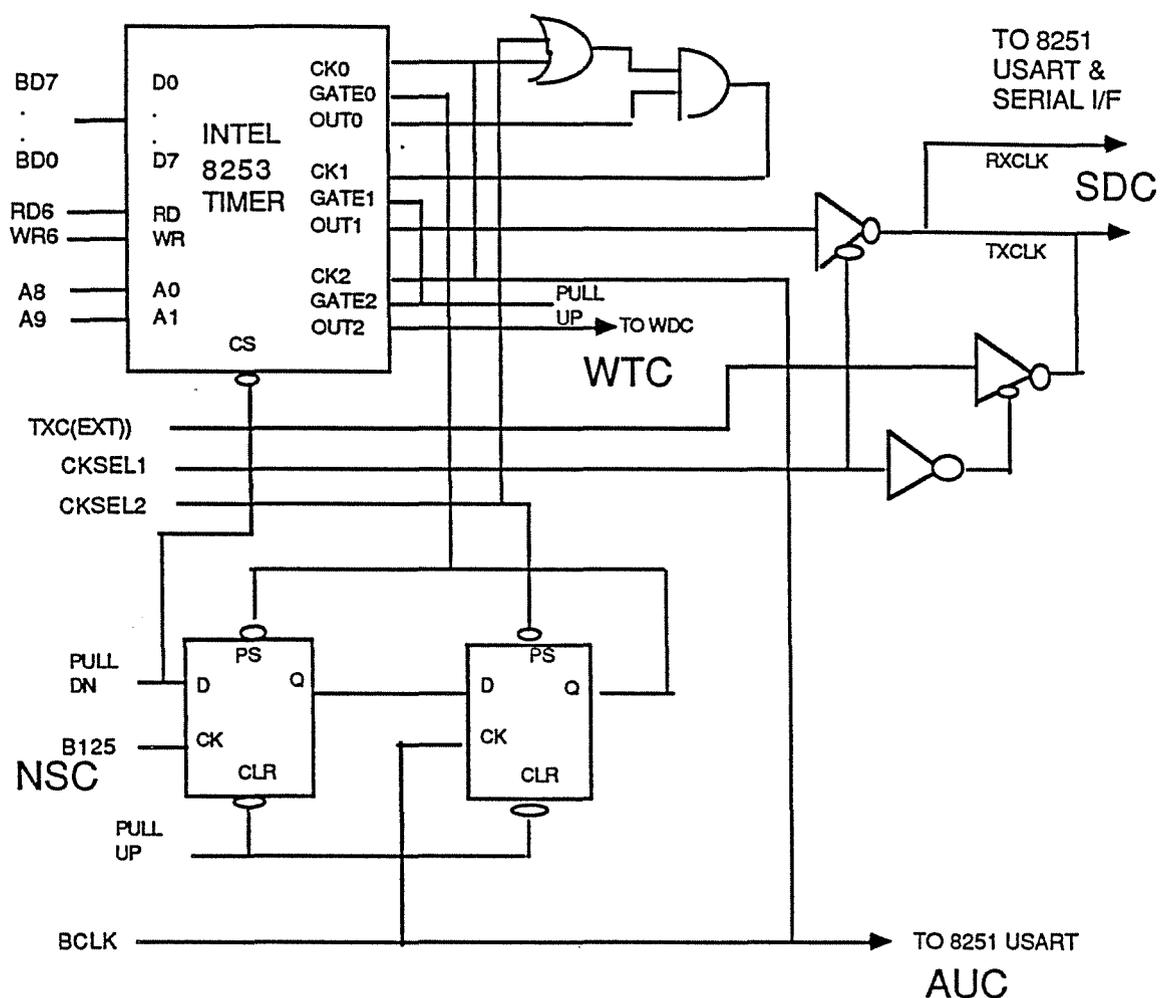


Figure 8-1. TBB Clock Circuits

The NSC is the 125 us interrupt at the Network Data Channel (B125). It is used by the INTEL 8253 Counter/Timer to maintain TBB synchronization with the digital network. When the interrupt occurs the 8253 will speed up or slow down the data clock slightly to insure that the terminal data link never gets out of sync with the network data link.

The SDC is generated by the 8253 from the 2.56MHz clock (Bclk) used to shift data in and out of the network I/F. The SDC is used to source both the USART synchronous clocks and the Serial Terminal Interface clocks .

The AUC is also created directly from the 2.56MHz clock (Bclk) and is passed to the INTEL 8251 USART. This ensures that the async data rate will be in sync with the TAD and the network.

The WTC is created directly from the 2.56MHz clock (Bclk) and is divided down to create an .6s clock at the output of the watchdog timer.

It is very important that all the clocks provided to the TBB be edge synchronised to each other. The 125us clock (B125) is a divided version of the 2.56MHz clock. (divided by 320)

8.3. Power/Grounding

The TBB requires only +5V and a power ground.

9. Data Register Bit Interpretation

The bit maps for the data register that corresponds to the control codes will be described in the following sections.

The extra bits and registers are intended to provide for future expansion and the flexibility to handle other Rate Adaption Protocols that may require more I/O. All unused bits are set to 0.

9.1. Command/Status Register - 0000

This register is to be used by the TAD to command the TBB and poll the status of the TBB. When commanding the TBB only bits D0,D1 and D2 are used, bits D3-D7 are not used. When reading the status of the TBB Bits D1,D2 and D3 are used to indicate which state the TBB is in and bits D3-D7 are used to indicate any additional status.

<u>D2 D1 D0</u>	<u>States</u>
0 0 0	Idle
0 0 1	Setup
0 1 0	Spare
0 1 1	Spare
1 0 0	Parameter Exchange
1 0 1	Data Mode
1 1 0	Spare
1 1 1	Spare

Table 9-1. State Table

The commands are always written by the TAD and the status is read by the TAD. The following table shows the list of commands and status.

State	Status Bits					State Code			Meaning	
	D7	D6	D5	D4	D3	D2	D1	D0		
Idle	0	0	0	0	0	0	0	0	Go to Idle	C
	0	0	0	0	0	0	0	0	TBB busy	S
	0	0	0	0	1	0	0	0	TBB idle	S
Setup	0	0	0	0	0	0	0	1	Go to Setup	C
	0	0	0	0	0	0	0	1	Checking configuration	S
	0	0	0	0	1	0	0	1	Bad setup	S
	0	0	0	1	1	0	0	1	Good setup	S
Parameter Exchange	0	0	0	0	0	1	0	0	Go to PX (originate)	C
	0	0	0	0	1	1	0	0	Go to PX (answer)	C
	0	0	0	0	0	1	0	0	PX in progress	S
	0	0	0	0	1	1	0	0	Sgvi received	S
	0	0	0	1	0	1	0	0	Inband sync found	S
	0	0	0	1	1	1	0	0	PX complete	S
Data Mode	0	0	0	0	0	1	0	1	Go to Data Mode	C
	0	0	0	0	0	1	0	1	TBB not in Data Mode	S
	0	0	X	X	1	1	0	1	TBB in Data Mode	S
				0	1	1	0	1	- far-end not looped	S
				1	1	1	0	1	- far-end looped	S
			0		1	1	0	1	- near-end not looped	S
			1		1	1	0	1	- near-end looped	S
		0			1	1	0	1	No restart received	S
		1			1	1	0	1	Restart received	S
	0				1	1	0	1	In sync	S
	1				1	1	0	1	Inband sync lost	S
Self-Test	0	0	0	0	0	1	1	0	Go to Self-Test	C
	0	0	0	0	0	1	1	0	Fail self-test	S
	0	0	0	0	1	1	1	0	Pass self-test	S

Table 9-2. Command/Status Register

When a command is sent to the TBB the state status register will continue showing the old state until the TBB has had time to update it. This prevents race conditions between the TAD and the TBB. Note that there is only 2 main state status in each state with some sub status bits where required.

For an explanation of how the states are used in operation see section 13.

In general when the TBB reads a register it will assume that configuration. When the register is read by the TAD it will always reflect the present configuration. The TBB will exchange parameters with the far end and once the parameter exchange is complete the TBB

will update the registers as follows. If the parameter exchange is successful the operational parameters will be written into the status registers. The parameters will only change if the near end has adapted to the far ends parameters.

If any parameters are incorrect or not allowed, the TBB will set the "invalid parameter" bit in the state status after reading all the params. Thus the status register (0000) would read 00001001.

9.2. Loopback Control - 0001

This register is used to drive/sense the status leads of the DTE/DCE interface. The only leads supported in the TBB are used to command and detect the Far Loopback condition. When the RLB bit is set the TBB will set the far loopback request bit in the T-Link parameter exchange. If the loop is successful the T-Link handshake will be successful and the TI bit will be turned on. If the near DCE is looped back to the far DTE as a result of a loopback request the TI bit will be activated.

9.2.1. Write (by TAD)

```
D7 D6 D5 D4 D3 D2 D1 D0
0  0  0  0  1  0  0  0 - Request far end loopback
```

9.2.2. Read (by TAD)

```
**  D7 D6 D5 D4 D3 D2 D1 D0
    0  0  0  0  0  0  0  0 - no loop requested by far end.
    0  0  0  0  1  0  0  0 - loop requested by far end.
```


9.4. Config 1 - General - Register 0101

This register is used to further option the TBB.

D7	D6	D3	D4	D5	D2	D1	D0	Meaning
							0	Normal RTS
							1	Assert RTS
						0		Normal DTR
						1		Assert DTR
X	X	X	X	X	X			not used

Table 9-4. Config 1 Register

9.5. Config 2 - General - Register 0110

This register is used to configure the parameters for asynchronous or synchronous operation when written to the TBB and to indicate the parameters in use when read by the TAD.

9.5.1. Asynchronous Parameters

D7	D6	D5	D4	D3	D2	D1	D0	Meaning				
								0	0	5 bits	Word Length	
								0	1	6 bits		
								1	0	7 bits		
								1	1	8 bits		
								0	0	1 stop bit	Stop Bits	
								0	1	1.5 stop bits		
								1	0	2 stop bits		
								1	1	spare		
								0	0	0	Parity	
								0	0	1		odd
								0	1	0		even
								0	1	1		spare
								:	:	:		:
1	1	1	spare									
X								not used				

Table 9-5. Config 2 Register

For Example; 7 bit characters, odd parity, 1 stop bit

D7 D6 D5 D4 D3 D2 D1 D0

0 0 0 1 0 0 1 0

9.5.2. Synchronous

D7...D0 not used

9.6. Config 3 Rate selection/indication - Register 0111

The following table contains the speeds available in the TBB. The TAD uses this register to set the desired rate by writing to the register. If after the parameter exchange occurs the TBB has adapted to a different speed then this will be reflected in this register when read by the TAD. The rate table contains some speeds that are only available for sync or async.

D7	D6	D3	D4	D5	D2	D1	D0	Speed			
								(bps)	Async	Sync	
			0	0	1	1	0	300	X		
			0	1	0	0	0	1200	X	X	
			0	1	0	1	1	2400	X	X	
			0	1	1	0	1	4800	X	X	
			0	1	1	1	1	9600	X	X	
			1	0	0	0	1	19200	X	X	
			1	0	1	0	0	48000		X	
			1	0	1	1	0	56000		X	
			1	0	1	1	1	64000		X	
X	X	X							not used		

Table 9-6. Rate Table

10. TBB Operation

The command/status register (0000) in the Command/Status Channel is the key to operating the TBB. This register is used to allow the TBB to accept commands and return status to the outside world. The following describes the actions that occur when the TBB is considered a black box.

As a TAD progresses through the states of a data call it will require control/status of the TBB. The following Figure shows the states that the TBB will pass through during operation.

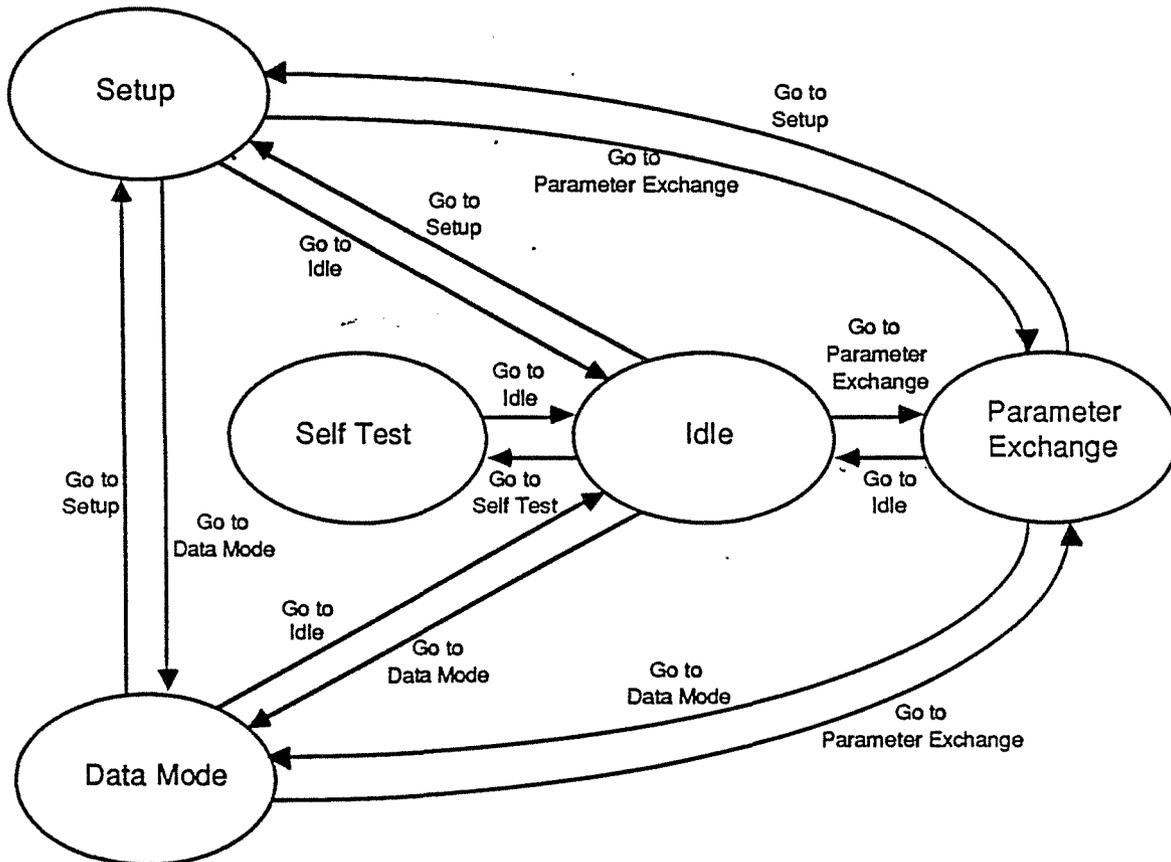


Figure 10-1. ** State Diagram

10.1. Sample State Sequence

A sample sequence is as follows.

- 1) ** The TBB is powered up and automatically enters the self test mode. If the self test fails the Self Test State will be set to "Failed" by the TBB. If self test is successful the TBB will set the status to "pass" and await further instructions from the TAD. It is up to the TAD to read the TBB status register to determine its present state and status.
- 2) The TAD will command the TBB to get to idle mode until it is ready to pass the T-Link options to the TBB. It may command the TBB to go directly to

data mode in some applications where the channel is to be used without the rate adaption protocol.

- 3) From the idle state the TBB will be commanded to enter the setup state to read the optionable parameters. The option registers are first written by the TAD. After all parameters are passed the TAD writes the command register to go to "setup" mode. The TBB then reads the registers, examines them and sets the state status register to show "setup complete" and whether or not the parameters are valid. If the status shows invalid then the TAD must reenter all the parameters. If the options have already been set in a previous call the TAD may command the TBB to go directly to parameter exchange when a call is detected or requested. If the TAD only wishes to change 1 option register it may do it and then send the setup command.
- 4) When the TAD has established a call it commands the TBB to enter parameter exchange. The TBB will use the parameters that it has been passed earlier. The TBB status will show that handshake is proceeding. At this time the status registers will reflect the options set by the TAD. If the parameter exchange fails the status register 0000 will reflect this. It is up to the TAD to decide on corrective action. If the parameter exchange is successful and the near end has adapted the TBB will write the operational parameters into the registers and then write the state status register to "complete". The TAD polls the state register and when it sees status "complete" it can read the parameters that the near end has adapted to. If the parameter exchange is successful the status will reflect this and the TBB will wait until told to go to data mode. At this time the TBB will pass data fill on the network I/F.
- 5) ** When commanded to go to data mode the TBB will reflect this in the status register. If a loopback has been established by either the far end or the near end this will be reflected in the state status register. The TAD may command the TBB to Restart the T-Link parameter exchange by writing that command to the TBB. This would be used to renegotiate parameters or to end the far loop condition. Dueto the T-Link protocol there is no method to end a high speed (48,56, 64Kbit/s) loopback except by taking down the line. The TAD may Reset the TBB to clear a call by writing the idle command to the command register. If sync is lost or a restart received from the far end during data mode the TBB will set the status to reflect "Sync Lost" or "Restart received". It is up to the TAD to realise that the call has failed by occasionally reading the TBB status.

11. Applications

TADs provide a transparent layer of protocol to ensure error-free transport of data between CPE DCEs. A TAD typically consists of:

- 1) Terminal I/F (eg. RS232C/V.24,V.35,RS449,V.36,WE303)
- 2) Human I/F (eg. DIP Switches, Keypads, Serial dialing I/F)
- 3) Rate Adaption Block
- 4) Network I/F

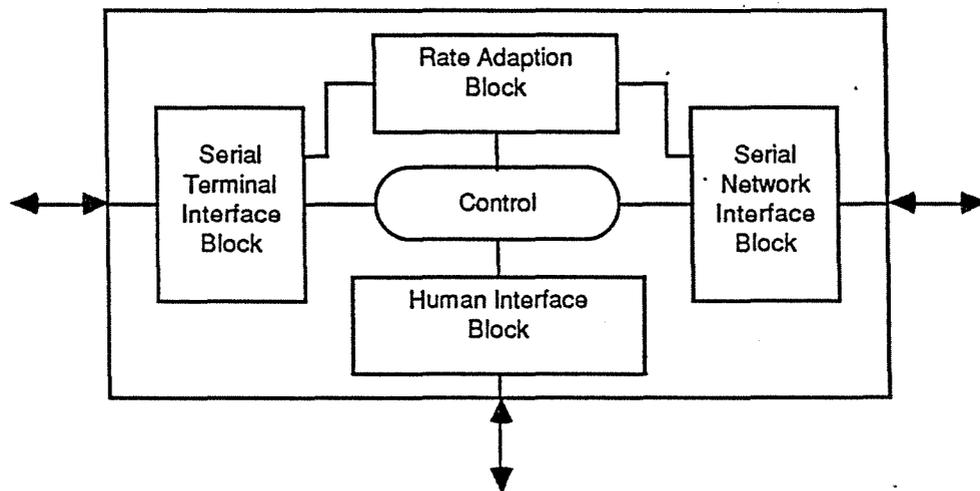


Figure 11-1. TAD Blocks

The Terminal I/F, Human I/F and Control are application dependent and are not discussed here.

The Rate Adaption Block is independent of the other blocks. If the interface is designed in a general way the building block can be changed to accommodate different forms of rate adaption that may be required for other products. The T-link Building block specified in this document has been designed so that the interface is in a form general to rate adaption and not specific to T-Link.

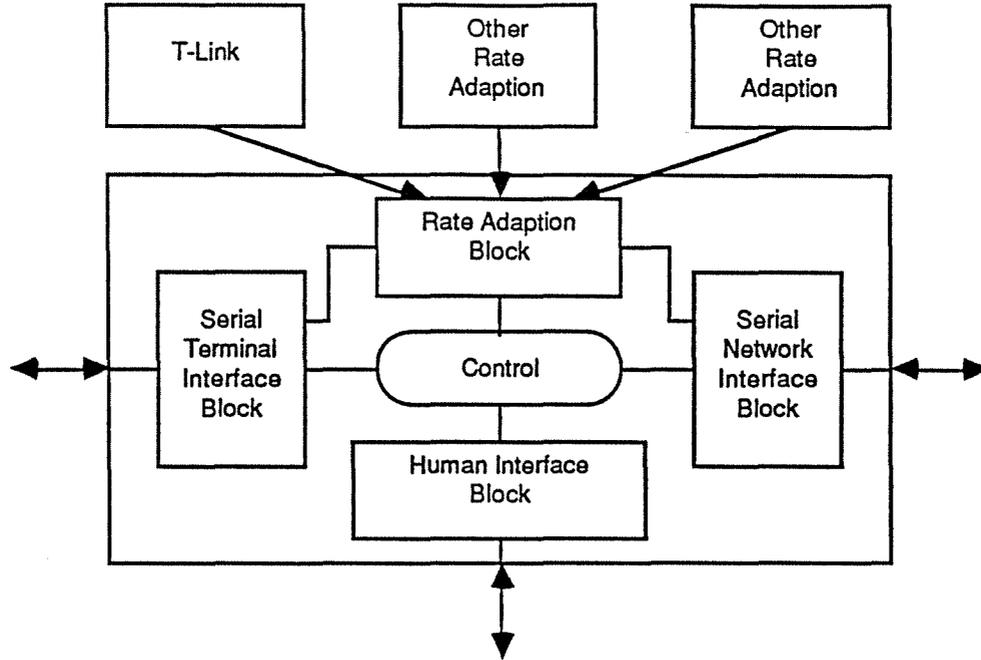


Figure 11-2. Protocol Independence

Also because the Rate Adaption Building Block is independent of the other blocks it can be used in different networks where the line I/F is closely linked to the network Vendor. This allows the same T-Link Building Block to be used in DMS Datapath, DMS ISDN, other vendors PSDS and other vendors ISDN applications.

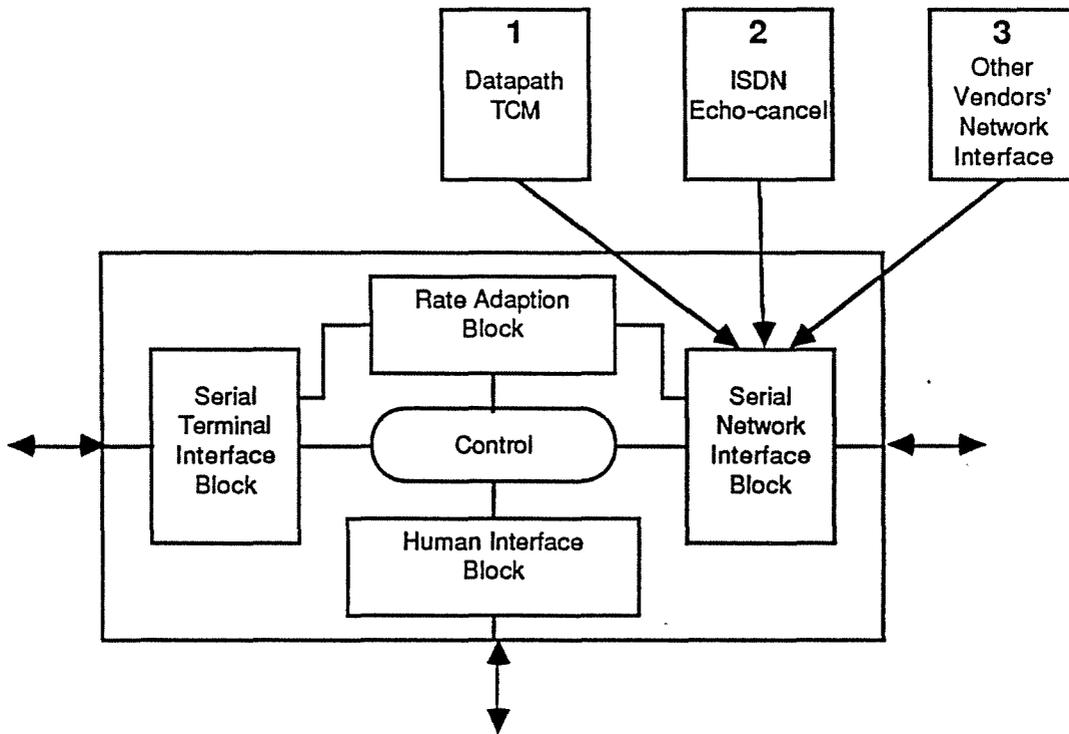


Figure 11-3. Network Independence

11.1. Datapath TAD

The following special building blocks are required for a DMS Datapath TAD:

- 1) TBB
- 2) Network I/F

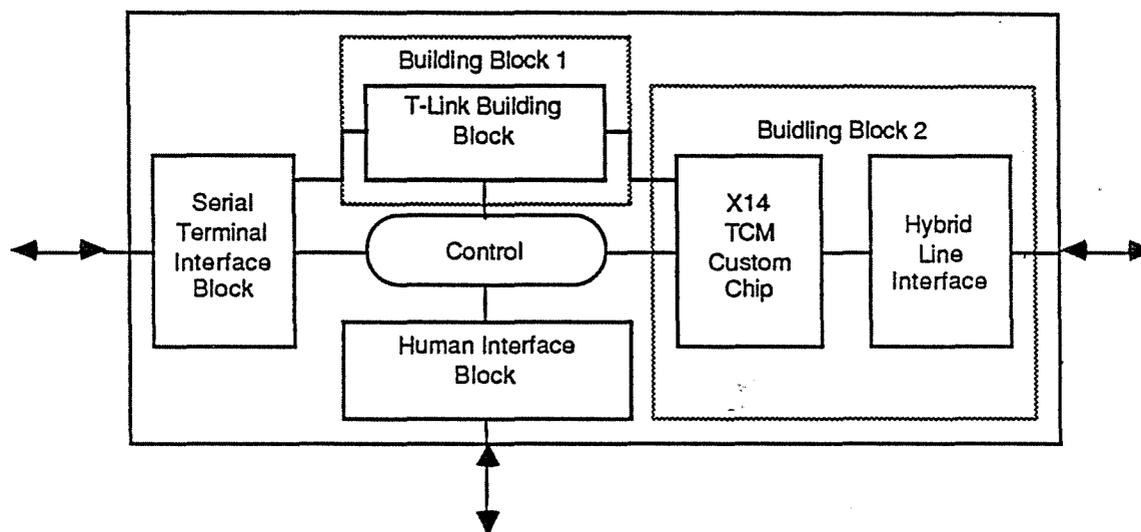


Figure 11-4. Datapath TAD

The TBB can be in any of the forms discussed in section 6. The Network I/F building block consists of 3 parts.

- 1) The Line I/F Hybrid
- 2) The X14 TCM Chip
- 3) The control software for the DMS network I/F

The line hybrid is a thick-film ceramic-substrate with all the required analog components to connect to the 2-wire telephone line.

The X14 TCM chip provides the low level Time Compression Multiplex protocol. This protocol provides a 64Kbit/s data channel and an 8Kbit/s control channel in both directions on the line by transmitting at 160Kbit/s in each direction. This is sometimes called ping-pong format. To each DCE the channels appear full-duplex.

The 8Kbit/s channel is used to pass control information used by the DMS switch and the TAD to set up calls, and control the rich feature set of the Datapath service.

This Network I/F Block will be specified in another document at a later date.

11.2. ISDN TAD

For ISDN the TAD must adapt terminal data to an ISDN B-Channel. This B-Channel may be attached to the T-Bus or it may be integrated with the NT1 and attached to the U-Bus. In either case the output of the TBB can be fed into the line I/F.

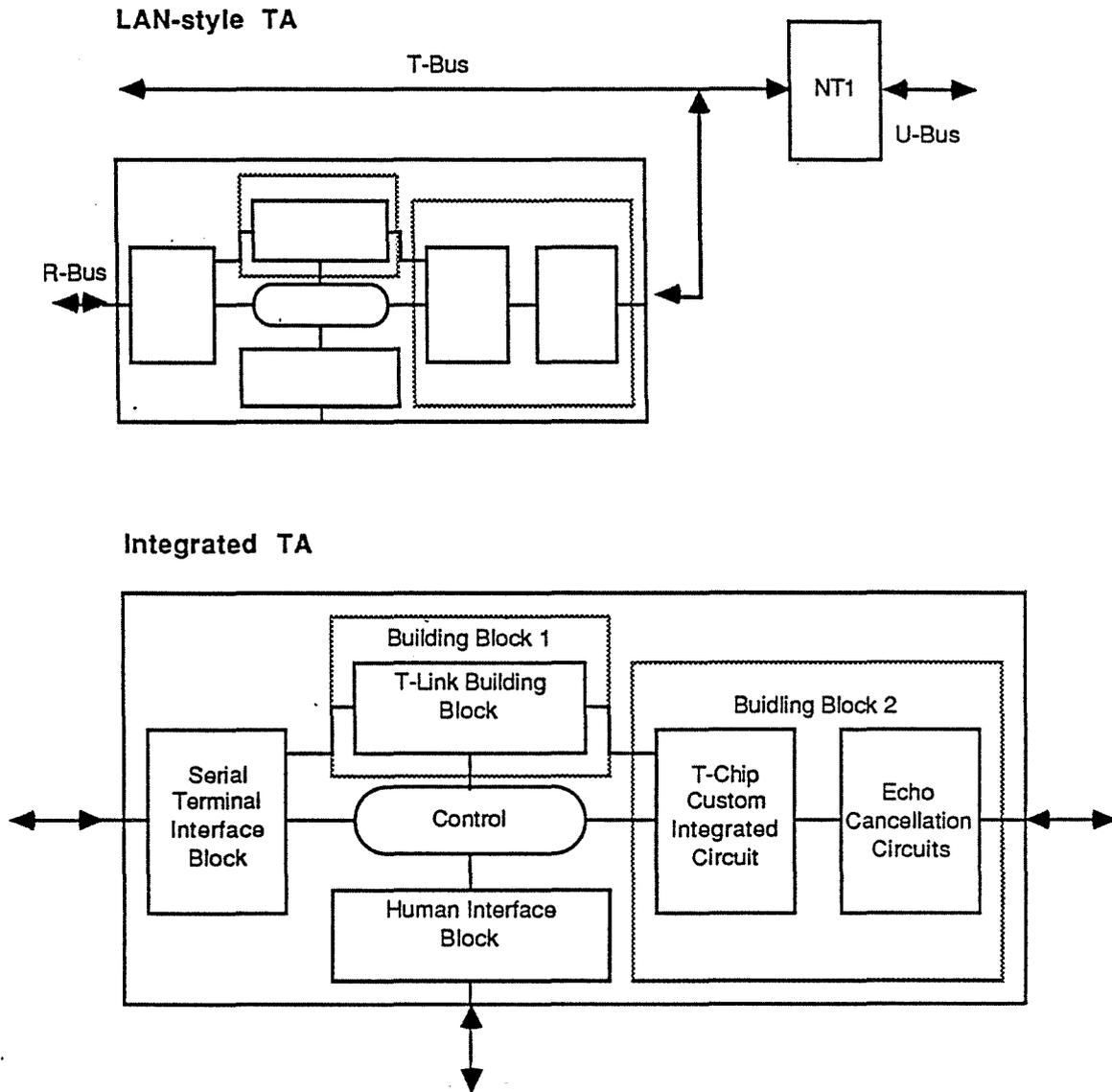


Figure 11-5. ISDN TAD

