PRODUCT DEFINITION AND
HARDWARE IMPLEMENTATION
FOR AN INTELLIGENT NETWORK TERMINATION

Report on a Master thesis project performed by Maurice J.L. Magdelijns
from the Eindhoven University of Technology
Faculty of Electrical Engineering, Department of Digital Systems

ISDN Access Systems Group
APT NEDERLAND B.V.
PREFACE

The final part of the Electrical Engineering study at the Eindhoven University of Technology (EUT) is the Masters thesis project. As a student at the Digital Systems department of the Electrical Engineering faculty I got the opportunity to perform this thesis project at AT&T and Philips Telecommunications (APT) in Hilversum, in particular the ISDN Access Systems Group (IAS).

The following persons were responsible for this project: ir. S.J.M. Tol and ir. J.E.W. Winkelman at APT and prof. ir. M.P.J. Stevens at EUT.

I would like to thank all members of the IAS Group who supported me during the establishment of my project especially Simon Tol and Bert Winkelman for their constructive criticism and intensive coaching. Furthermore I thank Jaap de Wildt (Delft University of Technology) for the pleasant and useful co-operation during the project. They all took part in the achievement of this report.

Maurice J.L. Magdelijns

Hilversum, September 1988
SUMMARY

The intelligent Network Termination, a product providing switching capability for B-channels added to the basic NT1 functions, offers features specified in a product definition. Based upon residential and small business applications and upon acceptability of the development efforts required, there has been decided upon implementation of the following calling features: hold, transfer, call waiting and three-way calling.

The basic intelligent NT configuration as it is studied in this report provides 2 U and 4 S interfaces (which can be expanded due to system modularity). Connecting up to two terminals (using circuit switched B-channels) per S-bus circuit prevents blocking at the S-bus as well as within the switching system of the NT supposing a 50%-50% internal-external call ratio per terminal. Optionally, the intelligent NT provides a packet switching capability via the D-channel. The hardware supports this already in the basic NT configuration.

Two different configurations are determined for the switching design: centralized and decentralized switching. Centralized switching turns out to be most applicable as it provides simpler manufacturing when Eurocards are used: smaller busstructure, fewer components.

For the centralized switching design a hardware implementation is presented. The components have been selected taking into account several aspects: costs, manufacturing and technical applicability. A price indication of the architecture is given based upon the key component prices.

As for the decentralized switching design, a possible implementation is started. Though not studied in detail, this architecture seems to be applicable for the intelligent NT as well. Further study on this configuration is recommendable in order to give a price comparison between the centralized and decentralized implementations.
### ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT</td>
<td>AT&amp;T and Philips Telecommunications</td>
</tr>
<tr>
<td>BAMX</td>
<td>Basic Access MultipleXer (APT)</td>
</tr>
<tr>
<td>CCIIT</td>
<td>Comite Consultatif International de Telephonie et Telegraphie</td>
</tr>
<tr>
<td>ch</td>
<td>channel</td>
</tr>
<tr>
<td>CI</td>
<td>Channel Identifier</td>
</tr>
<tr>
<td>CSU</td>
<td>Central Switching Unit</td>
</tr>
<tr>
<td>Clk</td>
<td>Clock</td>
</tr>
<tr>
<td>DMA</td>
<td>Direct Memory Access</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>DSU</td>
<td>Decentralized Switching Unit</td>
</tr>
<tr>
<td>HDLC</td>
<td>High level Data Link Controller</td>
</tr>
<tr>
<td>i8086</td>
<td>Intel component (e.g. 8086)</td>
</tr>
<tr>
<td>IAS</td>
<td>ISDN Access Systems</td>
</tr>
<tr>
<td>IBI</td>
<td>IST Bus Interface circuit</td>
</tr>
<tr>
<td>IEC</td>
<td>ISDN Echo Cancellation circuit (Siemens)</td>
</tr>
<tr>
<td>IOM</td>
<td>ISDN Oriented Modular interface (Siemens)</td>
</tr>
<tr>
<td>ISAC</td>
<td>ISDN Subscriber Access Controller (Siemens)</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>IST</td>
<td>Integrated Services Terminal</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAPB</td>
<td>Link Access Protocol for the B-channel</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LAPD</td>
<td>Link Access Protocol for the D-channel</td>
</tr>
<tr>
<td>MCU</td>
<td>Micro Controlling Unit</td>
</tr>
<tr>
<td>MPI</td>
<td>Micro Processor Interface</td>
</tr>
<tr>
<td>Mux(ing)</td>
<td>Multiplexing</td>
</tr>
<tr>
<td>NT</td>
<td>Network Termination</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>RClk</td>
<td>Receive Clock</td>
</tr>
<tr>
<td>SAPI</td>
<td>Service Access Point Identifier</td>
</tr>
<tr>
<td>SBC</td>
<td>S-Bus interface Circuit (Siemens)</td>
</tr>
<tr>
<td>SLD</td>
<td>Subscriber Line Data (Siemens)</td>
</tr>
<tr>
<td>SNIC</td>
<td>Subscriber Network Interface Circuit (Mitel)</td>
</tr>
<tr>
<td>ST-bus</td>
<td>Serial Telecom bus (Mitel)</td>
</tr>
<tr>
<td>TCk</td>
<td>Transmit Clock</td>
</tr>
<tr>
<td>TE</td>
<td>Terminal Equipment</td>
</tr>
<tr>
<td>TEI</td>
<td>Terminal Endpoint Identifier</td>
</tr>
<tr>
<td>TSA</td>
<td>TimeSlot Assignment</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>CONTENTS</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td></td>
</tr>
<tr>
<td>2. Productdefinition intelligent NT</td>
<td></td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td></td>
</tr>
<tr>
<td>2.2 Calling-features</td>
<td></td>
</tr>
<tr>
<td>2.2.1 Feature overview</td>
<td></td>
</tr>
<tr>
<td>2.2.2 Explanatory listing of the features</td>
<td></td>
</tr>
<tr>
<td>2.2.3 Classification of the features</td>
<td></td>
</tr>
<tr>
<td>2.2.3.1 Why classification?</td>
<td></td>
</tr>
<tr>
<td>2.2.3.2 The selection of the classification items</td>
<td></td>
</tr>
<tr>
<td>2.2.3.3 Explanation of the classification items</td>
<td></td>
</tr>
<tr>
<td>2.2.3.4 The classified feature table</td>
<td></td>
</tr>
<tr>
<td>2.2.4 How the features might be implemented</td>
<td></td>
</tr>
<tr>
<td>2.2.5 Selection of the features</td>
<td></td>
</tr>
<tr>
<td>2.3 Technical features</td>
<td></td>
</tr>
<tr>
<td>2.3.1 Introduction</td>
<td></td>
</tr>
<tr>
<td>2.3.2 Interfaces</td>
<td></td>
</tr>
<tr>
<td>2.3.3 Terminals</td>
<td></td>
</tr>
<tr>
<td>2.3.4 Circuit switched call routing configurations</td>
<td></td>
</tr>
<tr>
<td>2.3.5 Packet switching option</td>
<td></td>
</tr>
<tr>
<td>2.3.6 The basic NT configuration</td>
<td></td>
</tr>
<tr>
<td>2.3.7 System configuration</td>
<td></td>
</tr>
<tr>
<td>2.3.8 Expansion</td>
<td></td>
</tr>
<tr>
<td>3. Pre-study for the hardware design of the int. NT</td>
<td></td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td></td>
</tr>
<tr>
<td>3.2 Study on int. NT resembling systems</td>
<td></td>
</tr>
<tr>
<td>3.2.1 Introduction</td>
<td></td>
</tr>
<tr>
<td>3.2.2 Alcatel's intelligent NT</td>
<td></td>
</tr>
<tr>
<td>3.2.3 Siemens patent application (Ref. [4])</td>
<td></td>
</tr>
<tr>
<td>3.2.4 The star NT</td>
<td></td>
</tr>
<tr>
<td>3.2.5 Evaluating the systems</td>
<td></td>
</tr>
<tr>
<td>3.3 Component dependent interfaces between the S and U reference points</td>
<td></td>
</tr>
<tr>
<td>3.3.1 Introduction</td>
<td></td>
</tr>
<tr>
<td>3.3.2 The K(2)-interface</td>
<td></td>
</tr>
<tr>
<td>3.3.3 The IOM interface</td>
<td></td>
</tr>
<tr>
<td>3.3.4 ST-bus</td>
<td></td>
</tr>
<tr>
<td>3.3.5 Compatibility of the interfaces</td>
<td></td>
</tr>
<tr>
<td>3.4 Timing considerations at a multiple U-transceiver interface</td>
<td></td>
</tr>
<tr>
<td>3.4.1 Master clock generation</td>
<td></td>
</tr>
<tr>
<td>3.4.2 Activation and deactivation</td>
<td></td>
</tr>
</tbody>
</table>
4. Hardware configurations for the intelligent NT
   4.1 Introduction 4-1
   4.2 Targets for the hardware design 4-1
   4.3 D-channel processing considerations 4-2
       4.3.1 Introduction 4-2
       4.3.2 Separate D-channels 4-2
       4.3.3 Terminals sharing a D-channel 4-3
           4.3.3.1 Low layer-2 support by a single channel HDLC unit -both logical and physical common D-channel- 4-4
           4.3.3.2 Low layer-2 support by a multiple channel HDLC unit -logical common D-channel- 4-5
   4.4 Packet switched data support: general considerations 4-6
       4.4.1 Considerations using the OSI model 4-6
           4.4.1.1 Different D-data processing configurations supporting packet switching 4-10
   4.5 Two approaches for the switching system design 4-14
   4.6 Centralized switching 4-15
       4.6.1 Introduction 4-15
       4.6.2 Two concepts 4-15
           4.6.2.1 Functional description centralized switching architecture: all D-data to central unit (concept 1) 4-16
           4.6.2.2 Packet switching support 4-19
           4.6.2.3 Functional description centralized switching architecture: D-data partly processed decentralized (concept 2) 4-20
           4.6.2.4 Packet switching support 4-22
   4.7 Decentralized switching 4-25
       4.7.1 Introduction 4-25
       4.7.2 Two concepts 4-25
           4.7.2.1 Functional description decentralized switching architecture: all D-data via switch bus (concept 1) 4-25
           4.7.2.2 Packet switching support 4-33
           4.7.2.3 Functional description decentralized switching architecture: D-data partly processed decentralized (concept 2) 4-34
           4.7.2.4 Packet switching support 4-37
   4.8 Architecture review and conclusions 4-39
5. On the feasibility of a common D-channel 5-1
   5.1 Introduction 5-1
   5.2 Calculation of the acceptable number of terminals that can be serviced by one D-channel 5-1
   5.3 Software development aspects of a common D-channel 5-5
   5.4 Hardware development aspects of a common D-channel 5-8
   5.5 Conclusions 5-8
6. Implementation of the hardware designs establishing the intelligent NT's functional behaviour
   6.1 Introduction 6-1
   6.2 Implementation of the centralized switching design 6-1
      6.2.1 Introduction 6-1
      6.2.2 Description of the Mitel 8981D digital crosspoint switch [Ref. 8] 6-2
      6.2.3 Description of the architecture representing the centralized switching implementation 6-4
      6.2.4 Manufacturing aspects 6-10
      6.2.5 Price 6-11
   6.3 Implementation of the decentralized design 6-11
      6.3.1 The IST-bus structure 6-12
      6.3.2 The IST Bus Interface circuit (PCB2310) 6-14
      6.3.3 The IST bus applied to the decentralized switching architecture of the intelligent NT 6-15
   6.4 Conclusions 6-19

7. Conclusions and Recommendations 7-1
1. Introduction

The ISDN Access Systems Group of AT&T and Philips Telecommunications is a business unit that is focussing on the ISDN product market. The development activities of the group are concerned with products that provide access to the ISDN. Due to the high investments that are involved with ISDN product development, a pre-study often precedes the development of a particular product. This pre-study is useful for the scheduling of development investments (people and equipment) in a planning for the future. On the other hand, a pre-study may also be used to decide to abandon the development of the product because of a lack of technical or economical feasibility that has been demonstrated within the pre-study report. This report is part of a pre-study on a product that could be developed by IAS in the near future. It’s called the intelligent Network Termination (intelligent NT).

The CCITT has defined a reference configuration for the ISDN at the userside (Ref. [1]). This is shown in fig. 1-1.

IAS has implemented the functions established by some of these blocks into actual products or
product designs*. The intelligent NT is in fact the implementation of both NT1 and NT2 blocks. The functions performed by the NT1 block are stated by CCITT as follows:

NT1, network termination 1, includes functions broadly equivalent to layer 1 (physical) of the OSI reference model. These functions are associated with the proper physical and electromagnetic termination of the network. NT1 functions are:

- line transmission termination;
- layer 1 line maintenance functions and performance monitoring;
- timing;
- power transfer;
- layer 1 multiplexing;
- interface termination, including multidrop termination employing layer 1 contention resolution.

The NT2 functions stated by the CCITT are:

- layer 2 and 3 protocol handling;
- layer 2 and 3 multiplexing;
- switching;
- concentration;
- maintenance functions;
- interface termination and other layer 1 functions.

Examples of products including NT2 functions are PABXs and local area networks. They don't necessarily provide all NT2 functions stated above. It depends on the product which of the functions should be implemented. The pre-study of the intelligent NT should demonstrate which of the functions should be implemented and in what way.

As for the user of the intelligent NT some interesting features can be provided which aren't provided by the NT1.

- connection of a larger number of terminals
- internal communication possible without the intervention of a charging office

* The 'BITS': an ISDN terminal (TE1); for the USA market there is developed (and already being sold) an NT1 (using 2B1Q linecode) while a 'Euro NT' is developed for the CEPT countries (4B3T linecode); Furthermore, several TAs are being designed (for different protocol conversions)
• building up a small PABX is possible
• extra calling features are provided locally without the requirement for service subscription in the central office

The ability to establish local (i.e. within the NT's configuration) calls is quite an advantageous feature of the intelligent NT compared to the NT1*. For datacommunications a local area network can be established using packet switched B or D-channels.

The product can be applied in residential as well as business environments. The configuration size should therefore be adjustable to the customer's requirements/applications.

The intelligent NT’s pre-study project is split into two parts. One part, which is studied by Jaap de Wildt involves the software implementation for the NT in connection with the 5ESS™ (release 5E5) exchange.

The author of this report was concerned with the second part of the pre-study project: the hardware implementation of the intelligent NT.

In this report a functional description of the intelligent NT will be given in order to constitute a base upon which the study can be performed. In fact this is the first phase in the study process. The next step is the hardware design phase. During this phase different configurations are introduced which are built up out of functional blocks. The configurations will be discussed and using a design target list they will be valued upon their applicability for the NT.

The implementation phase is the last step to be taken. Using the design(s) that turned out to be most useful for the intelligent NT during the design phase, an actual implementation will be established. The functional blocks the design consisted of will be replaced by actual components. An important issue that is considered after implementation is a price estimation of the configuration that is established.

It should be remarked that the architectures that are created are not intended to be right-a-way implementable architectures. This is beyond the scope of this project. Only the functional blocks performing a major function within the design have been implemented.

The report is divided into chapters. The core of the report is represented by chapters 2 up to 6.

* The possibility to establish local calls without being charged is offered in the present Public Switched analogue Telephone Network too. The Dutch PTT offers a 'Klavervier' (nowadays called Homevox) configuration to its subscribers. This device enables the connection of 4 (internal separately addressable) telephone sets to a single subscriber line. The main features are: internal calling while simultaneously an external call is established, three-way-calling and call transfer.
Chapter 2 contains the product definition for the *intelligent NT*.

In chapter 3 the NT's hardware approach will be introduced by some information and analysis of systems like the *intelligent NT* and descriptions of interfaces and their compatibility and timing requirements.

Chapter 4 embodies the results of the design phase: the switching part of the *intelligent NT* is studied carefully and different designs are described.

Chapter 5 is a kind of intermezzo wherein the feasibility of a mechanism called 'common D-channel usage' is discussed. This mechanism could be applicable to the *intelligent NT*'s design and therefore is analyzed more carefully.

Finally, chapter 6 deals with the implementation of designs created in chapter 4.
2. Productdefinition intelligent NT

2.1 Introduction

A product definition specifies a product’s functional behaviour. It contains a for marketing oriented people readable/interesting description of a product and its applications. Instead of explaining how a product can perform certain behaviour, the product definition merely describes what a product can do and how it can be applied.

In a product’s design process, it’s very important to have a thorough base that defines the functions that should be implemented in the design of the product. So the product definition must serve the designers too.

For the intelligent NT the product definition embodies a description of telephony features and technical features. The telephony features are mainly a matter of software implementation supported by hardware that also defines the technical capabilities of the NT. Though the implementation of the telephony features by software isn’t part of this project, they yet have been mentioned here in order to offer a complete product definition which isn’t focussed on technical features only.

2.2 Calling-features

2.2.1 Feature overview

In order to obtain an overview what calling-features exist within the PABX-market and by which suppliers they are being offered, a small study has been performed. This study resulted in a large list of all kinds of services offered to the customer by a PABX or a terminal. It appeared that in some cases the network’s intelligence is mainly located in the PABX (so a simple subscriber-set is sufficient). But in other cases, the terminal (subscriber-set) contains a lot of intelligence which supports the network’s overall performance.

When studying the articles there has already been made a small selection. Sometimes some features definitely turned out to be implemented in the terminal and needed no special support by the network. These features have been left out when the table was constructed. Since this study is concerned with the product definition of an intelligent NT, terminal features are of minor importance.
2.2.2 Explanatory listing of the features

This paragraph contains short descriptions of the features listed in the table printed in the previous paragraph. These descriptions should provide the reader with enough information to understand what service is provided by a specific feature. This listing may help the reader when a feature is encountered which service isn’t clear to him/her.

**Abbreviated Dialling:**

The subscriber can specify a list containing directory numbers that are associated with a unique

*Italic printed features have some subfeatures listed in the next paragraph.

** 1 = 5ESS-CENTREX  2 = SEL System12 (a large switch)  3 = Hasler SL-1  4 = Philips TBX  5 = CCITT WP XI  6 = Ericsson MD110  7 = PTT "Klavervier"

***conference calling internal only.

---

**TABLE 2-1. Telephony features provided by several systems**

<table>
<thead>
<tr>
<th>Feature*</th>
<th>System**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Abbreviated dialling</td>
<td>X X X X X X</td>
</tr>
<tr>
<td>Aut. Answ. speech storing</td>
<td>X</td>
</tr>
<tr>
<td>Aut. Call Back</td>
<td>X X X X X</td>
</tr>
<tr>
<td>Call Forwarding</td>
<td>X X X X X X</td>
</tr>
<tr>
<td>Call Hold</td>
<td>X X X X X X</td>
</tr>
<tr>
<td>Calling Line Identification</td>
<td>X X X</td>
</tr>
<tr>
<td>Call Pickup</td>
<td>X X X X X</td>
</tr>
<tr>
<td>Call Transfer</td>
<td>X X X X X X X</td>
</tr>
<tr>
<td>Call Waiting</td>
<td>X X X X X</td>
</tr>
<tr>
<td>Conference Calling</td>
<td>X X X X X X X X***</td>
</tr>
<tr>
<td>Dial Access to Priv. Facilities</td>
<td>X</td>
</tr>
<tr>
<td>Distinctive Ringing Tone</td>
<td>X</td>
</tr>
<tr>
<td>Executive/Secretary Comb.</td>
<td>X</td>
</tr>
<tr>
<td>Group Numbering Plan</td>
<td>X X X X X X</td>
</tr>
<tr>
<td>Multiline Hunting</td>
<td>X X X X X X</td>
</tr>
<tr>
<td>Queuing for lines</td>
<td>X</td>
</tr>
<tr>
<td>Semirestricted Station</td>
<td>X</td>
</tr>
<tr>
<td>Three-way Calling</td>
<td>X X X X X X X</td>
</tr>
<tr>
<td>Traffic Restrictions</td>
<td>X X X</td>
</tr>
<tr>
<td>Uniform Call distrib.</td>
<td>X</td>
</tr>
</tbody>
</table>
abbreviated dialling code.

**Automatic Answering using speech storing:**

When the terminal is set into this mode, an incoming call will be answered automatically after a definite number of rings. The message has been recorded and digitally stored by the terminal user earlier. When the message has been told, there exists a possibility for the caller to drop a message into a (voice) mail-box.

**Automatic Call Back (ACB):**

This facility enables a subscriber encountering a busy destination, to receive automatically a call-back when the destination is idle again. This is achieved by dialling a code when the 'destination-busy'-tones are being received. The ACB-procedure will be initiated as soon as the (earlier dialled) destination will be idle. If the destination is busy again when the procedure has been initiated, the originating extension will receive busy tone but the ACB facility will remain active. When the call has been set-up or when the ringing has been timed-out, the ACB-procedure will be terminated.

**Call Forwarding:**

When this facility has been invoked by dialling the Call Forwarding-code and the destination’s address, calls to a particular address will be re-routed to the dialled secondary destination (eventually depending on conditions like 'Busy-line' or 'No Answer'). On the secondary’s display the originating address (when the 'calling line identification'-feature is active) and a 'Call Forwarding'-sign will be shown in case of a forwarded call.

- all calls
- incoming calls only
- within group only: the directory number dialled during the Call Forw. activation process will be modified into intragroup extension numbers

**Busy Line**

Calls attempting to terminate to a busy line, will be redirected to another specified destination (secondary terminal)

- all calls
- incoming calls only: forwarding restricted for incoming calls

**No Answer**

The idle destination will be rung during a specified number of seconds and if the call isn’t answered, it will be re-routed to another terminal.

- all calls
- incoming calls only: forwarding restricted for incoming calls

**Call Hold:**
Allows a subscriber to put on hold any in-progress call. When the call is put on hold, the subscriber can establish another call. The established calls can not be connected. When the terminal user disconnects when there's a party on-hold, he is automatically rung back and, upon answer, connected again to the held party.

**Calling Line identification:**

If not restricted, a caller's ID is shown when he's calling a terminal. This is possible in both directions.

**Call Pickup:**

Enables a terminal to answer a call, routed to another terminal within a defined pickup group. Call pickup is established by means of dialling a pickup code. When more terminals are being rung, the one which has been ringing longest is picked up first.

- directed: differs from ordinary call pickup in that there's no defined pickup group. The user dials a pickup code and a certain terminal address (assumed that terminal allows call pickup; otherwise a tone sequence is generated). When that (dialed) terminal is being rung, the user is able to pickup that call. If the terminal is busy, call is established after a three-way barge-in
- directed without barge-in: a three-way call can not be established

**Call Transfer**: 

Allows a terminal user to transfer calls to an individual line, to another terminal. This transfer is made by hanging up (withdrawal) after invoking the three-way calling feature.

- all calls
- incoming calls only
- internal calls only

**Call Waiting:**

A terminal connected to another terminal is being alerted by a call-waiting tone when another terminal is trying to establish a connection with that terminal. The hold-on, three-way and conference telephony features are available.

- incoming only
- originating: allows a subscriber to assign call waiting service to a called party for the duration of a call (restricted to interoffice calls)

* See also: three-way calling
cancel: disables a terminal's call-waiting feature for one call

Conference Calling:

By adding up to five other parties a terminal can form a six-way call. Adding parties is done by means of the hold-on feature. Initiating the conference mode is done by means of dialling an access code.

Dial access to private facilities:

By dialling a special access code, a terminal can be connected to a private facility like a foreign exchange trunk.

Distinctive Ringing Tone:

Different ringing patterns can be generated for internal and external calls.

Executive/Secretary combination:

This duo consists of two normal extensions. Additionally they can reach each other by abbreviated dialling and call diversion is allowed (secretary or executive dials 'present' or 'absent' code. Calls arriving at an 'absent' terminal will be automatically diverted to the 'present' terminal.

Group Numbering Plan:

Within the NT controlled network, groups can be determined. These groups may have their own numbering plan (like abbreviated dialling).

Multiline Hunting:

This feature performs a search to an idle terminal when a particular terminal is busy. There exists a user-specified list which is called a hunting group. This list contains those lines which initiate hunting when they are called in busy state. It also contains those terminals which should be hunted during the hunting process.

  • circular: all the lines within the hunting group can initiate a hunting process when they are busy

Queueing for lines:

If all lines within a hunting group are busy, a specified number of waiting calls can be queued. The calls on the queue can be given a distinctive tone or a message.

Semirestricted Station:

This feature provides the ability to restrict a terminal from having access to particular facilities and/or being reached by certain kinds of origins.
Three-way Calling:

Allows a terminal in the talking state to add a third party to the call. Adding is done using the hold-on key.

Traffic Restrictions:

Certain terminals are not allowed to complete calls to particular destinations specified by system controller. Changing a terminal's code can be done by authorized terminals.

Uniform Call Distribution:

In order to reach a uniform distribution characteristic, incoming calls to a group are automatically distributed to the terminals after having checked the number of incoming calls serviced by each terminal. It's a selective hunting arrangement.

2.2.3 Classification of the features

After having listed the most commonly provided (PABX-)services, they will be ordered using classes. However, before the items will be classified, the use of classifying and the selection of the classes will be explained to the reader.

2.2.3.1 Why classification?

As already told before, the product definition serves as a base upon which the decision of "marketing the product or not" is to be taken. Another conclusion on the product definition could be that it needs slight changes. Therefore, it should be structurized and flexible so it can be changed easily.

Implementing all the features mentioned before, into the intelligent NT, isn't conform the goals which are set before designing this product. It would lead towards the re-development of a PABX (which isn't even conform the APT-targets). So, since only a small featureset will be implemented within the intelligent NT, it definitely has to be a well-selected one. When selecting the set attention must be paid to e.g. the product's applications, the price and the size. These are a few criteria which take part in the decision of "implementing a feature or not".

In order to structurize this selection-process, a list will be generated where the features are classified. By means of defining several classes, the features' performance/character is described. Increasing the number of different classes, results in a refinement of the characterization and enables a more precise selection of the featureset. The items, upon which the classification is based have to be chosen in such a way that they support the requirements for making the decision of choosing the feature or not. For example: When a product's target is to cover usermarket ALPHA, a good feature-classification item could be the 'application area'. A class of features having their application area in common, can be distinguished then very easily.
2.2.3.2 The selection of the classification items

The first step in the featureset composition process is the selection of the classes. The choice of these classes determines the level of characterization of the features and therefore plays an important role when deciding whether a feature will be part of the featureset or not. So taking a look at the initially stated targets of the intelligent NT is useful before choosing the classification items. In chapter 1 a description of the product’s targets was given. These targets are:

• internal calls possible
• providing a (for the aimed user group) attractive featureset (esp. for internal calls)
• usage: in-house, small businesses
• limited size (4 cubic dm)
• flexible structure (as number and type of terminals and number of outgoing lines are concerned)

The classes which can be used now for the featureset composition are:

[1] Occurrence frequency amongst services/suppliers
[3] Used for internal calls often
[4] Besides business-applications, residential applications too
[5] Provided by public switch (5ESS-PRX) as supplementary service

2.2.3.3 Explanation of the classification items

Item 1.:

This item has a rather statistical character. It provides the reader with an indication of the popularity of a certain feature amongst those services and suppliers mentioned in the footer of Table 2-1. It won’t be the primary decision-item when selecting the featureset because the services (merely those of PABXs) aren’t really competitive with the intelligent NT. So it’s useful to take a look at this class but a decision will probably not be dependent of this class.

Item 2.:

This class is chosen to show whether a feature requires terminals being grouped. This means that terminals are being known by the switch as a group. This is a useful feature in big offices
where several departments can be distinguished. For the application area of the *intelligent NT* (small businesses and residences) this group character isn’t very feasible. Therefore, features belonging to this class won’t be selected for participation in the featureset.

**Item 3.:**

The processing of internal calls is one of the main objectives of the *intelligent NT*. The cost-effectiveness of the *intelligent NT* compared to networks supported by conventional NTs is high, only when the number of internal calls is high. Basically, the *intelligent NT* should be as cheap as possible and thus hard- and software investments should be as low as possible. Nevertheless, in order to make the product (besides the financial aspect) attractive to the customer and prepare it for competitiveness with products like the *intelligent NT* developed by other companies, a small featureset will be provided. This featureset will be focussed on internal use.

**Item 4.:**

Since the *intelligent NT* isn’t destined for business use only but also for residential use, it may be quite clear that it’s useful to check whether a feature is also feasible for latter. It should be prevented that the intelligent NT becomes a product which has a typical office-character.

**Item 5.:**

This item is strongly linked to item 3. Features which aren’t often used for internal calls will probably not be supported by the *intelligent NT* for reasons explained before (see 'item 3'). When those features are supported as supplementary services by the network (the customer has to pay extra for these services), there’s no doubt about it: they need not to be supported by the *intelligent NT*.

2.2.3.4 The classified featuretable
<table>
<thead>
<tr>
<th>Feature</th>
<th>Classification-item*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>Abbreviated Dialling</td>
<td>6  X  X  X</td>
</tr>
<tr>
<td>Aut. Answ. speech storing</td>
<td>1  X</td>
</tr>
<tr>
<td>Aut. Call Back</td>
<td>6  X  X</td>
</tr>
<tr>
<td>Call Forwarding</td>
<td>7  **  X</td>
</tr>
<tr>
<td>Call Hold</td>
<td>8  X  X  X</td>
</tr>
<tr>
<td>Calling Line identification</td>
<td>4  X</td>
</tr>
<tr>
<td>Call Pickup</td>
<td>5  X  X</td>
</tr>
<tr>
<td>Call Transfer</td>
<td>6  X  X</td>
</tr>
<tr>
<td>Call Waiting</td>
<td>7  X  X</td>
</tr>
<tr>
<td>Conference Calling</td>
<td>8</td>
</tr>
<tr>
<td>Dial Access to Priv. Facilities</td>
<td>2</td>
</tr>
<tr>
<td>Distinctive Ringing Tone</td>
<td>2  X  X</td>
</tr>
<tr>
<td>Executive/Secretary Comb.</td>
<td>1  X</td>
</tr>
<tr>
<td>Group Numbering Plan</td>
<td>3  X</td>
</tr>
<tr>
<td>Multiline Hunting</td>
<td>4  X</td>
</tr>
<tr>
<td>Queuing for lines</td>
<td>1</td>
</tr>
<tr>
<td>Semirestricted Station</td>
<td>1</td>
</tr>
<tr>
<td>Three-way Calling</td>
<td>6  X  X  X</td>
</tr>
<tr>
<td>Traffic Restrictions</td>
<td>4</td>
</tr>
<tr>
<td>Uniform Call distrib.</td>
<td>1</td>
</tr>
</tbody>
</table>

**TABLE 2-2. Classified featurelist**

### 2.2.4 How the features might be implemented

In this paragraph some considerations are written down which indicate how a feature might be implemented and what (hardware) investments it might take. This paragraph supports the selection of the featureset.

**Abbreviated dialling:**

This feature can be implemented within the terminal. It can merely be performed by storing a directory list which links the abbreviation code (e.g. `<function key> + <digit>`) and the

---

* 1 = Occurrence frequency amongst services/suppliers (see Table 2-1) 2 = group oriented 3 = used for internal calls 4 = besides business-applications, residential-applications too 5 = provided by local switch (e.g. 5ESS-PRX) as supplementary service

** In residential situations usually several terminals with the same address (dialling access number) so not very useful
subscriber number.

**Aut. Answering speech storing:**

Answering machines in the non-ISDN era make use of magnetic tape to store a message. Since ISDN-terminals digitize speech, we could easily use the digitized speech for abovementioned purpose. In case of the intelligent NT, this could be arranged as following: a subscriber who wants to store a message invokes the answering service routine via the D-channel. A B-channel will be reserved which will be routed to a memory in the *intelligent NT*. Together with the message the subscriber message will be stored within the memory. The answering controller will be warned by the terminal that a particular message has to be transmitted when a particular subscriber number (within the intelligent NT-TE configuration) will be called.

**Aut. Call Back:**

This feature can be invoked when a busy destination is encountered. A procedure has to be started which checks (over the D-channel) whether the calling state of the destination has been changed into idle state. When the destination becomes idle, the procedure automatically alerts the party which has invoked the feature and when this party answers by picking up the hook, the procedure starts alerting the destination. The procedure should be invoked again when the destination turns out to be busy again (due to the fact that a call has been arrived or is set up by that particular destination in the mean time). The feature is mainly based upon software implementation assisted by a hardware D-channel processor.

**Call Forwarding:**

This feature can be divided into several processes. First the Call Forwarding invocation. This process is initiated by the terminal user by means of D-channel signaling. The destination and the condition upon which the 'Call Forwarding'-decision is taken have to be set/stored in a code-list. The second process takes care of the actual check whether an arriving call should be forwarded or not. This is a D-channel address check which is partly software and partly hardware supported (D-channel controller). The last process performs the re-routing by adding info to the D-channel (e.g. 'call is forwarded' and changing the destination's address -layer 3-).

**Call Hold:**

By pressing a 'Hold'-key on the TE, the user can put a call on hold. In order to be independent of the local exchange, the hold feature should be performed by the *intelligent NT*. The call will be re-routed to the *intelligent NT*. So at the S-side of the *intelligent NT* a B-channel is released for the time the call is put on hold. When the local exchange's hold-feature is being used, the NT1's B-channel can be released from U-interface up to local exchange. In order to retrieve the put-on-hold call again, the *intelligent NT* must be capable of setting up the link again (layer 3 capabilities). In order to retrieve lost calls (the terminal user disconnects while a call is put on hold) the *intelligent NT* should (possibly) be able of setting up a call automatically. This is also dependent of the terminal which is connected.
Calling Line Identification:

This feature requires no extra implementations as those required for setting up a call.

Call Pickup:

This feature uses a kind of database which contains the terminals which belong to a defined pickup group. This database has to be stored in the intelligent NT. Above that, a monitoring program is required to store these grouping configuration within the NT's database. When a D-channel interpreter is informed by a TE that it wants to pick up a call which is alerting at another terminal, there has to be checked whether both the TEs belong to the same pickup group. If so the call has to be re-routed to the terminal requesting a call pickup.

Call Transfer:

This feature must be able to change a three-way call into a regular call (so a call which doesn't use the three-way call hardware). The party which wants to transfer a call may take part of the conversation which is to be altered into a regular two-party call. However, it should also be possible to transfer a call without having established a three-way call. This means that the user should be able to transfer individually a call to another terminal. This might be established by parking a call which is labeled and retrieving again by using eg. a retrieve + parkcode. In summary, this feature is mainly concerned with (re)-routing.

Call Waiting:

This feature is only applicable to those terminals that possess several (more than 2) Call Appearances. This means that if a B-channel at a terminal is in use, the user is being alerted by means of a tone generated by the NT, that another TE is attempting to establish a call to his TE. A tone generator has to be implemented to support this feature.

Conference Calling:

As hardware is concerned, this feature requires switching manipulations. Ordinary timeslot interchanges can't establish N-way calls (that establish a conference call): only one timeslot can be used per B-channel so it must be arranged that all information to be sent to one destination is transmitted in one timeslot of the Multiplexing frame.

2.2.5 Selection of the features

Now a structurized overview is obtained, the featureset can be selected. This selection is based upon the results of the feature gaining and classification phase. The last paragraph which gives a description of how the features might be implemented, takes part in the decision process as well though it's merely a global overview. As a result a change of the set may happen due to
the fact that its implementation may be too expensive (development costs too high).

In order to continue structurized selecting, the telephony features will be checked one-by-one using table 2-2. This check results into the conclusion whether a calling feature will be implemented or not with additional information.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Implemented</th>
<th>Additional info.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviated Dialling</td>
<td></td>
<td>Will probably be implemented in most ISDN terminals.</td>
</tr>
<tr>
<td>Aut. Answ. speech storing</td>
<td></td>
<td>Not useful for internal calls. Implementing this feature within the NT will probably be too expensive.</td>
</tr>
<tr>
<td>Aut. Call Back</td>
<td></td>
<td>Requires too much software investments.</td>
</tr>
<tr>
<td>Call Forwarding</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Call Hold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calling Line identification</td>
<td>X</td>
<td>Nice and sometimes very useful feature. It might not be necessary to implement this feature within the NT since the TE could support this feature on its own.</td>
</tr>
<tr>
<td>Call Pickup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call Transfer</td>
<td>X</td>
<td>Very comfortable and useful.</td>
</tr>
<tr>
<td>Call Waiting</td>
<td>X</td>
<td>For incoming calls very useful.</td>
</tr>
<tr>
<td>Conference Calling</td>
<td></td>
<td>Implementing three-way calling for internal (and external) calls is enough. Conference calling is used more often for external calls and can be supported by the local switch.</td>
</tr>
<tr>
<td>Dial Access to Priv. Facilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distinctive Ringing Tones</td>
<td></td>
<td>This feature isn’t necessary assuming that address info (calling line identification) can be shown on the terminal display.</td>
</tr>
<tr>
<td>Executive/Secretary Comb.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Numbering Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiline Hunting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queuing for lines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semirestricted Station</td>
<td></td>
<td>Should eventually be implemented in terminal.</td>
</tr>
<tr>
<td>Three-way Calling</td>
<td>X</td>
<td>Both for internal and externals calls.</td>
</tr>
<tr>
<td>Traffic Restrictions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniform Call distrib.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 2-3. Selection table telephony features

So the marked features in Table 2-3 should be provided by the intelligent NT. It has been selected carefully merely based upon the usefulness for internal calls and the application area. Above that, the product definer has the human and material resources which he estimates to be
needed for the development of the features, in mind. As already told before, it may turn out
during the hardware development phase, that a feature is too expensive or that another feature
can easily be implemented without large extra investments. Accordingly the product definition
will be adjusted then.

2.3 Technical features

2.3.1 Introduction

After having specified the telephony features that are supported by the intelligent NT, the
technical capabilities of the product are specified now. The interfaces, terminals and switching
capabilities will be described. Detailed information on the interfaces has been omitted but can
be found in the CCITT Recommendations (Ref. [1]). Furthermore some information will be
given on the system configuration.

2.3.2 Interfaces

The intelligent NT provides different interfaces. Towards the (ISD) Network the NT provides
a U-interface. This function is established by a U-transceiver. It transmits and receives
dataframes over the DSL that is in fact the gateway to the ISDN (or to the exchange). Dependent on the country the intelligent NT is developed for, the linecode used is 2B1Q
(North-America) or 4B3T (Europe).

Towards the user, the intelligent NT offers an S-interface. This interface enables the user to
connect ISDN terminals. As for the way these terminals can be connected, different
configurations are possible which are shown below:
Figure 2-1. Terminal connection configurations to S-interface

2.3.3 Terminals

The intelligent NT provides interfaces to which ISDN terminals can be connected. The terminals don’t differ from the terminals that are connected to NT1’s. The terminals communicate via circuit switched B-channels while the D-channel is used for signalling data transport. Examples of these terminals are voice terminals and fax (group IV). It will be shown in par. 2.3.4. that up to two terminals using circuit switched B-channels can be connected to each S-bus (in order to avoid blocking at the bus).

Optionally the intelligent NT can establish a LAN. In this case, terminals can be connected that communicate packet switched via the D-channel with a HOST-computer that is connected to the NT as well. The number of packet data terminals that can be connected to an S-bus circuit is dependent on the traffic load this terminal causes on the D-channel. This item is studied in paragraph 5.2.

Due to electrical requirements at the bus, the number of terminals (no matter what type) must never exceed 8.

Terminals can be moved from one location to another; it doesn’t matter whether it’s moved at the same S-bus or to another S-bus. This portability of a terminal is possible if no call is
established at the terminal but also if there is established one.

2.3.4 Circuit switched call routing configurations

The different types of calls (using a circuit switched B-channel) the intelligent NT is able to establish are described in this paragraph.

External call

An external call is a call between a terminal connected to the intelligent NT and the ISDN. The call is established using a B-channel that is available at both S and U-interfaces. The switching capability of the NT makes call routing between S and U-interfaces v.v. possible. This is shown below.

![Figure 2-2. External call routing](image)

This type of call requires:

- free B-channel at S-interface
- free B-channel at U-interface
- free switch path

Internal call between terminals connected to the same S-bus circuit

Terminals connected to an S-bus circuit in the NT1 situation can't communicate. The intelligent NT does provide this possibility by its switching function. The call routing with respect to B-channels is shown below.
Figure 2-3. Internal call routing between terminals connected to the same S-bus

As shown, this type of call requires 2B channels at a single interface which disables any other call-establishment at this interface. So if a third TE (using a B-channel for communication) is connected to this S-bus, it can't establish a call since no B-channels are available at that time. In that case there is blocking at the S-bus. Connecting up to two terminals (that use circuit switched B-channels) to the S-bus causes the blocking probability to be 'zero' at the S-bus.

Internal call between terminals connected to different S-bus circuits

This situation is shown below.

Figure 2-4. Internal calls between terminals connected to different S-bus circuits

This call is like the previous described except for the B-channel usage: per S-transceiver a single B-channel is occupied here. There remains a B-channel for any other terminal at the S-bus. Similar to the previous situation blocking can be avoided by connecting (only) up to two terminals to the S-bus circuit.
2.3.5 Packet switching option

The intelligent NT offers packet switching via the D-channel as an option. Dataterminals are transceiving datapackets via the 16 kbit/s D-channel for communication with a HOST. The intelligent NT routes incoming packets from the dataterminals to the HOST by multiplexing (on layer-2) 16 kbit/s channels into a 64 kbit/s channel v.v.

The protocol used for transmission over the serial D-channel is HDLC. Due to the fact that there's no switching capability required in the NT there are no strict limitations on the number of packet terminals that can be connected to the same S-bus (except for the electrical requirements: maximum 8 TEs). The D-channel however only can handle a 16 kbit/s datastream. Using HDLC frames for conveying the data packets, the maximum baudrate of terminals transceiving data in the D-channel is 9.6 kbaud. Due to the fact that this kind of terminals doesn't send data continuously, there can be connected more dataterminals to the same D-channel with a slight chance on collision*.

2.3.6 The basic NT configuration

The intelligent NT should be developed being a modular system. It should be possible to expand the system by means of applying units that enlarge the number of gateways to the ISDN (U-interfaces) or the number of terminals that can be connected (S-interfaces). Above that, adding the (optional) packet switching capability should also be easy e.g. by installing an extra Eurocard to the system.

A basic configuration will be defined now that will be used during the hardware design phase and which could be presented as a basic configuration to the customer. The basic configuration used here consists of 4 S and 2 U-interface circuits. The reasons to configure the basic intelligent NT like this are mentioned here:

- 2 U-interfaces: the decision to implement two gateways to the ISDN has merely been taken for technical reasons; there occurs an interesting timing problem as soon as two or more U-interface circuits are involved since they derive timing from different DSLs. This problem doesn't exist for a single U-interface configuration. Above that, though it might be a too large configuration for residential applications, for small businesses 2 DSLs (4 B-channels) are quite appropriate. Due to the modular system building strategy that is used for the development of the intelligent NT's hardware, the basic NT can also be adjusted for single DSL connections (and thus applying a single U-interface without having the necessity to add timing controlling hardware).

* for more detailed information on the data terminal type and traffic statistics: see par. 5.2.
4 S-interfaces: the number of S-interfaces determines the number of terminals that can be connected to the intelligent NT. It has been illustrated in par. 2.3.2 that different configurations are possible on the S-interface (point to point, point to multipoint) and if notion is taken of blocking probabilities (at the S-bus), a maximum of two terminals in a point to multipoint connection is mostly recommendable. In order to configure the basic intelligent NT in such a way that the switching system's blocking probability for circuit switched calls, caused by the non-availability of B-channels, is kept to a minimum, there has been looked at the assumptions that are made for PABX design*. These designs are based upon 50% internal and 50% external call traffic ratio at the terminals. Based on the 2 U-interface circuits implemented in the basic intelligent NT and providing the external call capability, 4 S-interface circuits are implemented providing enough B-channel capacity to establish the 50%-50% internal-external call ratio. Some of the possible call configurations are shown below, based upon the assumption that there are two terminals (using circuit switched B-channels) connected to each S-bus circuit.

---

* There has been made use of (confidential) information provided by PTDSN.
S1 & S2: all terminals (4) connected to these busses are involved with internal calls

S3 & S4: 2 external calls per S-bus which implies 4 terminals being used for external calls

S1, S2 & S4: 4 terminals involved with internal calls
S1, S3 & S4: 4 terminals involved with external calls

Both situations show, that there's no blocking caused by the switching in the basic intelligent NT and that 50% of the terminals are involved with internal calls while 50% is involved with external calls. All channels available at both S and U side are in use and this full coverage of channels proves the non-blocking character of the switch.

2.3.7 System configuration

The intelligent NT is built up out of functional modules that can be implemented on Eurocards. These are printed circuit boards sized 100mm x 160mm and equipped with a standard backplane connector. Using these cards for implementation of the hardware of the intelligent NT, a system is established that consists of several cards that are configured in a rack system and interconnected by means of a busstructure at the backplane. The housing's size should be limited to 4 cubic dm.
2.3.8 Expansion

Due to the modularity of the switching system it's possible to expand the intelligent NT's size gradually. The number of both U and S-interface circuits can be adjusted in order to be able to customize the product. A small PABX configuration can thus be established by adding hardware modules.
3. Pre-study for the hardware design of the intelligent NT

3.1 Introduction

In order to get acquainted with concepts and ideas for the intelligent NT some study is performed. This chapter contains a collection of those items that are valuable for the design of the NT's hardware architecture.

The starting point for this study is a group of concepts that resembles an intelligent NT system from functional point of view. The implementations of these concepts are being described and items which are of special importance for the hardware design will be given special attention.

A subject that is picked out for thorough study is interfacing. Especially implementations of transceiver interfaces like K and IOM are studied since the main part of the intelligent NT is located at this reference point.

System timing also requires special attention. Due to the fact that several U-transceivers are concerned, the situation occurs that several timebases are present (or if all powered down, no timebase is provided by the network at all). A master clock signal should be provided locally which is adjusted to the network timing (if present). A problem analysis for the timing is given in this chapter.

This chapter should be considered as a summary of the introductionary work to the hardware design performed by the author and furthermore as a problem detecting study. Those items which are possible problem sources for the design are being detected and analyzed to provide a base upon which the hardware design can be initiated.

3.2 Study on intelligent NT resembling systems

3.2.1 Introduction

This paragraph contains a description of systems that resemble the intelligent NT. There's been made use of descriptions supplied by the system's manufacturer which often are (still) in a preliminary state. The author had to face rather vague descriptions which made it seem that designers of such systems have ignored basic problems only for the sake of presenting them in an early stage of the ISDN equipment development process.
In a kind of review there has been tried to determine and analyze abovementioned items in order to be well prepared for the hardware design phase.

3.2.2 Alcatel's intelligent NT (Ref.[3])

Bell Telephone Manufacturing (BTM, a subsidiary of Alcatel N.V.) has developed an intelligent NT as shown in fig. 3.1.

For the design, BTM has stated 3 main objectives.

1. The model should be built in accordance with the functional diagram of the end user installation proposed by the CCITT. The intelligent NT is considered to embody both NT1 and NT2 functions (so it can be called NT12).

2. The design should enable study on the possibilities of using the NT12 as a cost effective alternative to a LAN (low data rates: < 64 kbit/s, few users: < 32 telephones).

3. Both circuit and packet switching should be supported by the NT12. Packet switching connections using the X.25 protocol on both B and D channels should also be supported.

Based upon abovementioned considerations, BTM has decided upon a blockstructure design as printed below.

![Blockstructure Diagram](image)

**Figure 3-1. Alcatel's intelligent NT blockstructure**

A short description of the modules will follow now.

- Switch and Tones Module
Switching is performed by a timeslot interchanger connected to a switch bus providing 96 timeslots of 64 kbit/s each. Each user has access to three of these channels (2B + D; D-data is also transported on 64 kbit/s channels). The number of user interfaces is restricted to 16 which leaves 48 channels free to be allocated to interfaces with other NT12s and/or the public network. The tone generator produces different tones controlled by software and it's included for use with non-ISDN terminals.

- **S-interface Module**

This module provides the user with a CCITT 1.430 recommended S-interface. It performs all according OSI layer-1 functions. Above that, status information is exchanged between this interface and the processor module to control these layer-1 functions. Up to 16 S-modules can be equipped by the NT12 system.

- **Link Layer Module**

This module supports low layer-2 functions. To this end, it's equipped with a 32 channel HDLC which interfaces to the other NT12's modules via the switch bus allowing packetized data to be handled on either B or D-channels. Towards the processor module, all channels interface directly with the system processor memory.

- **U-interface module**

This module provides the connection to the public environment according to the U reference definition. The module uses three channels on the switch bus.

- **Processor module**

This module provides the NT12's intelligence. It consists of a single board processor (from 8088 family) containing all required memory and interfaces to the link controller and time switch. The status bits controlling the hardware can also be accessed via this interface (maintenance, loop establishment etc).

3.2.3 Siemens patent application (Ref. [4])

The Siemens company has designed an *intelligent NT*-like system. It provides access to the public network via a single U-interface. Towards the user there can be configured several S-interface modules which allows connection of several TEs. The remarkable item that identifies this configuration and which patent has been requested for is the common D-channel solution towards the userside. This concept will be discussed here. The configuration is shown in the figure printed below.
A switching device takes care of B-channel switching between S-interfaces or between S and U interfaces. The switch is controlled by the switch control unit (a microcontrolling system) that retrieves its switch-setting info from the data provided by the D-channels.

As D-channel processing is concerned, Siemens makes use of a system in which (towards user side) only a single logical D-channel can be determined. This is established by 'and'-ing (& module) the demuxed D-data lines and leading the and-ed data to the processor module.

In order to prevent channel collision, the D-data access control mechanism as it's known on the S-bus (Ref. [1], I.430 par.6.1), is applied here. To support this mechanism, the and-ed D-channel data is echoed to all transceivers which enables them to monitor the common D-channel.

Since B-channel switching is concerned between S- or S and U interfaces, it's necessary for the switch control to have knowledge of the physical location of the terminals. By and-ing the single S-transceivers, this information can't be gained anymore. Therefore, Siemens uses an S-ID register (S-identification) which registers which D-line (or: at which S-transceiver) the terminal is located that causes activity at the D-channel. This identification enables the processor to link logical TE identification at layer-2 (TEi) and physical location (S-transceiver). The register is write-enabled by the output signal of the & module which corresponds with the physical layer.
state "activity detected at the D-channel".

The D-data originating from the network can’t be handled as described hereabove since the D-channel access mechanism isn’t defined at this interface.

3.2.4 The star NT

The star NT is a concept that resembles only slightly the intelligent NT system described in the paragraph before. The most important difference is that it doesn’t support B-channel switching. Though skipping a description of this concept from this report therefore might seem most reasonable, it yet has been inserted because parts of this concept might be implemented within the intelligent NT architecture. The configuration is shown in fig. 3.3.

![Star NT configuration diagram]

Notes:
- data down: 2B+D+layer1 control to S transce.
- data up: 2B+D+ layer1 control to U transce.

Figure 3.3. Star NT configuration

The most striking point in aboveprinted figure is that datalines provided by S and U transceivers at their component interfaces (i.e. between S and U reference; for APT this interface is called the K interface) can simply be (wired)and-ed. Physically this is just a matter of connecting data up- and downstream pins at the transceiver chips.

The data originating from separate S-transceivers is overwritten due to their simultaneous access. The definition of the component interfaces at layer-1 allows this overwrite.

As for the D-channel data, this access is controlled by the access mechanism (Ref. [1], I.430 par.6.1) which is known for TEs at the same S-bus. This mechanism has been extended to the level where several transceivers (instead of TEs) are using the same D-channel (see also Siemens patent). In order to provide the transceivers with info about the common D-channel,
this D-data is echoed towards all S-transceivers by means of connecting the E-pins (echo bit provided by the S-transceivers towards the TEs connected to them) of the S-transceivers and thus establishing a common E-channel.

For the B-channels there's no need for such a mechanism as known for the D-channel since they are not accessed simultaneously. Above that, they are not used/occupied at random (as the D-channel is) because a call-setup procedure has been performed over the D-channel before a B-channel is being used.

There's neither need for a S-identifications procedure as used in the Siemens concept since there is no B-channel switching. Due to the fact that messages over the D-channel are being broadcast, each terminal can be reached.

3.2.5 Evaluating the systems

The Alcatel/BTM design is an intelligent NT -like system with only one access port (U-interface) towards the public network. If the user wants to expand his system, he can append a second NT module or even more. In order to create a kind of local network consisting of these NTs, there are timeslots at the switch bus available for communication between these NTs. Basically, the intelligent NT design by BTM is quite reasonable but as the expansion is concerned towards a multiple U-interface system, the article is rather vague. If two NTs want to communicate, using the (same) switch bus, their timing should be synchronous. Otherwise, communication modules should be applied (like UART). This aspect isn't mentioned at all in the article.

Compared to the Siemens concept, the Alcatel design doesn't use a common D-channel which might raise the costs due to a higher component (i.e. HDLC) use.

Just like the Alcatel configuration, the Siemens concept also has a single U-transceiver. The article doesn't insinuate the application of a second U-transceiver at all! Therefore it is only partly conform the intelligent NT specification stated in this report (multiple access ports towards network possible).

As for the S-identification system applied in the Siemens concept, the following can be remarked: the article ignores a situation in which two TEs, connected to different S-transceivers, access simultaneously their D-channel (and thus the common D-channel). So two registers are written at the same time. Resultingly, there can't be made a decision by the D-channel processor which TE eventually will retain access to the D-channel. Concluding: the article is quite vague about the protocol used for S-transceiver identification which makes it difficult to give a good description of e.g. the size of the register which could be useful during the hardware design phase.

The star-NT isn't an intelligent NT -like system since it has no B-channel switching at its disposal. But the concept in which and-ing the 2B+D-layer1 up- and downstream datalines is enabled, may be interesting during the intelligent NT's design phase since it's supported by the recently redesigned APT S-chip (STT3).
3.3 Component dependent interfaces between the S and U reference points

3.3.1 Introduction

Though manufacturers are striving after standardization e.g. by taking part in CCITT study groups, there are subjects which haven’t been standardized. The interface localized between U and S reference points is such an item (see fig. 3.4).

![Diagram of component dependent interface between S and U reference points](image)

**Figure 3-4.** Component dependent interface between S and U reference points

Four telecommunication companies* in Europe recently have agreed upon the definition and acknowledgement of the IOM interface which can be considered as a step towards standardization of this interface.

However, nowadays one has to cope with the cold fact that a variety of component interface types exists. For the sake of having knowledge of these different interfaces and thus being able to check interworking performance of components meeting different interface definitions, this paragraph is inserted.

3.3.2 The K(2)-interface

This type of interface is provided by Philips/APT U and S chips. The K2 interface is an

* These are Siemens, Italtel, GTE/Plessey and Alcatel.
revised version of the K interface. The layer-1 control/maintenance features have been extended by increasing the number of control octets conveyed by the data frame. To this end the bitrate has been increased. The exact description of the K2 interface will be given after description of the K-interface.

The K-interface is physically represented by 4 wires. Their function and direction (related to U and S chips) is also described.

<table>
<thead>
<tr>
<th>Function</th>
<th>Direction (in NT1 configuration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameclock (8 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>bitclock (256 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>data upstream (256 kbit/s)</td>
<td>S → U</td>
</tr>
<tr>
<td>data downstream (256 kbit/s)</td>
<td>U → S</td>
</tr>
</tbody>
</table>

*Figure 3-5. K-interface physical description*

The data configuration with the according clock signals on the serial datalines is like shown in fig. 3.6.

*Figure 3-6. Frame and clock description K-interface*

The D-octet contains only 2 bits D-data; the remaining six bits are for layer-1 control/maintenance. The S-octet contains also bits for layer-1 control and maintenance. B1 and B2 contain B-channel data bits. B and D data is conveyed transparently over the K-interface to S and U interfaces.

The Data clock signal is equal to the data speed. So each clock period corresponds with a bit location within the dataframe.

The K2-interface has a physical configuration equal to the K-interface. But the frequency of the data clock has been increased in order to increase the data rate at the DD and DU lines (see fig. 3.7; DD is data downstream which is from U chip to S chip -in the NT case-; DU is data upstream which is from S-chip to U-chip).
<table>
<thead>
<tr>
<th>Function</th>
<th>Direction (in NT1 configuration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameclock (8 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>bitclock (512 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>data upstream (512 kbit/s)</td>
<td>S → U</td>
</tr>
<tr>
<td>data downstream (512 kbit/s)</td>
<td>U → S</td>
</tr>
</tbody>
</table>

Figure 3-7. Physical representation of the K2-interface

The according dataframe conveyed over the DD and DU lines and timing is shown in the figure printed below.

![Dataframe and timing at the K2-interface](image)

Figure 3-8. Dataframe and timing at the K2-interface

As in the K-interface case, B1, B2 and D octets are transparently passed over the K2-interface. The S1 octet is left unused but is reserved for future use by the NT1. In downstream direction the S2-S4 octets contain information from the dataframe that is conveyed over the DSL at the U-interface. They are transparently passed to the SK2 chip. The S5-octet is set by the UK2 chip and transmitted to the SK2-chip. In upstream direction similar remarks can be made. Only the exact bitfunctions differ when up- and downstream data flow is considered.

3.3.3 The IOM interface

The IOM interface can be compared with the K-interface. It’s a Siemens trademark and thus applied to many of their components. There are two versions/generations of this interface: the IOM1 and IOM2. The IOM2 is the latest redesign of the IOM interface. First the IOM1 will be discussed.
This interface can both operate in continuous and multiplexed mode. In the first mode the
nominal bit rate of the data is 256 kbit/s. By increasing the clock frequency to 2 MHz, the
number of devices that can be connected simultaneously using the interface in a timeshared
way, is 8. The multiplex mode will be discussed when the IOM2 is discussed since this most
recent (and relevant) interface type differs from the IOM1. The physical representation of the
IOM1 in continuous (256 kbit/s) mode is shown below.

<table>
<thead>
<tr>
<th>Function</th>
<th>Direction (in NT1 configuration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameclock (8 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>dataclock (512 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>data upstream (256 kbit/s)</td>
<td>S → U</td>
</tr>
<tr>
<td>data downstream (256 kbit/s)</td>
<td>U → S</td>
</tr>
</tbody>
</table>

Figure 3-9. Physical representation of the IOM1 interface

It can be determined that the dataclock signal has a frequency which is twice the data
transmission frequency. This is also shown in the dataframe configuration and timing figure.

Figure 3-10. IOM1 (256 kbit/s mode) dataframe and timing description

The IOM2 (2nd generation) is an expanded version of the IOM interface. Besides slight
changes in the use of the monitor channel the IOM2 interface has a more flexible multiplexed
mode. In continuous mode (256 kbit/s) the IOM2 resembles the IOM1 and timing and frame
format can be found in the according alineas. In multiplex mode, the dataclock frequency is
variable between 512 kHz and 8 MHz (dependent on the number of timeslots required). The
dataclock frequency is (just like in the continuous mode) twice the data transmission frequency.

The physical representation of this interface in mux mode is shown below. As an example the
2Mbit/s interface configuration is given.
<table>
<thead>
<tr>
<th>Function</th>
<th>Direction (in NT1 configuration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameclock (8 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>dataclock (4096 kHz)</td>
<td>U → S</td>
</tr>
<tr>
<td>data upstream (2048 kbit/s)</td>
<td>S → U</td>
</tr>
<tr>
<td>data downstream (2048 kbit/s)</td>
<td>U → S</td>
</tr>
</tbody>
</table>

**Figure 3-11. Physical representation IOM2 interface in multiplexed mode (2Mbit/s)**

The timing considerations of the IOM2 interface in multiplexed mode (2 Mbit/s) are represented in figure 3.12.

**Frame Configuration**

- **B1**
- **B2**
- **monitor**
- **D**
- **C/I**

**Frame Timing**

- **data frame timing**
- **data clock (4096 kHz)**
- **frame clock**

**Figure 3-12. Timing and dataframe configuration IOM2 interface**

The timing and dataframe configuration diagram is shown for two elements (B1 and B2 of a frame contained in a single timeslot. The monitor and C/I bits establishing layer-1 control and maintenance functions are of course submitted to equal timing.
3.3.4 ST-bus

This interface is a Mitel trademark and applied to many of their components. It's a bus structure to which several components can be connected in a multiplexed (timeshared) mode.

The physical lines available at the interface are described in the figure printed below. Since this interface is provided by many of Mitel's components, there can't be given a specific direction as in the K/IOM cases. The ST-bus is also not specifically designed for a NT1 implementation and therefore the accompanying text isn't focused on that particular application. The direction of the frame/data clock signal is dependent on the mode a particular component is set in: Master or Slave. A Master device provides the ST-bus and components connected to it with a clock signal while a Slave uses it (is slaved to it).

<table>
<thead>
<tr>
<th>Function</th>
<th>Signal label</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameclock</td>
<td>FOb</td>
</tr>
<tr>
<td>dataclock</td>
<td>C4b</td>
</tr>
<tr>
<td>data input</td>
<td>DSTi</td>
</tr>
<tr>
<td>data output</td>
<td>DSTo</td>
</tr>
</tbody>
</table>

Figure 3-13. Physical representation of the ST-bus

As shown in the above printed figure, the data clock is twice the data transmission frequency on the DSTi/DSTo (in and output serial datalines; comparable with up and downstream datalines).

The frame format and timing are shown in figure 3.14
Frame Configuration

<table>
<thead>
<tr>
<th>Data frame configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame timing at 2 Mbit/s</td>
</tr>
<tr>
<td>ST-bus interface</td>
</tr>
</tbody>
</table>

Frame Timing

<table>
<thead>
<tr>
<th>Data frame timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data clock (4096 kHz)</td>
</tr>
<tr>
<td>Frame clock</td>
</tr>
</tbody>
</table>

Figure 3-14. Dataframe format and timing at the ST-bus

The D-octet contains 2 bit D-data. This defines the D-channel to have a 16 kbit/s datarate. If however a 64 kbit/s channel is required, the whole octet is used for D-data conveyance. The C-octet contains 8 maintenance/control bits. In this way a so called C-channel is established (64 kbit/s datarate).

3.3.5 Compatibility of the interfaces

Since it may happen that components are to be used during the implementation phase of the switching system, that are obtained from different manufacturers, the possibility of using different interfaces simultaneously is studied. Timing aspects are studied in particular. A difference in this part of the interface will cause asynchronous performance of the components which isn't acceptable. Differences in bitlocations/functions are easily removed by means of microcontrolled operation of the switching system.

Each combination between the earlier discussed interface types will be described now. Possible solutions for reaching compatibility will be generated.
**IOM-K**

Due to the fact that the development of these interfaces by resp. Siemens and Philips has been performed in close cooperation, these interfaces have full physical compatibility. Only some bit functions within the layer-1 control part of the dataframe are different but this is removed by the microcontroller intervenience at the switching system of the NT.

**IOM2* (mux mode) - ST-bus**

These two interfaces can be compared since they both offer 2 Mbit/s transport of data. The IOM2 has to be set in the mux mode then and a 4MHz clock should be applied. Accurate consideration of these interfaces however, reveals some differences. They are listed below:

1. ST-bus’ framepulse is *LOW* active while IOM’s is *HIGH* active.
2. Dataclocks are inverted (or half a period shifted related to each other).
3. Framepulse appearance instants and durations differ.

These differences should be dealt with in order to achieve full timing compatibility. Adjustment will be performed by hardware implemented in the timing unit that takes care of splitting U and S side timing (see paragraph 3.4). The location of data within the slotted dataframes differs but this is removed by microcontrol of the switch system.

The most important timing difference is the last mentioned ([3]). The other 2 are only a matter of inverting the clock signals.

Problem [3] is approached in two ways:

a. ST-bus components slaved to IOM2 components (in mux mode) via the local timing (adjustment) unit.

b. IOM2 components (in mux mode) slaved to ST-bus components (in master mode) via the local timing (adjustment) unit.

**case a.**

The first case deals with this problem: how to transpose an IOM2 frameclock to an ST-bus frameclock?

Looking more carefully at the timing diagrams of both interface types (fig. 3.12/3.14) it can be seen that at the raising edge of the frameclock signal at the IOM2 interface, the data becomes valid. At the ST-bus the data becomes valid if the frame clock pulse has been active for half a...
data clock period. So the frameclock should appear before the data becomes valid.

By means of coverage of IOM2 and ST-bus timing diagrams (alignment of the framesignals which in fact is connecting the frameclock signals), the consequences of simply connecting IOM2 and ST-bus interfaces can be seen.

![Timing comparison diagram](image)

At this point ST bit overlaps two IOM bitlocations which causes data errors.

---

**Figure 3-15. Timing comparison of the ST and IOM2 busses**

It can be seen that different bitlocations (e.g. ST-256 and IOM 256/1) cover eachother which causes data to be undefined. This is unacceptable!

A solution for this problem is to delay IOM datatransport for one dataclock period. In that case the IOM2 dataframe in fig. 3.15 will be shifted over a quarter of a bitperiod to the right which causes total coverage of the dataframes and thus dataflow synchronism.

Another item to be taken care of is the generation of a one (data) clock-period-lasting framepulse for the ST-bus timing (instead of half a data frame like the IOM2 framepulse). This should be done in the local timing unit by means of hardware.

**Conclusion:** ST-bus component can be slaved to IOM2 components by limiting the IOM framepulse to a single clockperiod and by delaying the IOM dataflow over half a data-clockperiod using buffers. The frame signal should also be inverted.
case b:

This time IOM2 components are slaved to the ST-timing (so ST-component is in master mode). The actions to be taken by the local timing unit are described here.

First of all, the ST-bus frameclock signal should be inverted and stretched. IOM2 expects framepulse length to be equal to half a dataframe period (0.0625ms) instead of the ST-bus framepulse length of one dataclock period.

Furthermore, IOM2 expects data to be valid as soon as the frameclock edge raises. In the ST-bus timing data becomes only valid after half a clock period. This problem is similar to that in case a but can be solved more easy (without data buffering). By means of delaying the frameclock signal in the timing unit by half a dataclock period, IOM2 timing is achieved.

The last difference to be removed is the inverting of the dataclock signal.

Conclusion: by means of delaying the resulting framepulse generated in the timing unit out of several ST-frameclock generators, and furthermore inverting it together with the dataclock signal, IOM2 components can be slaved to ST-bus components (in master mode).

3.4 Timing considerations at a multiple U-transceiver interface

This paragraph describes the problems that arise when the NT1 configuration is 'upgraded' to an intelligent NT system with several U transceivers (for example two). In the NT1 configuration a single U-transceiver synchronizes a locally generated clock (crystal) to the network clock and generates a 8 kHz frame clock. When the interface between S and U transceivers is deactivated (for reasons of reducing power consumption by the NT), it can be awoke again by both TE and the network.

When the NT is considered now with e.g. two U and four S transceivers, a difference can be seen. Since each U-transceiver extracts its timing from the DSL it's connected to, the separate data and frameclock signals are not synchronous. The local switching system however must be data synchronized to all datastreams that are input, a master clock signal has to be generated.

The situation may occur that one interface or even all are deactivated. In that case the local timing unit should still be able to provide the S-transceivers with a master clock signal since local calls should be supported.

These two items (master clock generation and (de)activation processes) will be discussed in the next paragraphs.

3.4.1 Master clock generation

In order to describe the timing processes that are active in the intelligent NT, U and local/S
Each U-interface provides a different timebase which in fact is made up out of a frame- and dataclock signal. Dataclocks may be different as their phase is concerned. But their frequency (based upon a local crystal) is equal. Frameclocks however are equal as frequency is concerned but they can be a-synchronous (due to differences DSL linelengths which cause differences in frame arrival times). In the worst case they differ 125 µs (equal to one frameperiod). Based upon abovementioned information, the following method for master clock generation can be used.

The local timing unit monitors the incoming frameclock lines and checks which U-transceiver generates its frame active pulse latest (within an 8 kHz period). The latest appearing frameclock will determine the frameclock signal at the master clock side. The dataclock is created now by means of phase locking of a locally provided crystal clock base with the clocksignal belonging to the framesignal that has the Master status.

Since the controlling part of the intelligent NT can register how many interfaces are powered down, the timing unit can be informed about how many frameclock pulses can be expected. This will remove a situation in which the timing unit waits for a frameclock signal from a certain U-transceiver when this transceiver won’t send one as its interface is deactivated.

The according dataflow conveyance between U and switching unit must consist of elastic buffers (register with different clock in/ clock out timebases). In this way the a-synchronism of the frameclocks will be suspended. The buffers are required for both up- and downstream directions. Their required depth is equal to the dataframe length ($2B + D + \text{layer1 control}$). The dataflow configuration controlled by the clock generation unit is shown in fig. 3.17.
3.4.2 Activation and deactivation

The activation and deactivation processes are applied to the interface between S and U transceivers (like K and IOM). In the intelligent NT case this situation needs attention since local communication should always be provided.

Deactivation is initiated on command of the network. The U-transceiver will deactivate its interface towards TEs (or more specific, the switching system) in order to reduce power consumption. This system is only known in the CCITT ISDN specifications. The USA specs don’t mention this process.

When looking at the intelligent NT’s timing structure (fig. 3.16) two reference points can be determined: the U-side timing and S-side or local timing (in the NT1 case these points coincide). If a deactivation command is sent by the network to a U-transceiver, it should deactivate only its own interface. So the other interfaces remain in their current state. The local timing unit still processes the incoming time signals and generates a master frame and dataclock. Even if all U-transceivers have deactivated their interfaces, the timing unit should continue providing the S-side with clocks. This is achieved by the locally applied crystal oscillator.

Figure 3-17. Dataflow at the U-side controlled by local timing unit
It thus appears that the interfaces at S-side are never deactivated. Accordingly, activation will never be initiated by a TE. If the U-side interfaces are all deactivated and a TE demands a call setup to the network from the Intelligent NT, this activation procedure has to be performed by the Intelligent NT. Next, the interface will be activated again and the local timing unit has to be adjusted again to the clocks provided by the U-side interface.

In case the activation request is originating from the network, the procedure is similar.

The design of a local timing unit and the software to support it remain for further study.
4. Hardware configurations for the intelligent NT

4.1 Introduction

This chapter is concerned with a structural study on possible architectures, which implementations in components will show a performance as stated in the product definition part of this report. This study is quite a technical approach of the switching concept applied in the intelligent NT and it constitutes the base upon which the implementation action is initiated. Therefore, considerations that will play an important role during the implementation phase have been taken into account.

During this design phase, a target list is used to check the design upon compliance with design objectives desired.

An important choice for the implementation is: should the intelligent NT be built up out of custom designed ICs or 'off-the-shelf' components? This difference in implementation approach has its impact on the design phase; it has led to two different switching architecture systems: centralized and decentralized.

These two configurations will be studied thoroughly and each will lead to different architectures which are described more detailed.

Support of packet switched data transfer via the D-channel (LAN establishment) is described for each configuration in separate paragraphs.

4.2 Targets for the hardware design

In order to achieve a design for an intelligent NT which responds to ideas and requirements that are alive amongst technical and commercial people involving the product, a list has been made that contains those targets that are closely related to the hardware design of the product. During this architectural design phase of the project, these targets will be referred to when a particular part of the design has to be justified.
• Quick implementation acquired using a design based upon standard components

• Modular system: since there's uncertainty about the optimal configuration of the intelligent NT as number of U and S transceivers is concerned, it must be made possible to in- or decrease this number of transceiver chips and thus achieving a most cost-effective configuration.

• The basic configuration supports circuit switching between B-channels of terminals connected to the NT. Packet switching via the D-channel is an optional feature.

• Support of heavy packet switched data transfer (like host terminal communication) using a 64kbit/s B-channel by means of (a) special module(s) which is (are) easy to apply to the basic configuration and which possibly are shared by several terminals.

• Generic system (upgrading/updating the system should be easy; e.g. packet data control unit)

• The system should meet with certain physical dimensions.

These targets have to be kept in mind during the design of the system.

4.3 D-channel processing considerations

4.3.1 Introduction

During the design phase of the hardware for the intelligent NT there is put a considerable amount of effort in investigation of a cost-effective D-channel processing (in fact low layer-2 support) method for the terminal side of the NT (S-side). For the network side (U-side) it soon appeared that there are not many configurations possible (U-transceivers with on-chip HDLC and appropriate linecoding are not available -yet-). In this paragraph possible D-channel routing and processing schemes will be analyzed. In the switching design paragraphs, in which they will be referred to, they are shown applied to particular designs. The description is split into a separate D-channel configuration and a common D-channel configuration.

4.3.2 Separate D-channels

This D-channel processing method identifies a D-channel at each transceiver's S-interface. The low layer-2 functions are performed by multiple single channel HDLC units or a multiple channel HDLC unit. They both identify towards the micro controlling unit that takes care of the call processing what channel the data is originating from (or what S-transceiver the sending terminal is connected to). This is necessary from a B-channel switching point-of-view: the NT
should know where a terminal is located (down to S-interface level) in order to be able to establish a switched B-channel connection.

4.3.3 Terminals sharing a D-channel

If the S-bus circuit is considered, one can determine a group of terminals connected to a bus providing two B channels and one D-channel. The terminals are parallel connected to the bus which implies that they are physically connected to the same (4) wires. Being connected to the bus doesn’t yet imply that the terminals have access to the channels too. This access is controlled in order to prevent collision at the channels. A collision detection/resolution mechanism takes care of the controlled use of the D-channel. Access to a B-channel is obtained after a call set-up procedure via the D-channel.

The common use of the D-channel by several terminals as it exists on the S-bus is interesting for application in configurations where a lot of terminals are concerned. In that case, only one low layer-2 processing device (HDLC) has to be used which may reduce configuration costs. In the intelligent NT case with common D-channel, several S-bus circuits are concerned that share a D-channel while remaining separate use of 2 B channels per S-bus. The star-NT (par. 3.2.4) can be compared with this situation except for the fact that all (2B + C + D) channels are shared by the S-bus circuits.

The accessibility of the D-channel by several sources (terminals) is controlled at the S-bus by a mechanism. It should be investigated now how the shared use of a D-channel by terminals connected to different S-busses can be arranged.

In order to solve this problem, the star-NT should be looked at (par. 3.2.4). In that configuration the D-channel collision detection/resolution mechanism as it exists at the S-bus is applied to all terminals connected to several transceivers. This is established by means of connecting the Echo channel (available as a pin at the S-transceiver chips) and thus providing a D-channel monitor system for all terminals. This echo channel connecting could be called a logical common D-channel establishment (existing at the S-interface of the transceiver). If the D-channels at the S-transceivers would also be connected (and-ed), a physical D-channel would be established too (existing at the system interface of the S-transceiver).

As already pointed at in par. 3.2.5, the and-ing of 2B datalines which is also done in the star NT configuration, isn’t applicable to the intelligent NT since only two B channels would be provided (to several S-interface circuits).

The common D-channel low layer-2 support can be configured in two ways: using a single channel HDLC or multiple channel HDLC. These two situations will be discussed in the following paragraphs.
4.3.3.1 Low layer-2 support by a single channel HDLC unit - both logical and physical common D-channel -

In this case all terminals are low layer-2 supported by a single channel HDLC; the unit is shared by all terminals. The main problem concerning this configuration is the terminal localization. Due to the fact that the HDLC receives data over a single line, it can't determine from which S-transceiver the data is originating. Since the B-channel switching system of the NT must have knowledge of the location of a terminal (itself identifying by means of its layer-2 TEI value) with respect to the S-transceiver it is connected to, some hardware has to be added to establish this localization function.

A method that has been studied is shown below.

![Diagram showing active D-channel detection mechanism]

*Note: the detection device detects what channel is active by zero detection. The Dch portnumber that denotes the active D-channel is output.

**Figure 4-1. Active D-channel detection mechanism**

The system de-multiplexes D-data from the 2B+D-control line (at the system interface) for each transceiver and applies these serial D-lines to both an 'and' gate and a 'detection' device. The detection device checks what channel sends a zero. That channel is the channel at which the active terminal is sending. This zero will be detected at the beginning of the D-channel access procedure where a terminal sends consecutively one zero, six ones and one zero. There won't be sent a zero by a terminal using another physical D-channel unless it has received 11 consecutive ones, indicating the D-channel is free.

In case two (or more) terminals connected to different S-bus circuits (and thus using different physical D-channels) start sending the '01111110' pattern simultaneously, the detection mechanism still functions properly. Though it takes some time* before the actual active D-
channel can be indicated, eventually one terminal will remain access to the D-channel while others don’t. At that time the zero detection will deliver the one-and-only active D-channel. This concept is acceptable from hardware development point-of-view. It only requires a mechanism that performs zero-detection and generates the active D-channel number.

<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flag</td>
<td>Address (high order octet, SAPI)</td>
<td>Address (low order octet, TEI)</td>
<td>Control</td>
<td>(Control)</td>
<td>Information</td>
<td>Frame check sequence</td>
<td>Frame check sequence</td>
</tr>
<tr>
<td>Octet 1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>(5)</td>
<td>N-2</td>
<td>N-1</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 4-2. Layer-2 frame composition

The herementioned detection mechanism is described on its functional behaviour. An actual implementation is beyond the scope of this study.

4.3.3.2 Low layer-2 support by a multiple channel HDLC unit -logical common D-channel-

This configuration distinguishes separate D-channels at each S-transceiver interface. But due to the logical D-channel only one of them will be active at once. A multiple channel HDLC unit

* maximum 1.5 ms, the transmission time of 3 bytes over the D-channel; the first byte is the flag, the second the SAPI which still may be equal amongst the sending terminals and the third byte the TEI which is different amongst the terminals in a configuration like this. See also fig. 4-2.
is applied to perform low layer-2 functions (only one channel active at once unless several
terminals have initiated sending D-data simultaneously like described in the previous
paragraph; in that case several channels will be used until the D-channel contention/resolution
mechanism becomes active and only one terminal will maintain access to the D-channel).

What's the use of this configuration for it doesn't reduce the number of HDLC channels?

This configuration has been designed as it appeared during the physical common D-channel
investigations that the terminal localization could be a problem. This system detects by itself to
which S-transceiver the D-data sending terminal is connected since it detects the active D-
channel. It accordingly removes the terminal localization problem which is an advantage.

4.4 Packet switched data support: general considerations

4.4.1 Considerations using the OSI model

The intelligent NT should be able to support (as an option) the transport of packetized data (p-
type) via the D-channel. This data can be compared with the transport of signalling data (s-
type) via the 16 kbit/s D-channel conform the HDLC protocol. Both p- and s-type data packets
are contained by the info field of the layer-2 frame. They differ however as their length is
concerned. S-type data packets have an average length of 50 bytes while p-type data packets are
considerably larger (140 bytes average is a plausible value). This fact should be kept in mind
when the hardware is being designed.

A feature offered by the application of the packet data support option is the establishment of a
Local Area Network (LAN). In that case a configuration like shown below should be
established.

![Figure 4-3. Terminals connected to LAN](image-url)
The terminals connected to the LAN are ISDN packet data terminals (e.g. X.25 terminals) or terminals using a Terminal Adapter (TA) to perform like an ISDN terminal. The HOST terminal shown in the figure is a terminal like the abovementioned but its functional behaviour differs: it acts as a node in the network to which all terminals are connected. So it establishes the routing of the packets sent by the terminals. Though terminals virtually communicate in a point-to-point connection, they actually communicate via the HOST terminal.

In order to be able to handle multiple terminal-HOST links, the HOST-LAN connection should have higher transport capacity than the terminal-LAN connection. This requires special packet-handling functions from the LAN (multiplexing). The separate connections (terminal-LAN and HOST-LAN) are described now individually.

Terminal - LAN connection

The communication between a packet terminal and the LAN (or physically the intelligent NT) is performed using a 16 kbit/s D-channel. Each S-bus circuit has one D-channel available and there’s no common use (between terminals connected to different S-bus circuits) of the D-channels. Using HDLC frames to convey datapackets via the D-channel (LAPD protocol) causes the datarate of the packet terminals to be limited to 9.6 kbaud/s (using the D-channel; higher datarates can be served using a B-channel for the packet transport -LAPB protocol-).

At layer-2 each terminal has a datalink with the LAN identified by a TEI value. This TEI value may be unique per S-bus circuit or amongst all S-bus circuits (dependent on the TEI assignment procedure implemented in the NT). Together with a layer-1 S-bus identifier each terminal is uniquely identified (this S-bus identifier is not necessary in case TEI values are unique over all interface circuits).

At layer-3 a Channel Identifier (CI) is used to distinguish packet switched calls that exist using the same datalink (comparable with Call Reference value for circuit switched calls).

Abovementioned identifiers are used in the HOST-LAN communication discussion later on.

Since the implementation of layers within the intelligent NT should be minimized, it’s investigated if it’s possible to implement only up to layer-2. This is shown in the (OSI) peer-to-peer model in fig. 4-4.
Communication between terminal and intelligent NT is restricted to layer-2 peer-to-peer communication. LAPD is the protocol used for the establishment of layer-2. HDLC units embody the low layer-2 functions to be performed. The SAPI value (layer-2) is used to distinguish packet data (SAPI=16) and signalling data (SAPI=0): in case signalling data is concerned it should be routed to the microcontrolling unit that performs further layer-2 and layer-3 processing; in case packet data is concerned the packet should be routed to the HOST terminal using the datalink that is established between LAN and HOST (identified by TEI value). The communication between LAN and HOST is described in the next part of this paragraph.

HOST-LAN communication

Due to the fact that several terminals are connected to the intelligent NT and only a single HOST terminal using a channel having higher capacity than the D-channel, packet multiplexing should be established by the NT.

At layer-1 there exists a single physical link between HOST and LAN. The capacity of the channel is 64 kbit/s (B-channel).
At layer-2 there is established a single datalink between LAN and HOST terminal (in X.25 only one datalink can be established by a terminal at once). Since a B-channel is concerned the according layer-2 protocol involved in the datalink establishment is LAPB. At layer-3 there can be established several calls between the HOST and the LAN identified by unique Channel Identifiers (CIs; X.25 parameter).

The support of the packet switching functions by the intelligent NT embodies a parameter substitution to assure uniqueness of calls. The main cause of this requirement is the fact that there’s only one datalink between HOST and LAN and several datalinks between terminals and LAN. A kind of mapping should be made within the NT. A possible way to establish identifiability of the packets is to give them unique CI values. This mapping is shown in the figure shown below (TEI values are supposed to be unique at the S-side).

<table>
<thead>
<tr>
<th>Call identification in:</th>
<th>Terminal a</th>
<th>Mapping in int. NT</th>
<th>Call identification in:</th>
<th>HOST terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer-2</td>
<td>TEI = 1</td>
<td>1 → 20</td>
<td>TEI = 20</td>
<td></td>
</tr>
<tr>
<td>Layer-3</td>
<td>CI = 0</td>
<td>0 → 6</td>
<td>CI = 6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Call identification in:</th>
<th>Terminal b</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer-2</td>
<td>TEI = 2</td>
<td>2 → 20</td>
</tr>
<tr>
<td>Layer-3</td>
<td>CI = 0</td>
<td>0 → 25</td>
</tr>
</tbody>
</table>

Figure 4-6. Mapping of packet identifiers

The packets identified by different CI values at the HOST side of the network are transported using a B-channel conform the LAPB protocol. So each packet is nested within a layer-2 frame. The multiplexing is performed at layer-2.

Abovementioned is a possible way to support packet switched datacommunication between X.25 terminals using layer-2 multiplexing. In this way there’s no need to implement layer-3 for packet switching within the intelligent NT. Other possibilities to support packet switching which could be studied are:
• The use of the X.75 protocol. An advantage of this protocol could be the
existence of several datalinks for a terminal. This could enable several
point-to-point connections at layer-2 between HOST and terminals.

• Implementation of layer-3 supporting packet switched calls within the
intelligent NT. This would remove the use of HOST terminal. This is
advantageous from different objectives:

[1] the NT embodies a packet routing function. In the
erlier described example communication between
two terminals involves packet routing by the HOST;
there's no direct packet transfer between the
 terminals in that case which causes packets to use
long paths.

[2] from practical point-of-view the use of a HOST
terminal could be disadvantageous towards the NT's
user. The HOST is a terminal that embodies the
packet switching function and thus LAN
establishment of the NT. Databases implemented in
only this terminal make it determinable amongst the
other packet terminals connected to the NT. If for
example this terminal is switched off, the packet
switching support is disabled and the terminals can't
communicate anymore using the D-channel.

• CCITT is underway to support X.25 on B-channels using LAPD instead of
LAPB. This enables multiplexing on layer-2. It is required however that the
HOST supports such an interface.

4.4.1.1 Different D-data processing configurations supporting packet switching

Though the previous part was merely focussed upon software aspects of the packet data
switching support, the next part will show some configurations using particular hardware
components. This is in fact a matter of implementation for which this chapter may not be the
right place to discuss about. However, discussing the packet switched option using examples
make the packet switching support more tangible. During the study phase of packet switching,
it turned out that this item is subject to discussions due to the lack of 'guidelines' that indicate
'which configuration is most appropriate for which application'. In order to present a usable
configuration, some functional blocks have been replaced by actual components.

The first architecture is characterized by the fact that both p and s-type data are routed via a
UART to a central microprocessor. This central processor establishes layer-2 and 3 for
signalling data and only layer-2 for p-type data. This processor is powerful enough to handle
all D-channel data generated at all transceiver circuits (interfaces to terminals and network).
The configuration's functional behaviour is described after an illustration showing this D-channel dataprocessing architecture.

Data is being low layer-2 processed by the HDLC. It contains FIFO buffers that can be accessed via the microprocessor bus by the microcontroller. Parallel data is transferred between HDLC and memory under control of the microcontroller (e.g. 8051*) towards the central microprocessor (where packet and signalling data is distinguished and split) data is transported serially via a UART. Conveying the data via a serial line is necessary since this configuration is used in applications where layer-2 and 3 and low layer-2 processing are performed on different circuit boards. The use of parallel communication lines is not recommendable for the interconnection of different circuit boards: it’s expensive from manufacturing viewpoint and the distance to be bridged could cause communication errors. Using a UART to communicate with the processor with less error probability and serially, is a good solution. The local memory is used to act as a buffer between HDLC and UART since these components (may) have buffers with different depths which causes buffer over- or underrun.

Actually, the microcontroller only takes care of conveyance of data between HDLC and UART

* the application of the i8051 for microcontrolling functions is quite usual in the IAS environment; the experience and test tools that exist there can be made use of which saves time and money.
v.v. (interrupt driven). This loads the microprocessor bus and the microcontroller rather heavily. Application of a second HDLC isn't therefore quite recommendable (and may even be impossible if packet switched data is concerned). Servicing interrupts becomes a quite critical item then. A solution for this problem might be the application of a DMA controller. Such a unit performs the datatransfer between memory and HDLC/UART buffer without loading the microcontroller. But in that case the controller can perform other processes. A dedicated communications module that has implemented a 8051, DMA, UART and HDLC with collision detection/resolution mechanism is the i83c152. The configuration based upon this component is shown below. Just like the previous one, this component is located close to the transceivers (U&S). So the configuration is also a decentralized D-data processing configuration.

![Diagram of D-data processing using the i83c152](image)

- marks the components implemented on decentralized board
- - - - - - - marks the blocks implemented on the i83c152 chip

**Figure 4-8.** D-data processing using the i83c152

The configuration supports a single transceiver/D-channel (just like the previous one) since sharing the component by means of applying a second HDLC to the external microprocessor
Bus possibly can’t be handled by the microcontroller and DMA (only 2 channels). A remarkable feature offered by this component is its support of a packet bus which access is controlled by a collision detection/resolution mechanism built in the HDLC component. So this packet bus can be shared by several components like this communication controller. This bus can be used to multiplex packets to a HOST terminal. The datarate at this bus can be 64 kbit/s but higher rates are also applicable (dependent on the number of terminals connected to the intelligent NT).

Signalling data isn’t processed within this component but conveyed via the UART to the central layer-2 and 3 processor. So compared to the previous architecture, this configuration performs a p and s-type data splitting function using a dedicated component containing all blocks necessary to support this function.

A fully centralized processing configuration is shown in the next illustration.

---

* for s-type data layer 2 & 3
for p-type only layer 2

**Figure 4-9. Centralized D-data processing**

The transceivers (S&U) are considered to be located on different cards. They convey their D-
data over serial lines (conform HDLC protocol) to low layer-2 controllers. These are connected to a microprocessor bus. Their buffers are written and read controlled by the microprocessor but datatransfer between these buffers and memory is performed by a DMA controller (multiple channel). The microprocessor establishes layer-2 and layer-3 for signalling and layer-2 (only) for packet data. As in the previous configuration signalling and packet data are split at layer-2 (using the SAPI value). Packet data is conveyed to the HDLC that communicates with the HOST using a B-channel.

This architecture is quite flexible: the packet switched data support option is a matter of adding an HDLC unit, applying a microprocessor that is powerful enough and implementing another software release supporting this option. If implementation of layer-3 for packet switching is also desired this is a matter of implementing the appropriate software in ROM.

So the last architecture turns out to be a quite feasible configuration. However, it should be remarked that there definitely can be compared other architectures that might be applicable and even more feasible for the packet switching support. The framework of the study however didn't allow a thorough investigation of this complex D-channel processing problem. As remarked earlier, the packet switching support implemented in the BAMX is comparable to the packet switching support in the intelligent NT. Keeping up with the developments achieved in the design of the BAMX is therefore very recommendable.

4.5 Two approaches for the switching system design

As already discussed in the introductionary paragraph of this chapter, being prepared for the implementation of the hardware design to be established in this chapter, one should already take into account possible impacts this implementation phase may have on the architecture. This might not be in accordance with design methodologies that determine individual design levels corresponding to a design phase.

Theoretically, a designer being in a particular design phase shouldn't face the problem from other levels at the same time. He ought to stick to his design level. In practice however, this 'rule' can't be obeyed totally. It appears then that descending and ascending to lower levels* in the design (JoJo-effect) can't be eliminated. On the contrary, this shouldn't even be omitted since it eventually benefits the design as a whole!

As for the implementation, an item to be discussed is whether to use custom designed chips or 'off-the-shelf'-components to establish the switching function of the intelligent NT. In order to be able to draw a conclusion, this chapter should provide some issues that can support such a conclusion. Therefore, two different switching architectures have been investigated.

* Low corresponds here with a more detailed design approach. In a top-down design, that is being pointed at here, the highest level corresponds with system description level where large blocks are used while the low level identifies the functional behaviour as defined on higher levels represented by small, detailed blocks.
The first architecture uses common transceiver chips (U&S) as supplied by different companies (APT, Siemens, Philips and Mitel). A centrally located switch and control unit takes care of the actual data processing and switching.

In the second approach the transceivers are connected via separate, decentralized switching and control units to a busstructure. These units could even be implemented in the transceiver chips.

Both configurations will be discussed; their functional behaviour and detailed descriptions shall be given in separate paragraphs. Since D-channel data processing can be performed in different ways and causes the existence of different configurations, each switching configuration will be described conform two concepts for D-data handling.

Finally, packet switching support will be described for each configuration.

4.6 Centralized switching

4.6.1 Introduction

The centralized switching architecture is a method on which the intelligent NT's hardware design can be based. It mainly consists of a central switching block to which peripherals like user and network interface components and control devices are applied. A functional description will be given to illustrate the configuration's main functions and requirements. The implementation of the central switching block is likely to be an off-the-shelf component since custom designing such a dedicated chip takes too much time and is quite risky as product's market perspectives aren't very well known. Therefore a detailed description of the central switching block is omitted here. It will be given when a possible implementation is discussed.

Two concepts will be used for this functional description. They differ as low layer-2 processing of D-data by an HDLC unit is concerned.

The functional description of the central switching configuration is mainly focussed on the functional behaviour/capabilities of the central unit and a description of the units applied to it. Their system interfacing is of great importance.

4.6.2 Two concepts

There can be distinguished two concepts for the centralized switching architecture. The difference between the concepts is based upon the transport of D-data to-and-from S-transceivers and to-and-from the central switching unit.
The first concept conveys all D-data to the central switching unit where it is transported to and from a HDLC unit the in/output buffer that embodies the gateway between register and peripheral. The HDLC can be accessed by the micro controlling unit.

The second concept is based on the application of S-transceivers that have on-chip an HDLC and microprocessor interface circuits. This doesn't change the central switching unit architecture drastically but analogous to the decentralized switching concepts, the centralized architecture description has been split too in two concepts concerning the way D-data is processed.

4.6.2.1 Functional description centralized switching architecture: all D-data to central unit (concept 1)

This paragraph describes the centralized switching architecture where concept 1 is used for D-data routing/processing. So D-data from/to both userside (S-side) and networkside (U-side) is routed to and from the Central Switching Unit (CSU). To the CSU an HDLC unit is connected via serial bidirectional lines, that performs low layer-2 functions. D-data from/to the CSU is conveyed via these lines. The configuration will be described now structurally: blocks that are shown in fig. 4-10 are separately described.
Blocks

- **S-transceiver**: This unit provides towards the user a CCITT 1.430 S-interface which consists of 4-wires. Towards the switching system the unit provides two serial datalines (up- and downstream data; format: \(2B + D + C\) [layer-1 control/maintenance]). Above that, the transceiver has two clock input ports where Master clock signals (system clock signals
provided by local timing generator) are applied to. In order to reduce input ports at the CSU (see also: CSU block description), it is very useful if the interface towards the CSU (component dependent) is able to operate in multiplexed mode. In that case the datalines of several S-transceivers (and U-transceivers if this mode is supported at the U-side) can be combined to form a multiplexed bus which reduces in- and output lines at the CSU.

• **U-transceiver:** This block is the physical interface towards the network via the CCITT recommended U-interface (two wires). It extracts a frameclock signal from the DSL by means of locking-in at frame arrival. Since there can be several U-transceivers involved with different clocks, these clocks can’t just be led to the S-transceivers and switching device for there would be no synchronous data transport. Therefore, these signals are connected to a timing unit that provides master clock signals*. The a-synchronous datalines provided at the U-transceivers' interfaces towards the switching system are being entered to the elastic buffer module that aligns the dataframes originating from the U-transceivers and adjusts dataframe timing for frames being sent towards U-transceivers by applying the timing of those particular interfaces.

If multiplexed data inputs are used at the CSU (mentioned when S-transceivers block is discussed) the U-transceivers not necessarily need to support this (multiplex) mode due to the fact that their data is being transceived to/from the CSU by the elastic buffer module. Therefore, this buffer module must support this mux-mode (if used).

• **CSU:** The Centralized Switching Unit is the block that performs the actual (B-channel) switching. B-data can be switched between several peripheral blocks. They are connected to the CSU via the in- and output ports. As described already earlier (S- and U-transceiver descriptions) data from transceivers can be in- and output to/from the CSU most economical using time divisioned multiplexing for the data transmittance (reduced number of pins at component). Layer-1 control should be performed via microprocessor access of the (control) data transceived to/from the U- and S-transceivers. To this end the CSU is equipped with a parallel microprocessor interface. Layer-1 control consists of control of the bitfunctions of the Layer-1 control/maintenance fields within the dataframe to/from U- and S-transceivers. This is necessary to adjust the communication between these components due to the fact that they are performing in a different situation (compared to the NT1). For example: a command to de-activate S-transceiver as it is sent by a U-transceiver shouldn’t be conveyed transparently to an S-transceiver since local communication should always be possible and S-transceivers accordingly shouldn’t be de-activated.

Serial data retrieved from the transceivers is parallel written into the B, D and layer-1 data register and memory-mapped accessed by the MCU. Since D-data should be transported via serial lines between HDLC units and CSU, it is useful to reserve a special D-data input/output buffer. Again, this port could consist of separate physical lines or two multiplex lines (transmit/receive) to the HDLC unit. Latter reduces the number of pins at the CSU. The CSU receives clock signals from the local timing unit.

• **HDLC:** This unit establishes low layer-2 functions of the D-datatransport between user and intelligent NT and network. It transceives D-data either via serial datalines or via

---

* will be discussed in 'local timing generator' block description.
multiplexed receive and transmit lines (this depends on the in/output port configuration at the CSU). De-packetized data is microprocessor accessible via a Micro Processor Interface. The configuration of the HDLC block depends on the D-channel processing method used. This will be discussed in the next paragraph.

- **MCU**: The Micro Controlling Unit consists of a microprocessor and peripherals like memory and DMA controller. It communicates with the CSU, HDLC, PCM adder and the tone generator via a MPI. The MCU takes care of Layer-1, 2 and 3 processing.

- **Local timing unit**: This clock generator supplies the switching system of central/local clocksignals (which are: frame and dataclocks). This is necessary to support local calls. Because, if the U-transceivers would be in a "powered down" state (not for USA systems), there would be no local clock and the local call feature would be disabled. This is prevented by the master clocks provision. Above that, within this switching system there should be bit synchronism which requires a master clock.

- **Elastic buffer module**: This module establishes the timing adjustment for datatransport between the local clock and U-transceivers’ clocks. It consists of a FIFO bank for each U-transceiver and data transmittance direction. At the CSU-side data is clocked (or: received and transmitted) using the master dataclock that is provided by the local timing generator. At the U-side the dataclock provided by the U-transceiver the data is originating from/destined for is applied (the principle of U-timing adjustment is explained in par. 3.4). The depth of the buffers is equal to the length of the $2B + D + C$ dataframe.

- **PCM adder**: This unit reads data from the CSU and after adding this data originating from different sources, writes it to the B-register again. The actual implementation of this component is for further study.

- **Tone generator**: It interfaces via an in/output port with the CSU and writes continuously different digitized tones (PCM encoded) to register locations that can be switched to B-channels requiring particular tones. The tones that it is able to generate are: dial, call waiting, busy, congestion and alerting. The functions established by this unit may also be established within the terminals connected to the intelligent NT.

4.6.2.2 Packet switching support

The centralized switching architecture according to concept 1, routes all D-data to a specific in-and output port. The serial lines (or single line in case the data is being multiplexed by the CSU) are connected with a separate D-channel processing card, establishing layer-2 and 3 for signalling data. In case packet switched data is supported, p-type data is layer-2 multiplexed and led to the host terminal using the HDLC protocol. This centralized D-channel data processing method is most appropriate, considering the fact that D-data is already routed to a central switching unit by the S and U transceivers.
As the MCU is concerned (shown in the functional behaviour diagram of the centralized switching architecture) the following should be remarked. Physically this controlling unit will be spread over two cards: the central switching card where a microprocessor performs layer-1 and switch setting functions (CSU control). The second part of the controlling is the layer-2 and 3 establishment. These functions are performed on the D-channel processing card, together with the p and s-type data splitting. Information transfer between these processing cards (and thus between layer-1 and layer-2 & 3) is performed using a UART device. Further information on this processing configuration can be found in par. 4.4.

4.6.2.3 Functional description centralized switching architecture: D-data partly processed decentralized (concept 2)

The second concept for the centralized switching architecture is based upon the implementation
of S-transceivers with HDLC and microprocessor interface provided on-chip. The configuration will be described blockwise. Blocks that don't differ compared to the description given according to the first concept, are labeled r.t.s. (remains the same). The configuration is shown in fig. 4-12.

Figure 4-12. Functional diagram centralized switching: concept 2
Blocks:

- **S-transceiver:** Apart from the features mentioned in the block list of the central switching architecture conform the first concept, this component also provides low layer-2 functions. These are performed by an on-chip HDLC protocol controller. A microprocessor interface (MPI) provides a communication port towards the micro controlling unit.

- **U-transceiver:** r.t.s.

- **CSU:** This unit remains the same as compared with the concept 1 CSU except for the fact that D-data needs not to be routed to the D-in/output port anymore. At that D-port only D-data to/from the U-transceivers is provided.

- **HDLC:** Except for its size (now there are as many HDLC channels necessary as there are U-transceivers) the HDLC unit is equal to the one mentioned in the concept 1 blocklist.

- **MCU:** r.t.s.

- **local timing unit:** r.t.s.

- **elastic buffer module:** r.t.s.

- **PCM adder:** r.t.s.

- **tone generator:** r.t.s.

4.6.2.4 Packet switching support

Packet switching support for the centralized switching architecture with D-data being locally (in S-transceivers) low layer-2 processed, can be established in different ways. Two architectures, that have been discussed in par. 4.4 are illustrated here.

The first one performs low layer-2 processing controlled by a microcontroller like the i8051.
Figure 4-13. D-data processing centralized switching architecture concept 2 using i8051

Each S-transceiver is equipped with a microprocessor port and HDLC device performing low layer-2 functions. The microcontrol is performed by an i8051 microcontroller located at the same card. A UART transfers both p- and s-type data to and from the central layer-2 and 3 control card via a serial bus; this bus is shared between several of the above described low layer-2 units at the S-side. D-data at the U-side is switched to an in-/output port of the CSU. HDLC devices located on the layer-2 and 3 control card are connected with this port.

An advantage of this configuration could be the reduction of processor load on the layer-2 and 3 control card due to the low layer-2 processing performed by the HDLC and i8051 placed at the same card where S-transceivers are located. But the hardware efforts necessary to perform (only) these low layer-2 functions make this configuration from costs and (Euro)card occupation rate viewpoint quite ineffective.

In the second configuration abovementioned disadvantage can still be mentioned but not to an equal extent. D-data is not only low layer-2 processed at the S-transceivers but p and s-type
data is also split there.

![Diagram](image-url)

**Figure 4-14.** D-data processing centralized switching architecture concept 2 using i83c152

As described in par. 4.4 this configuration routes p-type data to a packet bus and s-type data to the centralized layer-2 and 3 control card. The component establishing this function is the i83c152. The reduction of processing load caused by the application of this component is an advantageous item. But the remote (related to layer-2 and 3 control card location) performance of this splitting is not recommendable (as indicated in par. 4.4.1.1. already). Abovementhat, due to the fact that this configuration is only applied at S-side, the centralized layer-2 and 3 processing card must perform similar actions since packet switched data should also be supported at U-interface reference points. This causes the system to become a bit unbalanced: for the S-side packet switching is supported decentralized while for the U-side this function is performed centralized (the main cause is the lack of U-transceivers with low layer-2 functions implemented on chip).

Both configurations with low layer-2 functions performed decentralized at S-side don’t seem to be very advantageous for reasons described above. Except in case S-transceivers with low layer-2 functions on chip must be used, abovementioned configurations should be considered.
4.7 Decentralized switching

4.7.1 Introduction

This paragraph describes -for the intelligent NT - most applicable decentralized hardware architectures. The concept of these architectures consists of the decentralization of switch and control functions in (custom designed) components. These components are hardware implemented interfaces/switching blocks between user-network interface devices (transceivers) and a busstructure to which a main control unit is applied.

It will be shown in this paragraph that there are several possible busarchitectures that lead to differently configured switching blocks. The functional architecture of these blocks will be given in order to be able to determine the complexity of the block design.

4.7.2 Two concepts

D-data processing esp. low layer-2 functions (data packetizing and depacketizing) is performed by an HDLC controller. The location of this unit, processing data originating/destined for transceivers can be different. Though this might seem a matter of implementation, the hardware architecture is also influenced by this item. Therefore, the description of the decentralized switching architecture is given according two concepts that are mainly characterized by the location of this HDLC unit.

In the first concept the low layer-2 control is located centralized. The D-data is conveyed over the switch bus to the HDLC which is accessed by the centralized microcontrolling unit.

In the second concept the HDLC for processing the U originated/destined D-data is still applied to the switch bus while the S D-data is processed locally. This concept is mainly based upon the nowadays available S-transceiver chips with on-chip HDLC/microprocessor interface unit. When similar U-transceivers become available (e.g. Siemens ISAC-P) HDLC processing can be performed decentralized for both kinds of transceiver.

4.7.2.1 Functional description decentralized switching architecture: all D-data via switch bus (concept 1)

A general overview of the location of the Decentralized Switching Unit (DSU) will be given first. This scheme shows the in- and output datalines of the DSU to transceivers and busses. The actual functional description of the DSU will follow then.
Figure 4-15 shows this general overview of the decentralized switching architecture with the low layer-2 processing devices (HDLC) connected to the switching bus. The dataflow between the modules at the interfaces is an important issue in this scheme.

Figure 4-15. Functional diagram decentralized switching, first concept

The functional behaviour and interfaces will be described of each block that can be determined in the figure. Some remarks on other aspects of the architecture and functions will terminate the description.

Blocks

• S-transceiver: At the left side this unit provides a CCITT 1.430 4-wire S-interface. At this side the terminals can be connected. The transceiver exchanges data with the DSU over 2
serial lines: one transmit and one receive line. The data that is transferred is 2B + D + C (C is layer-1 control/maintenance data). This data is transceived timemotion multiplexed. The transceiver is provided with a frameclock and a dataclock that both originate from the DSU*

- **U-transceiver:** This is the physical interface towards the network via the CCITT recommended U-interface (two wires). It retrieves a frameclock signal from the DSL by means of locking-in at frame arrival. Since there are several U-transceivers with different clocks, these clocks can't just be led to the S-transceivers and switching devices for there would be no equal timing. Therefore, these signals are connected to a timing unit that provides master clocks signals*. The dataclock signal generated by the U transceiver (and typical for each U-transceiver too!) is used by the DSU to clock data synchronously to the U-transceiver. Just like the S-transceiver it has two serial datalines that transfer 2B + D + C data.

- **DSU:** The DSU interfaces to the transceivers and to a switch and a control bus. As the interface with the transceivers is concerned, there can be determined 2B + D + C receive and transmit datalines and bit- and frameclocksignals (the direction of these signals is different for S and U transceivers; this will be discussed during the detailed DSU description). The interface towards switch bus also consists of transmit and receiver lines but they only transfer 2B + D data. Access to the bus i.e. transmitting and receiving data is established respectively by means of fixed and micro programmed timeslot assignment. The local timing is provided by the master clocks signals that are generated by the local timing unit. Layer-1 data (C-data) will be available for microcontroller access in the DSU via the Micro Processor Interface towards the Control bus.

- **HDLC:** This unit establishes lower layer-2 of the datatransport between user and intelligent NT and intelligent NT and network. It transceives its D-data to and from the switch bus over serial transmit and receive lines. To this end it's assigned one timeslot for each direction (these timeslots are also used by the DSUs for transmitting and receiving their D-data). The deformatted data is transferred to the MCU for further interpretation. The HDLC is therefore equipped with a Micro Processor Interface to communicate with the MCU over a micro processor bus. In fact this functional block as it's drawn in figure 4-15 represents several HDLC receive and transmit channels. Each U-transceiver provides one pair of HDLC channels: one receive and one transmit channel.

- **MCU:** The Micro Controlling Unit consists of a micro processor and peripherals like memory. It is connected to the control bus and therefore can communicate with HDLC and DSUs. The MCU is able to handle basic call processing and supplementary services (layer-3) and data link establishment/maintenance (layer-2). Layer-1 control is established via C-channel access at the DSUs. The actual switching control is also performed by this unit since it programs the timeslot assignment in the DSUs.

- **Local timing unit:** This clock generator supplies the switching system of central/local clocksignals (which are: frame and dataclocks). This is necessary to support local calls. Because, if the U-transceivers would be in a 'powered down' state (not for USA systems), there would be no local clock and the local call feature would be disabled. This is prevented by the master clocks provision. Above what, within this switching system there should be bit

---

* Timing will be concerned when the 'local timing unit' is discussed.
synchronism which requires a master clock.

- **PCM-adder**: This unit is applied to support the three-way call feature. B-data is retrieved from the switch bus by means of timeslot access under MCU control. After adding this data (belonging to different data streams), it's written to the switch bus again. It provides both interfaces to switch bus and control bus. The actual implementation of this functional block remains for further study.

- **Tone generator**: A tone generator is the part of the configuration to generate the appropriate tones for 'dial', 'call waiting', 'congestion', 'busy', 'error' and 'alerting' states in case local calls are processed and during call set up when a terminal communicates with the intelligent NT. The unit interfaces with the switch bus possessing timeslots to which continuously tones are being sent. For each tone a particular timeslot is assigned. DSUs can access these timeslots to receive these tones. The generator communicates with/is controlled by the MCU via the control bus. The functions established by this unit may also be established within the terminals connected to the intelligent NT.

### Busses

- **Switch bus**: This is a time multiplexed single wire bus on which data is contained in timeslots. Each DSU can write (transmit) and read (receive) data from that bus over separate serial datalines. The occupation of the bus by DSUs is: B-channel - one timeslot; D-channel - two timeslots (transmit and receive). So in the basic configuration (2U and 4S transceivers) 18 timeslots are occupied by the DSUs. The occupation of the bus by the tone generator depends on the different tones that are generated by this device. The number of timeslots assigned to this device is equal to the number of different tones required. The timeslots occupied by the HDLC block are already accounted for by the DSU D-channels.

How is the switching function established? This is illustrated in fig.4-16.
Transmit data:

```
3 2 1
X

DSU1

DSU2

DSU3
```

Receive data:

```
3 2 1
X

DSU1

DSU2

DSU3
```

switch bus to which data is transmitted. Transmittance is performed with fixed timeslot assignment.

switch bus from which data is received by the DSUs in a time divisioned way. Timeslots are assigned in a programmed way.

* Time divisioned multiplex frame, timeslots 1-3. A cross in a timeslot means function (transmit/receive) enabled. So when e.g. location 1 is considered, one can see that DSU1 transmits data at that moment (i.e. in that timeslot) and DSU3 receives that data simultaneously. So data is transferred from DSU1 to DSU3.

Figure 4-16. Transmitting and receiving data from the switch bus by the DSUs using respectively fixed and programmed timeslot assignment

This figure shows how data can be exchanged by means of transmitting data to fixed assigned timeslots and receiving data in a programmed way (via a register that is accessed by a microprocessor). Though the same switching function can be established with a fixed timeslot assignment for data reception at the DSU and programmed data transmittance, the method shown above is safer from a practical/implementational point-of-view: it prevents simultaneous writing of data at the bus in case the timeslot assignment register isn’t loaded by the microcontroller yet (e.g. after reset). As an example, the dataflow concerning an internal call between two terminals connected to the same S-bus is shown.
Logical representation internal call between terminals connected to the same S-bus

Internal call switched by DSU over Switch Bus using timeslot interchanging

Figure 4-17. Internal call between terminals connected to the same S-bus

- The Control bus: This is a micro processor bus with the according (parallel) address and data lines and (serial) control lines. It establishes the communication between the DSUs and the MCU. It enables C-data (layer-1 data) accessibility by the MCU which is necessary to support layer-1 control. Each DSU is equipped with a Micro Processor Interface (MPI) to be connectable to the control bus.

Implementing this bus requires some adaptions: a parallel bus at the backplane of Eurocards is not recommendable from both cost (many wires) and reliability (parallel bus over long distance may cause errors) viewpoint. Modifying the parallel structure into a serial one is a good solution to reduce costs and error probability. A device establishing this parallel to serial function v.v. is a UART.

The DSU in detail

The architecture of the DSU will be given in case it's connected to an S- and to a U-transceiver. Though it would be most comfortable if DSUs for U- and S-transceivers would be equal but due to differences in timing a slight difference can't be prevented.

The first configuration shows the DSU for connection to a U-transceiver.
The transceiver provides the DSU of receive and transmit (2B + D + C) datalines and a clocksignal. The data is received/transmitted via the interface with the transceiver using the clock that is provided by the U-transceiver. The time-adjustment between transceiver and DSU is established by means of buffer elasticity built-in the S/P (serial to parallel) converter. The buffer input is clocked by the transceiver clock while the buffer output is clocked by the system dataclock (this is valid for the DSU data input from the transceiver; for data output to the transceiver a buffer is applied that acts the other way around). The master clocks are derived from the frame- and dataclock signals originating from two (or more) U-transceivers. So the frame clock of one of the U-transceivers is the Master. Frames arriving at the DSU before the last frame within a 125µs period should be buffered before offering them to the DSU (see also par. 3.4).

As inputdata is concerned, this data is converted to parallel data and written into a registerbank to provide microprocessor access (layer-1 maintenance and control). This transmit registerbank gates the B and D data bytes parallel to the P/S unit under control of the fixed TSA unit (the assigned timeslots differ for all DSUs connected to the same switching bus in order to prohibit undesired simultaneous access (they are externally set: different for each DSU within a NT system). This unit addresses the registers and causes the according contents to be put on the parallel bus to the P/S unit on fixed instants (corresponding with timeslots on the switch bus). The size of the registerbank is equal to the number of bytes out of which the 2B + D + C frame is made up.
The P/S unit at the control/switch bus interface finally transmits parallel to serial converted data to the switch bus.

As far as the lower part (data transmittance to the U-transceivers) of the scheme is concerned, an equal mechanism as described above can be determined except for the control of the (receive) registerbank. The receive timeslot assignment is microprogrammed and not fixed. To this end the programmable TSA unit is applied which is MCU accessible via the Micro Processor Interface (MPI). The TSA unit generates sequentially clockpulses that will activate the serial data input from the bus by the S/P device and it also addresses the register where data should be written to (which corresponds with a datachannel towards the transceiver).

The DSU to be connected to an S-transceiver and with D-data transfer to an HDLC via the switching bus is shown below.

![Diagram](image)

**Figure 4-19.** DSU configuration when connected to an S-transc. (S-DSU) and with D-data transfer via switch bus

Since this module differs only slightly from a DSU connected to a U-transceiver (U-DSU), only the different items will be discussed here. This similarity makes it possible to design a DSU that is able to act as U or S-DSU by means of external (e.g. pin on chip) setting of the unit/chip.

Though the U-DSU is only partly slaved to the master clocks, the S-DSU is slaved to this clock totally. So data input/output from/to the S-transceiver is synchronous to datatransfer internally and to the control and switch busses. So the buffer configuration as it is required in
the U-DSU to establish timing adjustment is not necessary here. The 'S-DSU to transceiver' interface differs from the 'U-DSU transceiver' interface as the direction of clock signals is concerned. The S-transceiver is slaved to the frame- and dataclocks that are generated by the timing unit in the DSU based upon the master clock.

4.7.2.2 Packet switching support

Dedicated packet switching support for the decentralized switching configuration where all D-data is transceived by the DSUs via a switch bus, is performed by a centralized unit.

* layer-2 and 3 processing for s-type data
only layer-2 processing for p-type data

Figure 4-20. D-data processing decentralized switching architecture concept 1

The centralized D-channel processing method is most applicable here since all D-data is accessible at a serial bus. In par. 4.4 it has been shown already that a central D-channel processing card is favourable compared to a situation where D-channel processing is performed at several cards. The central card is equipped with a set of HDLC devices. Their number is equal to the number of transceivers that are applied in the intelligent NT system. As discussed already in the centralized switching case, the layer-1 and switch control functions are
performed separately from the layer-2 and layer-3 functions. Communication between these separate processors is established using serial communication lines driven by UART. Further information on the way packet switching is performed can be retrieved from par. 4.4.

4.7.2.3 Functional description decentralized switching architecture: D-data partly processed decentralized (concept 2)

This architecture is characterized by the application of S-transceivers that are equipped with an on-chip HDLC unit. Though this might seem 'only' a different way of implementing, it determines the architecture in such a way that it requires discussion in this paragraph. The configuration is shown below.
Since this concept only differs slightly from the first concept, only those items will be discussed that actually differ. Other blocks remain the same (r.t.s.) and their functional description can be found in the description of the first concept.

**Blocks**

- **S-transceiver**: This component is different to the one mentioned earlier since an HDLC unit is connected to it directly. So there's no need anymore for the S-DSU to transfer D-data to the switch bus. The HDLC transfers formatted data to the D-channel towards the terminals connected to the S-interface and transfers de-formatted data via the control bus to the MCU.
• **U-transceiver:** r.t.s

• **DSU:** As interface signals are concerned there can be seen a difference between the transmit and receive lines connected to the switch bus from an S-DSU and a U-DSU. Due to the fact that D-data is processed locally at the S-transceiver, there's no need for the S-DSU to transceive D-data to the switch bus. This need *does* exist for the U-DSUs since their D-data has to be transferred to HDLCs connected to the switch bus. Choosing for a configuration where equal data is transceived to/from the bus may be advantageous from 'aiming for similar U and S-DSUs design' point-of-view, but not from switch bus use; each transceiver connected to the switch bus occupies two timeslots for D-channel transmit/receive data. Because an S-DSU would also occupy two timeslots though the data is not relevant due to the fact it has been processed by the local (near S-transc.) HDLC already, this would cause ineffective switch bus use.

• **HDLC:** The block labeled with HDLC points at the low layer-2 processing unit that takes care of packetizing and depacketizing of D-data from/to the network (U-transceivers). D-data from/to S-transceivers is processed by on-chip implemented HDLC units.

• **MCU:** remains the same (r.t.s.).

• **local timing unit:** r.t.s.

• **PCM-adder:** r.t.s.

• **Tone generator:** r.t.s.

**Busses**

• **Switch bus:** The timeslot structure of this bus is different to that of the first concept due to the fact that S-DSUs don't occupy timeslots for D-data transfer. This increases the maximum number of transceivers connectable to the bus.

• **Control bus:** r.t.s.

**The DSU in detail**

The architecture of the DSU will be given in case it's connected to an S- and to a U-transceiver. Though it would be most comfortable if DSUs for U- and S-transceivers would be equal but due to differences in timing and data transfer to/from the switch bus differences in architecture can't be avoided.

The U-DSU for this second concept is equal to the one applied in the first concept-architecture (see figure 4-18). Its functional description can be found there. The architecture for a DSU connected to an S-transceiver with the local HDLC processing capability, is shown below.
This S-DSU differs from the U-DSU on those points mentioned during the description of the first concept. Above that, the data transferred to/from the switch bus by this DSU is also different from the 'concept 1 S-DSU' due to the local HDLC. In this case there's no D-data transferred to the switch bus. This is shown in the above printed figure. As implementation is concerned, this is a matter of setting the TSA units in such a way that D-data is not written/read to/from the switch bus.

4.7.2.4 Packet switching support

The decentralized switching architecture according to concept 2 has S-transceivers that have implemented an HDLC unit on chip. So low layer-2 functions are established decentralized at the S-side. The configuration as shown in the figure below resembles the centralized switching architecture according concept 2. Here only the architecture using the i83c152 chip is shown.
The architecture as shown above performs p and s-type data splitting at the S-side and routes these separate datastreams respectively to packet bus (using HDLC with collision det/res mechanism) and to serial bus to D-data processor card (using UART with collision det/res mechanism). The packet bus is accessible by all terminals connected to S-transceiver cards where the i83c152 is applied. Furthermore, packets that are transceived via U-transceivers can be placed on the packet bus supported by the HDLC that is placed on the D-data processing card. Layer-1 and switching control is established by a separate processor. Information transfer between the layer-1 and layer-2 and 3 processor is established via a serial line (using UARTs).

Figure 4-23. D-channel processing configuration for decentralized switching architecture concept 2 using i83c152
As described in par. 4.4 this configuration routes p-type data to an packet bus and s-type data to the centralized layer-2 and 3 control card. Similar to the centralized switching architecture using the i83c152 for packet and signalling data splitting, the following can be remarked here. The reduction of processing load caused by the application of this component is an advantageous item. But the remote (related to layer-2 and 3 control card location) performance of this splitting is not recommendable (as indicated in par. 4.4 already). Above that, due to the fact that this configuration is only applied at S-side, the centralized layer-2 and 3 processing card must perform similar actions since packet switched data should also be supported at U-interface reference points. This causes the system to become a bit unbalanced: for the S-side packet switching is supported decentralized while for the U-side this function is performed centralized (the main cause is the lack of U-transceivers with low layer-2 functions implemented on chip). For these reasons the application of this decentralized D-data processing system is not recommendable.

4.8 Architecture review and conclusions

This paragraph terminates the hardware architecture chapter. It provides an overview of the different architectures that have been discussed throughout this chapter. This overview is mainly concerned with the applicability of the architectures on the intelligent NT. It's tested whether the designs meet the requirements as stated in the design targets in par. 4.2. To this end each target is discussed for each architecture.

- **Modularity:** The switching system should be modular. This is one of the design targets stated in par. 4.2. It's an important item since it determines the effectivity of the design. If e.g. in the intelligent NT design several modules are used that are (rather) similar and which number corresponds with the NT's configuration (as U and S transceivers are concerned) the design can easily (and cost-effective) be adjusted to the size that is desired. Checking the implementation on the application of this target in the centralized and decentralized switching architectures as discussed in previous paragraphs, the next remarks can be made.

The centralized architecture is a rather modular structure looking at the transceivers and support modules that can be added to the central switching unit. But when the switching module is taken into consideration it should be remarked that this module handles a fixed number of peripherals (e.g. transceivers) and thus covers a particular range of configurations. Of course, it depends on the size of the module, but usually the size won't fit the configuration size exactly. Or, the system is a bit oversized as switching is concerned. This is no problem unless the cost-effectivity isn't affected. So, one may conclude that the modularity of the switching part of the centralized architecture is rather small and if this part makes up a great deal of the product's price, this lack of modularity is a disadvantage.

In the decentralized switching architecture a large modularity is achieved since DSUs are used. The switching modules applied to each transceiver and thus each intelligent NT configuration corresponds with a fixed number of DSU modules. This is a great advantage
since oversized configurations are prevented while adjusting the system is still supported. Applicability of the DSUs for both U and S sides of the architecture should also be provided when the modules are being manufactured in order to reduce the number of different modules within the design. This will increase the cost-effectivity of the architecture even more. This applicability can be implemented e.g. by means of providing a possibility to set the module in master or slave mode for the timing. The differences between U and S DSUs are described more thoroughly in the previous paragraphs.

Concluding: the decentralized switching architecture shows great modularity which can be made even more effective if DSUs are made applicable to both U and S sides.

- **Busstructure:** the busstructure that is used for the decentralized switching architecture is more complex (according to both concepts) than the one used in the centralized architectures.

As for the centralized architecture there should be made some difference between the architectures according concept 1 and according concept 2. Since in the second concept a small part of the controlling is performed de-centralized (near the S-transceivers with built-in low layer-2 functions) the busstructure is a bit more complex since there is some interprocessor communications (the central MCU and the layer-2 supporting microcontrollers).

The difference in busstructure complexity is caused by the fact that the decentralized architecture has modules that should be controlled for switching and call control purposes by a microcontroller that is centrally located. In the central switching architecture however, this control is performed centrally and only data is transported from the transceivers to the central unit (the board that contains switching and control devices). Especially the concept 1 architecture shows a full centralized control and the busstructure only consists of serial datalines carrying data to and from the central switching unit (possibly multiplexed to reduce the number of datalines too). From manufacturing point of view a simple busstructure is preferable since it reduces wiring and removes problems that are caused by the several (parallel) wires (e.g. crosstalk and interference).

Concluding: the concept 1 centralized switching architecture (and the concept 2 version a bit less) offers a simple busstructure and is most recommendable viewed from manufacturing angle. Especially if transceivers are used that support a multiplexed bus the busstructure becomes very simple.

- **Number of components:** a disadvantage of the decentralized architecture is the complexity of the system cards due to the number of components used. A reduced number of components is preferable from manufacturing point of view. There is less effort required to place components and furthermore there’s less space needed on the cards. If the size of the switching components is concerned the following can be remarked. Though the DSUs are relatively smaller compared to the CSU, they do have more impact on the system’s wiring configuration since the relative number of pins they provide with specific switching and control functions is less than provided by a larger chip (e.g. each chip requires power feeding).

Concluding: the centralized switching architecture has reduced wiring compared to the
decentralized architecture and furthermore has a smaller number of components that should be located on cards. Both arguments are advantageous as manufacturing is concerned and thus should be taken into account.

• **D-data processing (and packet switching support):** it turns out that centralized D-data processing (esp. in case p and s-type data is split) is most preferable. The application of S-transceivers with on-chip HDLC units performing low layer-2 functions don’t offer great advantages: the system becomes unbalanced (U-side doesn’t provide HDLC on U-transceiver chips) and there’s an increase of serial communication between decentralized cards (low layer-2 functions for S-side) and a central card performing layer-2 and 3 processing. In case the D-data is directly routed via a serial (multiplexed) bus to a central card where p and s-type data is being split (and p-type data layer-3 multiplexed to host terminal) and layer-2 (and 3 for s-type data only) is performed, the busstructure becomes more simple and the hardware for D-data processing is located mainly on a central card.

**Concluding:** centralizing the D-data processing (as shown in concept 1 configurations) is most preferable from busstructure- and hardware location-viewpoint.

Based upon the abovementioned results, the centralized switching architecture is preferable to the decentralized. Though the cost-effective modularity rate of the decentralized switching architecture is higher than in the centralized case, manufacturing considerations on the busstructure and number of components cause the centralized switching architecture to be most recommendable from technical design objective. Centralizing the D-data processing hardware (concept 1) makes the busstructure even more simple and locates all hardware supporting this processing on a central position of the system. This makes modifying the system e.g. to a packet switching supporting system quite simple and cost effective.
5. On the feasibility of a common D-channel

5.1 Introduction

This chapter describes the (dis)advantages of a common D-channel architecture that can be applied to the intelligent NT's configuration. The most important motive for the application of such a system is a cost reduction due to component reduction. Since the application of the common D-channel might cause software/hardware changes and/or additions, this has been investigated more profoundly before the decision was made to implement this D-channel processing system.

In order to assure the decision to have a thorough base, it'll be approached from different points of view.

First traffic aspects of the channel will be discussed. Due to the fact that the channel can only be accessed by one TE at once, the probability of waiting-for-access arises. It will be studied how many TEs may share the D-channel while the waiting time for access is still acceptable.

A second aspect which is studied is the layer-3 call procedure protocol. What are the consequences for message handling due to the shared use of the channel.

The last approach of this common channel concept is a study on the impacts it may have on the hardware configuration. Since a physical localization down to S-interface level of TE can't be established by layer-2 procedures (there's one logical D-channel), some hardware has to be added that takes care of the localization.

5.2 Calculation of the acceptable number of terminals that can be serviced by one D-channel

This paragraph is intended to indicate how the performance of the intelligent NT might change due to the fact that packet switched data is transferred via a common (amongst TE) D-channel. To this end, a calculation has been made showing how many voice terminals can share a D-channel and how large this number is when packet terminals are also sharing that D-channel. Above that, the maximum number of packet terminals is calculated that doesn't cause unacceptable D-channel access times.

As already discussed earlier, it's possible to establish a so called common D-channel for TEs connected to different S-transceivers. In this configuration the collision detection/resolution mechanism as it's known at the S-bus is applied to several S-transceivers. Due to the fact that
several data-sources (TEs) are temporarily using one and the same channel via which their data is transferred to an HDLC unit, there exists a chance that a TE has to wait for the D-channel to get idle before accessing it.

There are two kinds of data that are transceived via the D-channel: p and s-type. To start with the last type: s-type data is signalling data for e.g. setting up a call. In that case the info-field of the layer-2 frame (L2-frame) like shown in fig. 4-2, contains signalling data like SETUP or ALERTing. P-type data is packet data contained by the info-field of the L2-frame (see fig. 4-2). The length of these messages is usually longer than signalling messages.

In table 5-1 some traffic statistics are shown indicating what the D-channel load is caused by different terminals. These data are based upon Ref. [5].
<table>
<thead>
<tr>
<th>Type of TE</th>
<th>Circuit switched Voice Calls: the D-channel use by signalling data (s-data)</th>
<th>Packet switched Data Calls using the D-channel: the use of the D-channel by p-type data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE's characteristic parameters</td>
<td></td>
<td>interactive data communications 4800 - 9600 bit/s transaction oriented traffic data communications 2400 - 9600 bit/s file management software development and scientific calculations</td>
</tr>
<tr>
<td>1 Traffic intensity</td>
<td>0.25 $E$</td>
<td></td>
</tr>
<tr>
<td>2 Call duration</td>
<td>150 s</td>
<td>30 s</td>
</tr>
<tr>
<td>3 Calls per second per terminal</td>
<td>0.0017 Calls/s</td>
<td>0.0014 Calls/s</td>
</tr>
<tr>
<td>4 Average frame-length</td>
<td>400 bits/fr</td>
<td>1088 bits/fr</td>
</tr>
<tr>
<td>5 Average frame duration</td>
<td>25 ms/fr</td>
<td>68 ms/fr</td>
</tr>
<tr>
<td>6 L2-frames per call</td>
<td>16 frs/call</td>
<td>6 frs/call</td>
</tr>
<tr>
<td>7 Number of frames per second per terminal</td>
<td>0.0272 frs</td>
<td>0.2 frs</td>
</tr>
<tr>
<td>8 D-channel occupation rate per TE</td>
<td>0.68 ms/s</td>
<td>14 ms/s</td>
</tr>
</tbody>
</table>

**TABLE 5-1.** Traffic statistics concerning D-channel use for voice and data TEs

Remarks on the table:

- How some characteristics can be calculated:

The numbers put in front of the traffic parameter entries are used in the next part of this remark as abbreviations for the traffic parameters.

5-3
\[
\begin{align*}
\text{and finally} \\
\end{align*}
\]

- As data terminals are concerned, their baud rate can be up to 9.6 kbaud. Data communication at higher baud rates can’t be performed via the 16 kbit/s D-channel.

The D-channel occupation rate can be considered as the expectation value for the generation of a layer-2 frame by a TE. This figure will be used to calculate the number of TEs that may use the common D-channel. Comparing the use of the D-channel by s-data and by p-data, it can be determined by looking at the occupation rate that the D-channel use by p-data is about 1.4 - 1.8 % while for s-data it’s only 0.07 % which is considerably smaller (p:s like 26:1).

If \( n \) is the maximum number of terminals that may use a common D-channel, then \( n \times (\text{occupation rate/TE}) \) is the expectation of the total D-channel traffic. Simultaneous activity by two or more sources (collision) should occur only with a 1% probability (i.e.: the probability that a TE has to wait more than the duration of a L2 frame should be less than 1%). Considering the L2 frame generation process as a Poisson process, the according cumulative Poisson table can be used to calculate the maximum number of TEs.

If only s-type data (for voice calls) considered (in case only voice terminals are connected to the common D-channel), the maximum number of terminals turns out to be:

\[ n = 214 \]

So up to 214 voice terminals may share a common D-channel in case the abovementioned conditions are taken into account.

Making such a calculation for a configuration where packet terminals are sharing a D-channel and s-type data generated for setting up a packet switched data call can be neglected compared to p-type data, it turns out that the maximum number of terminals sharing a D-channel conform abovementioned conditions is:

\[ n = 8 \]

So due to the fact that packet terminals generate quite a lot more D-channel data than voice terminals, the maximum number of TEs is reduced considerably. Since the s-data D-channel load is negligible compared to the p-data load, one can also add some voice terminals to the (8) packet terminals (e.g. 8 voice TEs; the p:s ratio remains 26:1 then which allows neglecting the s-data influence on the D-channel).
Ref.[6] shows a relation between D-channel load by p-type data and (voice) call delay caused by D-channel access delay. An acceptable value for this delay is 200 ms. This is based upon the assumption that 10% of the total call delay (i.e. D-channel access, transmission and switching delay) may be caused by D-channel access delay. Since a watchdog timer built in TE exceeds 2s after the D-channel access has been initiated, 200 ms is the maximum acceptable delay caused by D-channel access delay. The load put upon the D-channel by the packet terminals shown in table 5-1 doesn't exceed the 200 ms limit.

The calculation made above shows that a configuration of 8 packet terminals (worst case: file management, software development) and some 8 voice terminals may share a D-channel (and accordingly a single HDLC unit) under condition that the probability that 2 or more L2-frames are generated within an interval during which the D-channel is already occupied by another TE, is less than 1%. Increasing the number of voice terminals causes the calculation to become less exact due to the fact that signalling influence on the D-channel can't be neglected anymore.

The according call delay for voice calls, caused by the intensive channel use by packet terminals is less than 200 ms which is an acceptable value.

For the basic intelligent NT configuration (4S and 2U transceivers) the abovementioned TE-configuration is quite suitable. Two packet terminals for each S-transceiver, completed with one or two voice terminals is an interesting configuration that merely covers the application areas the intelligent NT is intended to be used for.

For typical LAN applications where 8 packet terminals of the type mentioned here isn't sufficient, more dedicated HDLC modules should be applied to each S-transceiver in order to support high data traffic over the D-channel.

5.3 Software development aspects of a common D-channel

Since software development has become very important in the high-tech product development process, it has been investigated what impact the use of a common D-channel has on the intelligent NT's software investments. To this end the differences between the situation where separate D-channels can be determined per S-transceiver (so a maximum of 8 terminals connected to the S-bus circuit are using the same D-channel), and the common D-channel situation (one logical D-channel for all S-transceivers) have been studied.

When the network terminates a call it sends a layer-3 message with a layer-3 address on the D-channel (broadcast on layer-2, TEI 127). In the NT1 situation this message appears at the D-channel belonging to the S-bus circuit to which max. 8 terminals are connected. If a single terminal is supposed to be addressed, a unique layer-3 address is required for that terminal at that S-bus circuit. Otherwise two or more terminals may get alerted by the message which isn't the caller's purpose.
The requirements that are existing for the terminals connected to the S-bus in the NT1 case, are also valid for terminals connected to the several S-busses provided by the intelligent NT. In case separate D-channels are considered, one for each S-transceiver, layer-3 addressing can be compared with the NT1 situation. An address should be uniquely defined for terminals on a single S-bus. But as there are several S-bus circuits concerned in the intelligent NT case, the message should be routed over the right D-channel in order to reach the right terminal. So the layer-3 address together with the S-transceiver identifier, identify each terminal connected to the NT uniquely.

In case a common D-channel is used and the network sends call set-up messages to all terminals, layer-3 addresses should be unique amongst all the terminals in order to be able to reach separate TEs.

In both cases the intelligent NT must have knowledge to what S-bus a specific terminal is connected for the purpose of B-channel allocation. Since layer-2 provides layer-3 with this knowledge it's necessary in both cases that the NT keeps up a list with TEI - S-bus identifier relations.
TE group of terminals connected to the S-bus circuit provided by NTI. Only one D-channel shared these terminals.

NT1 situation

intelligent NT situation with separate D-channels

small groups identify TEs connected to S-bus circuit. These groups share a D-channel over which messages are broadcast. From this point-of-view the collection of small groups can be considered as a large group.

intelligent NT situation with common D-channel

Figure 5-1. Message transport in NT1 and intelligent NT configurations

The common D-channel architecture looks quite advantageous compared to the separate D-channel architecture since there’s no need to send several messages amongst separate D-channels from NT to TE. However, a problem arises when a call should be offered to several terminals; in the separate D-channel configuration these terminals connected to the same S-bus circuit should have equal layer-3 addresses. In that way a call set-up message, containing a layer-3 address is sent on the D-channel and those terminals identified by the layer-3 address and able to accept the call (i.e. not busy with other call yet) are alerted. In case these terminals' addresses are connected to different S-bus circuits, the call set-up messages have to be sent over D-channels of all S-bus circuits. Depending on the B-channel available at each S-bus circuit, either B1 or B2 should be included in the call set-up message.

In the common D-channel configuration we suppose those terminals that should be offered the same call to have equal layer-3 addresses. A call set-up message with the layer-3 address that identifies all terminals to be offered the same call is being sent over the common D-channel. Due to the fact that terminals with identical layer-3 addresses may be connected to different S-bus circuits, it’s impossible to define a B-channel to be allocated for the call (for some
terminals it can be B1 for others B2). Since the call set-up procedure doesn't allow sending two messages containing different B-channels, it can be concluded that terminals in the common D-channel configuration must have different layer-3 addresses even if they should be offered the same calls. In this case, the abovementioned problem doesn't exist. But now there have to be generated separate call set-up messages for each terminal. Since there is only one D-channel to send these messages consecutively, the load put upon this channel is quite high. In the separate D-channel configuration, it's also necessary to send an individual message to each terminal (connected to a different S-bus), but there is made use of the separate D-channels. So the load put upon these channels isn't considerably higher.

Conclusion: it turns out that the apparent advantage of the broadcast to terminals capability of the common D-channel configuration is disadvantageous in the abovementioned case. Therefore it isn't recommendable to apply this common D-channel architecture due to the increment of D-channel load.

5.4 Hardware development aspects of a common D-channel

From hardware development point-of-view, the application of the common D-channel requires some effort. In par. 4.3.3. when the common D-channel architecture is discussed, the hardware implications are pointed at.

It appears that the active D-channel detection mechanism implemented as hardware is a system able to perform zero detection on a two bit nibble using two registers/flip-flops per D-channel. Furthermore, a system should be implemented that establishes the generation of a code indicating what D-channel is the active one (to which S-bus the active terminal is connected to).

The active D-channel detection problem can also be solved as indicated in par. 4.3.3.2. by applying a multichannel HDLC. In that case, if DMA control is used, the DMA channel indicates the active D-channel.

Conclusion: from hardware development (and implementation) point-of-view, the application of a common D-channel configuration is possible but requires some further investigation. The exact implementation and timing requirements are items that should be studied more thoroughly.

5.5 Conclusions

In the previous paragraphs the common D-channel architecture is illustrated from different points-of-view. These objectives are listed here with a summary of the conclusions that are
drawn.

- the number of terminals able to be serviced by a single D-channel/HDLC: in the basic NT configuration with (maximum) 8 packet and 8 voice terminals a common D-channel architecture can be applied without causing access problems at the D-channel.
- in case several terminals should be offered the same call simultaneously, problems arise with respect to B-channel assignment. A solution could be to send multiple call set-up messages but this increases the (single) D-channel load too much and is from call processing point-of-view not recommendable.
- hardware development: a circuit can be designed that performs the terminal localization function. Though the configuration of this circuit should be studied more carefully, its design and implementation don't seem to require an unacceptable amount of efforts.

Based upon these separate conclusions it isn't recommendable to apply a common D-channel configuration to the intelligent NT design.
6. Implementation of the hardware designs establishing the intelligent NT's functional behaviour

6.1 Introduction

This chapter gives a possible implementation of the centralized switching design. As indicated in earlier stages of this report, this design is mostly apt to be implemented using off-the-shelf components. The implementation given here is built up around such a component. A description of the architecture will be given.

The architecture is furthermore considered from manufacturing viewpoint. This gives an impression on how the NT's architecture is configured if the hardware is placed on Eurocards. An estimation on the costs of the architecture will finally end the centralized switching implementation.

The decentralized design's implementation will be discussed less detailed. The hardware design study favoured the centralized architecture above the decentralized which caused the implementation phase to be initiated by investigations after a centralized implementation. However, this chapter yet will provide comment on a possibly applicable busstructure (IST) which may be used for the implementation of the decentralized switching design.

6.2 Implementation of the centralized switching design

6.2.1 Introduction

The implementation phase is the last phase in the intelligent NT's project performed by the author. Based upon the results of the design phase, there's initiated a search after components showing the functional behaviour as defined in the design phase. As concluded in this design phase, the centralized switching design is most favourable for being implemented in the intelligent NT.

A chip that seems to be very useful to perform the CSU (central switching unit) function within the NT is the Mitel 8981D digital switch. It's functional behaviour will be described. Furthermore, the component will be shown being part of the intelligent NT's configuration.

The last part of this paragraph gives some considerations on the configuration being implemented on Eurocards.
6.2.2 Description of the Mitel 8981D digital crosspoint switch [Ref. 8]

This VLSI-CMOS device is a digital time/space crosspoint switch. It can be used for the intelligent NT as an implementation of the CSU functional block (see par. 4.6). The functional diagram of the component is shown below.

![Functional block diagram Mitel 8981D](image)

**Figure 6-1.** Functional block diagram Mitel 8981D

The device's main functions and features are listed here.

- **ST-bus in- and output ports:** The 8981D is equipped with 4 input ports and 4 output ports. In figure 6-1 they are denoted respectively as STi0 - STi3 and STo0 - STo3. The 'ST' characters in these port labels already point at the Mitel ST-bus compatibility of this component.

A detailed description of this bus structure is given in par. 3.3.4. Each in- and output port configures a 2Mbit/s multiplexed datastream. This is established by multiplexing 32 channels having a 64kbit/s capacity each. To this end data is being framed in timeslots of 8 bit each. A frame accordingly contains 32 timeslots of 8 bit each. By means of transceiving this frame each 125µs a channel capacity of 64kbit/s is supported.

The number of ST-bus compatible transceiver chips that can be connected to this chip can
be calculated as follows. Each transceiver occupies 4 channels of the ST-bus. Since each ST-bus provides 32 channels, up to eight transceivers can be connected at each inputport of the chip. The chip supports 32 transceiver chips (128 channels) totally.

- **Time-space divisioned digital switching:** The device is able to switch data contained by any of the input channels (32 per ST-bus). The timeswitching function is the switching between channels of equal in- and output lines (e.g. channel 3 of STi2 to channel 28 of STo2). Space switching is performed if data is switched between different in- and output ports (e.g. channel 4 of STi1 and channel 4 of STo3).

These time- and space switching functions are established by the switch using serial-to-parallel conversion (v.v.) of the muxed input (output) data. All data provided at the input ports is written parallely to the data memory (DM). The size of this memory is 128x8 bits. Accordingly, each memory location is related to a particular input channel. This DM can be read by the microprocessor that controls the chip via the control interface.

The Connection Memory (CM) is split into high and low parts. The locations within this memory are associated with particular ST-bus output channels. Data contained by these output channels can either be switched from an input channel or it can originate from the microprocessor (latter mode is called Message Mode). A CM low location associated with a particular output channel either contains the DM address containing the data of the input channel to be switched to that output channel or the actual data to be switched to the output channel. The contents of a CM low location is output repetitively on the channel once every frame until the microprocessor intervenes.

Data contained by the CM is received via the datalines of the Control Interface (CI). This interface furthermore provides address lines (A0 - A5) and handles the microprocessor control signals Chip Select (CS), DaTa Acknowledgement (DTA), Read/Write (R/W) and Data Strobe (DS). The Control Register controls the mode setting of the switch. It can be accessed by the microprocessor via the address lines.

- **Microprocessor control interface:** The interface towards a microprocessorbus is provided by the Control Interface block of the switch. It contains an 8 bit parallel databus and 6 bit parallel addressing bus. Furthermore it has serial control lines that support the microprocessor control (R/W, DTA, CS and DS). The microprocessor applied should establish switch setting functions and layer-1 control for the transceiver chips. The possibility to set individual channels in message mode without having to re-set each time a frame is transmitted, is a very advantageous (from microprocessor load viewpoint) feature. Some layer-1 supporting bits at the transceiver's system interface need not to be changed for several framecycles; the ability to set this data (only once) for consecutive frames removes processor intervention each time a frame is to be transmitted.

- **Timing:** Timing of the device is established by applying a 4096 kHz dataclock and an 8 kHz framclock to the respective C4i and F0i input pins. As illustrated in the architecture design phase, these clocksignals should be Master Clock signals received from a local timing unit. Units that are added to this switching device should be compatible with the ST-bus specifications. In paragraph 3.3.5 this item is discussed more detailed.

- **Power supply:** This device is powered using the Vdd (+5V) and Vss (ground) input pins. The power consumption of the device is 30mW.
**Chip dimensions:** The package of this chip can be:

- 40 pin Ceramic Dual In Line (CERDIP)
- 40 pin Plastic Dual In Line
- 44 pin Plastic J-lead chip Carrier

**Modularity:** In case double switching capacity should be required, Mitel provides a 256 channel switching device (MT89800) which has an equal number of pins. Pins at the 8981D that have no function are used in the 8980D (e.g. to provide more serial in/output ports).

6.2.3 Description of the architecture representing the centralized switching implementation

The implementation of the centralized switching design with centralized D-data processing (concept 1) is shown in fig. 6-2. This architecture should not be interpreted as straight-a-way manufacturable. It only determines those (main) components that are also shown in the centralized switching design pictures. Not all functional blocks are implemented by actual off-the-shelf components since they require further study and development.
Figure 6-2. Implementation of centralized switching design for the basic intelligent NT configuration (4S, 2U)

The description starts with an overview of the implemented blocks.

- SBC - PEB 2080:
  This Siemens S-transceiver provides a 4-wire S-interface identified in fig. 6-2 by \(Sx_{1/2}\) and \(Sr_{1/2}\) (for transmit and receive lines respectively). Since this component provides also an
IOM system interface, it can be connected to a multiplexed bus conveying 2B + D + C data in a multiplexed way (serial lines SDI and SDO).

The timing of the SBC is taken care of by the local timing generator. The SBC is slaved to this device; the clock signals used are: frame clock (FCS), 8 kHz and a data clock (DCL), 4096 kHz. The input of these signals is labeled in fig. 6-2 by 'IOM-timing'.

In the basic configuration 4 SBC circuits are implemented. They provide 4 S-bus circuits to each of which a maximum of 8 terminals can be connected.

Other implementations for the S-transceiver that have been studied are:

- **MT8930 (SNIC):**
  This Mitel component, though able to perform low layer-2 functions by means of an integrated HDLC, can be used for bare layer-1 S-transceiver functions too. Making use of the integrated HDLC would be an implementation of the concept 2 centralized switching design which is omitted here since it has been shown in chapter 4 that concept 2 designs aren't recommendable. The application of this component in the here discussed architecture should be dissuaded for economical reasons: comparing the prices this can be illustrated.

  Mitel's MT8930 (SNIC): $ 7,80 Siemens' PEB2080 (SBC): $ 6,50

- **APT's STT chip:** This S-transceiver chip designed by APT doesn't support the 2 Mbit/s multiplex bus. In the STT3 design however the application of this chip connected to a 2 Mbit/s (GCI/IOM2) busstructure is foreseen but not implemented yet. So in future, if the GCI interface support is implemented this chip can replace the SBC.

- **IEC - PEB 2090:**
  For the implementation of the U-transceiver there has been chosen for the Siemens IEC chipset. These analog and digital chips transceive dataframes over the DSL towards the central office using a 4B3T code. Since the intelligent NT is primary intended for application on the USA market, the according linecode should be supported. At the time this report was written there was no 2B1Q chip available by Siemens. However, in Ref. [11] a Siemens spokesman announces the PEB 2090 chip for the 2B1Q code to be available in mid-1989. In contrast with the 4B3T version of the PEB 2090, the 2B1Q version will be a single chip implementation.

  The IEC supports an IOM system interface. The SDI and SDO serial datalines, shown in fig. 6-2 transfer 2B + D + C data between the elastic buffer module and the chip.

  In the basic configuration two IECs can be determined. Each provides frame and dataclock signals (FCS and DCL). In order to provide a master clock signal the clock signals of the two IECs are led to a local timing generator (discussed later on).

Other alternatives for implementation studied are:
- Philips LTT chipset (PCB 23901 + 23902): this chipset doesn’t support a 2 Mbit/s multiplexed bus interface and therefore isn’t applicable to the centralized switching design as presented here. Above that it doesn’t support the 2B1Q linecode which causes it to be not useful for USA applications. The AT&T UBAT chipset does provide 2B1Q linecode for transmission on the DSL but it neither supports the 2 Mbit/s multiplexed busstructure as used in this switching design.

- Digital crosspoint switch Mitel 8981D:
  This chip performs the actual switching function to be established within the intelligent NT. Its ST-bus in- and outputs carry 2B + D + C data to and from transceivers. Switching of B-channels is possible between all transceivers connected to the ST-bus. Layer-1 data control is supported by the microprocessor connected via the microprocessor interface.

  Up to 32 transceivers can be connected to the 8981D component when applied in above described way. A maximum configuration of about 10 U-transceivers (20 B-channels towards the network) and 22 S-bus circuits (44 B-channels available for the user) to which between 44 and 172 terminals can be connected.

  Other components that could establish similar functions have also been studied though not as thoroughly as the 8981D. It might be valuable to check whether the application of Siemens’ PBC (2050/2051) or EPIC (2055) is interesting in the intelligent NT’s architecture.

- SPYDER-T: This unit is the implementation of the HDLC block that performs low layer-2 functions on the D-data. This AT&T device embodies 32 HDLC units with DMA control on chip to support fast FIFO-memory datatransfer. The component can be set to demultiplex data from a serial multiplexed line by means of clock and HDLC-channel control. Transmitting data to the multiplexed bus is performed similarly. The device is connected with its serial receive and transmit lines that convey D-data in a multiplexed way, to the STi0 and STo0 lines. In this configuration they directly access the D-channels. However, packet switched datatransfer via a 64 kbit/s B-channel is also supported due to the fact that these channels can also be accessed by this device.

  The functions established by this component can also be established by separate components. In that case 6 HDLC devices and DMA control providing 12 channels should be used plus some hardware that establishes the multiplexed bus access control. However, this isn’t recommendable from several viewpoints:
    - Power dissipation: the SPYDER-T has (compared to separate HDLC devices) a low power-consumption (1 W) which is equal to the power dissipation of a single HDLC device (like Mitel MT8952).
    - Multiplexing: the SPYDER-T supports multiplexed bus access by internal hardware which is easily controlled by the microprocessor that is applied. In case separate HDLCs are concerned, this multiplexed bus access should be supported by external hardware which requires board space and power.
• Board occupation: an implementation that is built up out of six separate HDLCs with a DMA control unit and external hardware occupies a larger board area. This isn’t recommendable since the size of the NT should be kept to a minimum; besides this fact, a large number of components isn’t recommendable from manufacturing viewpoint.

The abovementioned components are actual implementations of functional blocks that can be determined in the centralized switching design. As it turned out during the study that there has to be developed some small architectures that consist of simple analog and digital components (like e.g. gates, flip-flops or transistors). These blocks that require further study to be able to design a detailed architecture, are listed below. Their functional behaviour is described in chapter 3 of this report.

• Elastic buffer module: takes care of frame alignment and adjustment. Therefore both system timing and individual IEC IOM timing should be applied to this unit.

• Local timing generator: this unit derives a system clock from the IOM timing of both IEC devices. It generates a system clock based upon these clock signals. Due to the fact that devices are used that require different electrical timing characteristics, both IOM and ST-bus timing signals are generated. Although they have different electrical characteristics, they must assure synchronous datatransfer within the switching system.

Other blocks (like 'memory', 'processors' and 'UART') are not implemented. This is due to the fact that their implementation is dependent on the capacity required by the software that supports the intelligent NT.

It’s very interesting to have knowledge of the backplane configuration of the intelligent NT, considered as a Eurocard rack system because manufacturing (and thus costs) of the system come into play.

However, this isn’t yet for sure since the blocks that are dashed-outlined in fig. 6-2 are not definitely Eurocards. The placement of components on Eurocards will be discussed later on. The outlined blocks in this figure represent the dashed-lined blocks in fig. 6-2. The ellipses point at the serial busses existing between two blocks.
The busstructures shown in fig. 6-3 will be described now.

- **Busstructure I**: This ellips points at the two serial lines (receive and transmit) that convey the 2B + D + C data transceived at the S and U interfaces. The data rate at these lines is 2 Mbit/s providing 32 timeslotted channels of 64 kbit/s each and allowing 8 transceiver circuits to be connected. In case a second, third or fourth ST in/output combination should be applied (larger intelligent NT configurations) the number of serial datalines is increased accordingly.

- **Busstructure II**: These serial busstructure is concerned with timing. Each U-transceiver has two clock lines (FCS and DCL) that are routed to the local timing generator so four lines exist in the basic *intelligent NT* configuration. This generator derives master clock signals (according to both ST and IOM specs) which causes the existence of four more serial lines.

- **Busstructure III**: These two serial lines convey the D-data that is transceived from the STi0 and STo0 lines. Each extra ST input/output combination at the CSU adds two more lines to this (basic) structure.

- **Busstructure IV**: This single serial line transceives data between the layer-1 & switch setting microprocessor and the layer-2 & 3 microprocessor. The communication is controlled by UART devices. If the *intelligent NT* configuration is extended by adding transceiver circuits, the number of layer-2 & 3 processors may increase. If UARTs are used supporting a multiple access mechanism, the busstructure can still be limited to a
single wire.

So adding up the serial lines, the basic intelligent NT configuration has 13 serial lines at its backplane. This estimation doesn't imply the lines that should be applied for functions that aren't described here (like power feeding) or lines for inter-Eurocard communications.

6.2.4 Manufacturing aspects

This paragraph gives some remarks on the manufacturing aspects of the implementation of the centralized switching design. It will roughly describe how the architecture is configured using Eurocards that eventually will be placed in a rack system. Their interconnection is established using wiring at the backplane of the rack. Refering to fig. 6-2 where the architecture is shown, the number of Eurocards can be estimated now:

[1] S-card: the 4 S-transceiver chips (22 pins) can be placed on a single Eurocard. About half the S-card will remain free for usage. This free space can be used for location of the local timing hardware or for increasing the number of S-transceivers. If the size of the local timing hardware allows it to be located on this free area of the S-card, this should be done.

[2] U-card: two U-transceiver chipsets plus the hybrids and additional components can be placed on a single Eurocard. It depends on the size of the buffer module whether it can be placed on this board too. If not, the use of an extra card (card 6) should be considered where both buffer module and timing are located.

[3] Switch and control-card: the switching device (MT8981D chip plus control hardware) is located on this card. If the intelligent NT should be expanded as number of transceivers is concerned and if the 8981 doesn't offer enough in- and output ports anymore, one could replace the 8981 easily by the 8980D. In the basic configuration this option should be taken care of already.

[4] D-data control card: this card provides the HDLC channels for a single ST-in/output stream (e.g. STi0/STo0) and the according layer-2 & 3 processing devices. Dependent on the application this card may support the packet switching/multiplexing feature. In expanded versions of the NT, it's quite probable that another D-card is necessary (based upon the use of an extra SPYDER and the probable use of an extra Layer-2 & 3 processor).

[5] Power supply card: this subject hasn't been studied but it seems necessary to reserve a special card for this module.

The application of the next card is dependent (as mentioned in the description of card 2 already) on the location of the buffer and timing modules on resp. U and S cards or together on a separate card (card 6).

[6] Timing and buffer module card: this card is used if it might occur that the buffer module can't be placed on the U-card nor together with the timing module on the S-
card. In that case this sixth card should be applied on which these modules are placed. On the S-card some space remains free for the location of extra S-transceivers in case the basic configuration should be expanded.

The basic intelligent NT configuration, offering 4S and 2U transceivers may well be implemented on 5 - 6 Eurocards configured in a rack system and interconnected by serial lines on the backplane.

6.2.5 Price

An indication for the architecture's price can be given here based upon the price of the separate, main components. Due to the fact that some functional blocks haven't been implemented yet (like timing and buffer module) there remains some uncertainty within the price estimation. However, a rough calculation has been made based on a 10K pieces supply and for the year 1989 (when the development of the intelligent NT will be initiated).

Component prices*:

- 4 x PEB2080  
  4 x US$ 6,10 = US$ 24,40
- 2 x PEB2090  
  2 x US$ 33,30 = US$ 66,60
- 1 x SPYDER-T**  
  1 x US$ 97,--- = US$ 97,---
- 1 x MT 8981D  
  1 x US$ 5,40 = US$ 5,40
- 2 x micro (68000, Hitachi)  
  2 x US$ 13,--- = US$ 26,---

US$ 218,40

6.3 Implementation of the decentralized design

* Based upon US 1,--- = DM 1.80

** The SPYDER-T is used instead of the SPYDER-S. This 32-channel version of the SPYDER-S is lower priced and therefore applied here. Above that it provides flexibility for configuration expansion.
6.3.1 The IST-bus structure

The Integrated Services Terminal bus integrates voice and data services in a single communication network that can be used in-house. Up to 31 terminals within a range of 300 meters can be connected to the two-wire bus. An overview of the physical busstructure is shown in the next figure.

![Physical configuration of the IST bus](image)

**Figure 6-4. Physical configuration of the IST bus**

Four calls can be established simultaneously involving 8 terminals. Packet switched data transfer is supported at the same time. Terminals can be connected and disconnected any time. In order to expand the network, a gateway can be created to other networks (like the ISDN).

The control of the protocol on the IST bus is fully decentralized. It's embodied by interface circuits to which terminals are connected. For voice communication, the IST bus offers 8 circuit switched channels with a capacity of 64 kbit/s each. These channels are referred to by 'b1 - b8'. On the bus the channels are multiplexed within an 8 kHz frame structure. Packet switched data transfer is supported by a 64 kbit/s channel called 'bd'. The 'b' and 'bd' channels could be compared to the B and D channels in the ISDN.
Both layer-1 and layer-2 are specified for the 'bd' channel. Access to the channel is controlled by the slotted CSMA/CD* protocol.

The IST Bus Interface circuit (IBI, PCB 2310) embodies the protocols for both 'b' and 'bd' channels.

Ref. [12] contains a description of the electrical characteristics of the bus structure. The frame structure to which the 'b' and 'bd' channels are subjected is shown below.

![Frame Structure Diagram]

* Carrier Sense Multiple Access with Collision Detection

The first channel is a synchronization channel of 32 kbit/s (F) which is applied to synchronize all terminals to the 8 kHz frame. The terminal that transmits in this channel is the master on the bus. If the bus is used locally only (so no gateway to other networks) a distributed algorithm determines which channel becomes master. In case there exists a connection to other networks via a (gateway) terminal, that terminal will be master in order to reach frame synchronization between both networks.

The 'occupied' bit that is sent in each channel (F, b and bd) determines whether a channel is used by a terminal or not.
6.3.2 The IST Bus Interface circuit (PCB2310)

This VLSI CMOS circuit is the component that establishes layer-1 and 2 interfaces for the IST bus. At one side it interfaces to the two-wire IST-bus while at the other side intra terminal interfaces are provided: Terminal HighWay (THW), Subscriber Line Data (SLD) bus and an 8-bit microcontroller with multiplexed address/data lines. The component and its interfaces will be described here.

The interfaces:

- IST-bus: see par. 6.3.3.
- THW: this bus is a three-wire busstructure. It multiplexes 32 64 kbit/s channels in a time slotted way. The wiring is as follows:
  - THDA: THW data I/O 3-state line
  - THCL: THW 2048 kHz clocksignal
  - THSC: THW frame synchronization 8 kHz signal

The IST bus channels b and bd are mapped upon fixed timeslots of the THW. For each b channel in both transmit and receive directions (2x8) and for the bd channel (1) timeslots are reserved on the THW (totally: 17). This mapping is shown below.

<table>
<thead>
<tr>
<th>IST bus transmit timeslots on THW</th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
<th>b6</th>
<th>b7</th>
<th>b8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>16</td>
<td>19</td>
<td>22</td>
<td>26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IST bus receive timeslots on THW</th>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>17</td>
<td>20</td>
<td>23</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IST bus channels (64 kbit/s)</th>
<th>F</th>
<th>bd</th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
<th>b6</th>
<th>b7</th>
<th>b8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>17</td>
<td>20</td>
<td>23</td>
<td>27</td>
<td>28</td>
</tr>
</tbody>
</table>

Figure 6-6. IST - THW mapping

The THW can operate in 3 modes

- Master
- Slave
Monitor

In **Master mode** the IBI provides the THW bus with 2 MHz and 8 kHz clock output signals. In **Monitor mode** the bd channel on the IST bus is monitored and the IBI provides a timeslot assignment signal on the 8 kHz synchronization output. In **Slave mode** the THW synchronization and clock inputs are provided by peripheral circuits.

- **SLD:** this bus is an 8 kHz three-wire bus for on-board B, C(ontrol) and S(ignalling) channel routing. C and S channels are microprocessor accessible via the microprocessor port provided by the IBI. The wiring configuration is represented by the following lines:
  - **SIP:** SLD data I/O 3-state
  - **SCLK:** SLD clock in/output
  - **ESC/SDIR:** 8 kHz synchronization in/output

The pins at the IBI perform either the SLD interface functions or timeslot assignment on the THW. These two modes are selected by the TSA/SLDEN-pin.

- **TSA-mode:** In this mode output strobe pulses are generated to clock data to and from the THW. These pulses cover specific timeslots on the THW in order to establish TSA.
- **SLD-mode:** In this mode the SLD feature switch is selected. B1 and B2 channels of the SLD bus can be switched to any of the THW timeslots and accordingly to IST b channels. The IBI can be set in master or slave mode on the SLD bus. In master mode the SLD bus is provided with clock signals by the IBI whilst in the slave mode a device connected to the SLD bus provides the IBI with clock signals.

- The 8-bits microcontroller bus: This is a 8051 compatible bus.

6.3.3 The IST bus applied to the decentralized switching architecture of the intelligent NT

The IST is in fact an implementation of the decentralized switching architecture. In this paragraph it will be investigated how the intelligent NT is configured and if the specifications stated in the product definition can be answered to.

A block scheme shows a possible intelligent NT implementation using a single IST bus.
Though the IST-bus is able to handle directly terminals without intervention of a layer-2 device (like Siemens ICC) and layer-1 device (like Siemens SBC, APT STT or Siemens IEC, AT&T/Philips LTT), the application of these circuits has yet been scheduled in the above shown figure. The reason to do this is to be able to offer ISDN interfaces as standardized by the CCITT towards the network as well as to the user.

**Blocking:** a blocking probability at the S-bus should be removed by applying up to two terminals using circuit switched B-channels. Blocking at the IST bus as b channel availability is concerned can be avoided by applying a maximum of 4 IBI circuits interfacing to S- or U-transceivers. Since each interface circuit offers a 2B (+D) interface, it should always have 2 b-channels at the IST-bus available. This is guaranteed by applying the abovedescribed configuration.

**Interface circuits:** in this configuration (in contrast with the earlier described central switching implementation), the AT&T and Philips U-transceiver chips and the APT S-transceiver can be applied. This can be advantageous as component independency is concerned.

**Configuration:** this single IST-bus configuration could establish intelligent NT functions. Its
configuration size could be: 1 U and 3 S-interfaces. For residential applications the IST configuration is a quite simple structure.

In contrast with the earlier shown configurations, this system isn’t one that can be accommodated in a centrally located housing. Each interface module* is located at different points of the bus. This bus is a pair of wires placed in the wall of the house the system is installed in. Its length may be up to 300m. The power supply of the system is provided by a unit located at the end of the bus. It possibly could contain the interface towards the network (U).

**Figure 6-8. IST bus installation establishing intelligent NT functions**

**Modularity:** The IST-bus offers possibilities for expansion. In Ref.[13] a configuration is shown where several IST busses are configured within a single switching system. This system can be adjusted in such a way that its performance is conform the intelligent NT’s performance. The figure printed here shows how several IST-busses may operate together controlled by a switch.

---

* an interface module consists here of IBI, layer-2 device, layer-1 device and the according processing capacity
The switch has 2 Mbit 32 channel PCM in- and output ports. Accordingly, terminals connected to different busses can be interconnected. This enables the establishment of an expanded IST bus configuration.

In order to avoid blocking at the IST-bus, one shouldn't connect more than 2 interface modules to a single bus (together with an IBI to interface to the PCM switch). The 2 interfaces at this bus provide 4 B-channels. If these B-channels are to be switched to interface circuits at other IST busses, all 8 b channels at this IST are in use and trying to establish another call using a B-channel is disabled (blocking).

So, if configured amongst other IST-busses interconnected by means of a PCM switch, the IST supports a non-blocking switching system (as circuit switched B-channels are concerned) for a maximum of two ISDN interface circuits (e.g. 2 S-interfaces/busses supporting 2 terminals using circuit switched B-channels).

**Summarizing:** the IST bus offers (at this stage of investigation) capabilities to establish an intelligent NT configuration. Its main features are:

- 4 ISDN interface circuits for a single IST-bus so a 3 S - 1 U configuration is supported providing switching capacity and circuit switched B-channels for up to 6 ISDN terminals.
- A 64 kbit/s datachannel is provided that takes care of signalling and control messages transport but also supports packet switched data traffic.
- centralized power-supply.
• Expandable/modular system. The application of a PCM switching device enables several IST-busses to be interconnected. In that case each IST bus supports up to two ISDN interfaces (so a maximum of 4 terminals per bus plus packet data terminals).

• Interface circuitry not placed in central housing but in separate housings establishing local interfaces between bus and terminals.

It is recommendable to proceed studying on the IST bus as an alternative for the intelligent NT’s hardware configurations described in this report. A price comparison with the centralized implementation can be established then.

6.4 Conclusions

The centralized switching architecture using the Mitel 8981 Digital Switching device for B-channel switching is described. For interfacing towards network and user/terminals, the Siemens chips are most applicable due to their pricing and multiplexed IOM2 interface characteristics. For low layer-2 processing of the data conveyed over the D-channel the AT&T SPYDER-T chip is used. This chip supports expandability of the system due to its availability of 32 HDLC channels. By means of accessing a 64 kbit/s B-channel this component can establish packet multiplexing for packet switched data support. Though this component is expensive (which may change in the future) it's very useful for this application: its multiplexed bus access capability and on board DMA causes it to be more useful than separate HDLC devices which consume more power and require more boardspace. Dependent on the location of buffer and timing modules this architecture can be placed on 5 or 6 Eurocards. The price of the implementation based upon the key-components (transceivers - 2U, 4S -, SPYDER-T, MT8981D and microprocessors - Hitachi 68000 - ) is estimated at U$ 218,40.

A possible implementation of the decentralized design is discussed too. The configuration is based upon the IST bus (Philips). The configuration provides a 64 kbit/s channel for packet switching support. Further study is required in order to be able to compare this architecture with the centralized architecture as e.g. costs are concerned.
7. Conclusions and Recommendations

- For the intelligent NT a centralized switching architecture is preferable to a decentralized one:

  In the centralized architecture all data originating from and destined for the terminals and public network is transferred to a central processing unit using a 2 Mbit multiplexed busstructure.

  In this central unit B-channel switching, D-data control and processing (layer-2 and 3) and maintenance (layer-1) is established. This requires no control - via a busstructure - of remotely located units as in the decentralized architecture. A simple busstructure is advantageous since it reduces manufacturing and communication component (UART) costs.

- The application of a common D-channel configuration is not recommendable:

  In this configuration all terminals connected to different S-buses of the intelligent NT use the same D-channel for signalling (and optionally packet) data transfer. Since a single HDLC unit is necessary for the low layer-2 functions the component costs of this configuration are reduced in comparison with the separate D-channel configuration.

  Traffic calculations for the D-channel occupation by the terminals and considerations on the hardware responsible for the necessary terminal identification at the S-bus circuits show that the common D-channel configuration is realizable for the intelligent NT.

  But the unique layer-3 addressing amongst all terminals which is required by this multiple S-bus configuration and B-channel assignment, forces the call processing software to send multiple messages when one call should be offered to several terminals. Due to the fact that this is an unacceptable situation merely from call processing viewpoint, the common D-channel configuration should not be implemented.

- D-data processing should be performed centralized:

  In case packet data switching is supported, packet (s) and signalling (s) data should be split. P-type data is conveyed via a 64 kbit/s channel using layer-2 multiplexing. S-type data is layer-2 and layer-3 processed. These functions can be performed best at a central location since - compared to a configuration where these functions are performed at different locations e.g. linecards - the busstructure is simpler and the total amount of components is reduced as functions can be integrated.

- The implemented architecture based upon the Mitel 8981D switch responds to the design targets:

  Modularity: the configuration is built up out of modules. This increases system flexibility and expandability. Fast implementation: due to the application of off-the-shelf components the architecture can be implemented and manufactured fast. There is no customized chip design necessary.
Items which require further study in the future are mentioned below:

- PCM adder implementation for establishment of the 3-way calling feature
- an implementation for the tone generator
- an implementation for the local timing generator
- an implementation for the elastic buffer module that supports the master clock generation
- layer-1 maintenance and control support at IOM interface
- microprocessor and peripherals configuration based on the software (layers 1-3) requirements

and

- IST bus implementation in order to be able to give a price comparison between this decentralized architecture and the centralized one
REFERENCES

In: Documents of the eighth Plenary Assembly, 

DEFINITIE AFSTUDEERPLAATS M.J.L. MAGDELIJNS. 
JNL233-B-87316 
Hilversum: AT&T and Philips Telecommunications, IAS department

INTELLIGENT NETWORK TERMINATION. 

[4] Fundneider, Oswald, 
SCHALTUNGSANORDNUNG FUER DEN TEILNEHMERANSCHLUSS IN EINEM 
DIENSTEINTEGRIERENDEN DIGITALNETZ (ISDN) 
Siemens aktiengesellschaft 
publication number: 0 124 859 
European Patent Office

ISDN TRAFFIC ASSUMPTIONS AND REPERCUSSIONS FOR SWITCHING 
SYSTEM ARCHITECTURES. 

PERFORMANCE OF THE ISDN USER-NETWORK INTERFACE FOR 
SIGNALLING AND PACKETIZED USER-DATA TRANSFER. 
Vol. 2, p. A4.5.1-A4.5.5

[7] ISDN ORIENTED MODULAR INTERFACE 2nd GENERATION IOM2 
SPECIFICATION REV. 1.1. 

[8] ST-BUS FAMILY OF ISDN COMPONENTS. 
In: Microelectronic Products Data Book Issue 4 '87-'88. 
P. 5-1..5-361.

[9] DATA COMMUNICATIONS. 
P. 3-261..3-216.
[10] ICs FOR TELECOMMUNICATIONS.  

THE ISDN PUZZLE TAKES SHAPE AS THE PIECES FALL INTO PLACE  

[12] IST BUS SPECIFICATION  
Philips Development Data, February 1988

[13] IST BUS INTERFACE (IBI)  
Philips Development Data, January 1987