290
2-13-78

Sh. 1816

DOE/EV-0006
(Previously announced as TID-27675)

Std.
Plus
Special EHR Distr

Recommendations for CAMAC* Serial Highway Drivers and LAM Graders for the SCC-L2 Serial Crate Controller

Adopted by
U.S. NIM Committee
(Nuclear Instrumentation Methods)
and ESONE Committee of European Laboratories
(European Standards on Nuclear Electronics)

January 1978

U.S. Department of Energy
Assistant Secretary for Environment
Division of Biomedical and Environmental Research

* Computer Automated Measurement and Control

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DOE/EV-0006
(Previously announced as TID-27675)
UC-37

Recommendations for CAMAC* Serial Highway Drivers and LAM Graders for the SCC-L2 Serial Crate Controller

Adopted by
U.S. NIM Committee
(Nuclear Instrumentation Methods)
and ESONE Committee of European Laboratories
(European Standards on Nuclear Electronics)

January 1978

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

U.S. Department of Energy
Assistant Secretary for Environment
Division of Biomedical and Environmental Research
Washington, D.C. 20545

* Computer Automated Measurement and Control

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

RECOMMENDATIONS FOR CAMAC SERIAL HIGHWAY DRIVERS
and
LAM GRADERS FOR THE SCC-L2

Prepared by:

ESONE Committee of European Laboratories

and

NIM Committee of U.S. Department of Energy

ABSTRACT and FOREWORD

This document describes the functional requirements of Drivers for the CAMAC Serial Highway defined in IEEE Standard 595-1976. The description is independent of the implementation, and in particular no assumption is made about the boundary between hardware and software within the Driver.

Topics covered are the user interface, the supporting system services required, demand handling, and a detailed discussion of the message analysis for various levels of error recovery. An appendix describes the recommended features of LAM Graders for use with the Serial Crate Controller Type L2 of IEEE Std 595-1976.

The ESONE Document corresponding to IEEE STd 595-1976 (to which this document is supplemental) is EUR 6100e. The basic CAMAC Specifications are defined in IEEE Std 583-1975 and the corresponding ESONE publication, EUR 4100e (1972).

The ESONE document corresponding to this report is ESONE/SD/02

Key Words

Data Transmission
Electronic Equipment
Computers
CAMAC
Drivers
Data Processing
Interfaces
Standards

CAMAC SPECIFICATIONS AND REPORTS

Title	IEEE Std No.	IEC No.	DOE No.	EURATOM (EUR) No. or ESONE No.
Modular Instrumentation and Digital Interface System (CAMAC)	583-1975*†	516	**	EUR 4100e
Serial Highway Interface System (CAMAC)	595-1976	††	**	EUR 6100e
Parallel Highway Interface System (CAMAC)	596-1976*	552	**	EUR 4600e
Block Transfers in CAMAC Systems	683-1976	††	**	EUR 4100 suppl
CAMAC Instrumentation and Interface Standards***	SH06437*** (Library of Congress No. 76-39660)	-	-	-
Amplitude Analoge Signals within a 50Ω System	-	-	TID-26614	EUR 5100e
The Definition of IML A Language for use in CAMAC Systems	-	-	TID-26615	ESONE/IML/01
Multiple Controllers in a CAMAC Crate	-	-	DOE/EV-0007 (Announced as TID-26617††)	EUR 6500e††
CAMAC Tutorial Articles	-	-	TID-26618	-
Real-Time BASIC for CAMAC	-	-	TID-26619	ESONE/RTB/02
Recommendations for CAMAC Serial Highway Drivers and LAM Graders for the SCC-L2	-	-	DOE/EV-0006 (Announced as TID-27675)	ESONE/SD/02

*Includes supplementary information
**Superseded by corresponding IEEE Standard listed.
***This is a hard cover book that contains IEEE Stds 583-1975, 595-1976, 506-1976 and 683-1976 plus introductory material
†Also American National Standard ANSI/IEEE 583-1975
††In preparation

AVAILABILITY OF DOCUMENTS

IEEE	- IEEE Service Center, 445 Hoes Lane, Piscataway, New Jersey 08854, U.S.A.
IEC	- International Electrotechnical Commission, 1, rue de Varembe, CH-1211 Geneva 20, Switzerland
DOE (TID-Reports)	- National Bureau of Standards, Washington, D.C. 20234, U.S.A., Attn: L. Costrell
EURATOM	- Office of Offical Publications of the European Communities, P. O. Box 1003, Luxembourg
ESONE	- Commission of the European Communities, CGR-BCMN, B-2440 GEEL, Belgium, Attn: ESONE Secretariat, H. Meyer

CONTENTS Page No.

CAMAC SERIAL HIGHWAY DRIVERS

CONTENTS (Continued) Page No.

ILLUSTRATIONS

CONTENTS (Continued) Page No.

TABLE

APPENDIX

ILLUSTRATIONS

CAMAC SERIAL HIGHWAY DRIVERS

1. Introduction

The CAMAC Serial Highway defined in Euratom Report EUR 6100e and IEEE Standard 595 (1976) provides a means of interconnecting up to 62 CAMAC Crate Assemblies (conforming to EUR 4100e and IEEE Standard 583-1976) to a computer. The Serial Highway communicates with the Dataway in each Crate Assembly via a Serial Crate Controller, and with the computer via a Serial Driver. The recommended Serial Crate Controller is Type L2, defined by an appendix to EUR 6100e and IEEE Standard 595.

This paper discusses the functional requirements for Serial Drivers for the CAMAC Serial Highway. It is intended to give some guidelines, especially on analysis of messages received at the Serial Driver Input, and on the internal processing required; these guidelines are a basis for a recommended practice to assist designers of Serial Drivers.

This paper is not intended to define any specific method of implementation or, in particular, the boundry between the hardware and software within a Serial Driver. However, it defines the interface between the Serial Driver and the User Program. This interface also includes facilities needed to drive parallel CAMAC systems.

It is noted that there is a fundamental difference between a Serial Driver (SD) and a Serial Crate Controller (SCC), in that the SD must accept and process all information from the highway, while the SCC only accepts and processes information which appears to be addressed to it. Thus an important feature of the SD is the ability to distinguish between relevant messages and spurious information (e.g. noise or distorted messages).

2. The Serial Driver

In this document the term 'Serial Driver' is used for the link between the User Program and the CAMAC Serial Highway. The Serial Driver is a compound of hardware and software. Obviously the hardware must include the ports to the physical highway, and the software must have an interface to the User Program. The internal boundaries between hardware and software are implementation dependent.

In order to examine the Serial Driver it is regarded as a combination of a Transmitter and a Receiver. These are effectively independent paths for information flow from the User Program to the CAMAC Highway and from the Highway to the User Program. However, some interaction between the transmitting and the receiving parts of the Serial Driver is essential for effective operation.

It should be recognised that the time required to perform a complete Command/Reply transaction, including a Dataway cycle, will generally be longer in a serial system than on a parallel highway. This time depends on the number of crates, logical byte delays, and the byte frequency. This may require some buffering between the user and the Serial Driver (see Fig. 2 and section 6).

Serial Crate Controllers Type L1, conforming to the final version of a preliminary document on the Serial Highway (ESONE document SH/01 and US ERDA TID 26488) are equivalent to type L2 for most practical purposes.

The algorithms described in this paper for the Serial Driver assume:

- that all the Serial Crate Controllers are the defined type SCC-L2, which truncate Commands (see message types, section 4)

- that all modules generate X=1 to indicate that the Command has been accepted (see error recovery using the Delayed Error-bit DERR, section 9.2).

- that old modules conforming to EUR 4100e 1969 (gating the L-signal with Dataway Busy) are not used in conjunction with SCC-L1 (generating Busy during Read Local LAM Pattern). This problem does not occur with SCC-L2, or if SCC-L1 is used with new modules.

3. Messages

The input to the Serial Driver Receiver from the highway is a succession of bits (in bit-serial mode) or bytes (in byte-serial mode). The Receiver must assemble messages from this input, and provide status information relating to the assembly process.

- A message is defined as the sequence of bytes starting with the first non-delimiter byte after a delimiter byte, and ending with the first delimiter encountered subsequently.

- A delimiter is any byte with bit 7=1 and correct parity. Thus any other byte (i.e. bit 7=0 with any parity, or bit 7=1 with wrong parity) is a non-delimiter byte.

Section 3 of the Serial Highway Specification EUR 6100e 1976 (IEEE Std 595-1976) refers to four message types (Command, Truncated Command, Reply and Demand), and these are identified by the contents of the MI-field or their length (see Fig.1). Because errors may occur, the Serial Driver has to handle all messages, including any that differ from the standard types.

4. Message Types

In a Serial Highway System incorporating only the defined Serial Crate Controller SCC-L2 the messages may be divided into the following categories (see also Fig. 3a, 3b, 3c, column 7):

4.1 Demand

This message is an asynchronous request from any SCC. It can occur anywhere in the sequence of messages received by the SD.

4.2 Reply without Read Data Field (Write/Control Reply)

This is the response of the addressed SCC to a Write or Control Command.

4.3 Reply with Read Data Field (Read Reply)

This is the response of the addressed SCC to a Read Command.

4.4 Error Reply (ERR=1)

This may be generated in response to any Command, and indicates a transmission error in the Command received at the SCC, such that the CAMAC operation was not executed.

Since the message indicates that a transmission error has been detected it can be generated either by the SCC to which the transmitted Command message was addressed, or by some other SCC (if there were errors in the Header byte of the command such that byte parity was preserved).

4.5 Complete Command

A Complete Command message is received by the SD if the transmitted Command is passed around the Serial Highway without acceptance at a Serial Crate Controller. Each byte has a byte parity bit. Column parity (SUM byte) is presented in the ninth byte for Write Commands, and in the fifth byte for Read and Control Commands. The remaining bytes (SPACE plus END) have no column parity.

4.6 Truncated Command

This is the Header byte of a Command message followed by an END byte. It has only byte parity. The receipt of a Truncated Command indicates that the Command message has been accepted by an SCC. However, if one WAIT byte within a stream of WAIT bytes is corrupted to a non-delimiter, it appears to be a Truncated Command message if the byte parity is correct.

NOTE:

The Serial Highway system Specification allows SCC's other than type SCC-L2. Thus, when some SCC's accept a Command message, they may retransmit all bytes up to and including the SUM byte. The length of the message received by the SD still indicates whether the Command message has been accepted by an SCC.

4.7 Undefined Message

All other messages which do not fall into the classification given above are labelled "Undefined Message".

5. Organisation of Command/Reply Transactions

Command/Reply transactions and recovery from errors are organised as follows:

5.1 Single Command/Reply Transactions

The normal sequence is that each command message is transmitted after the previous CMD/RPY transaction has been completed. The SD transmits the bytes of the Command message and receives the bytes of the truncated Command message and the expected reply to the command.

A single transaction is completed successfully when the SD has received the expected Write/Control/Read reply message. Demand messages can occur at any time within the sequence of received messages, and the appropriate actions to service the demand can be taken after any individual transaction.

A single transaction is completed unsuccessfully if the SD receives an Error Reply, a Reply message with unexpected Header byte, or a complete Command message. If none of these conditions for completion are satisfied, the transaction is abandoned after a preset time-out period has elapsed since the Command was transmitted.

Immediately after unsuccessful completion of a transaction, or after abandoning a transaction at the end of the time-out period, the SD can take appropriate action to recover from the error or to report it to the user program.

5.2 Bursts of Command/Reply transactions

An alternative method of organising transactions, appropriate to high-performance systems with low error rates, is described in Section 4.6.2 of EUR 6100e (IEEE Std 595). In this method the SD does not wait to receive the reply to one command before starting to generate the next. This document gives an example of one way in which the SD can handle burst transactions. A burst of transactions is completed successfully when the last Command message has been transmitted and the expected number of Reply messages !(Read/Write/Control Replies only) have been received. Demand messages can be received at any time during the burst, but the appropriate servicing action can only be taken after the burst is completed.

A burst of transactions is completed unsuccessfully if the SD receives an Error Reply, or a complete Command, or a Reply with unexpected Header byte. If none of these conditions for completion is satisfied, the burst is abandoned after a preset time-out period has elapsed since transmission of the last Command. In burst transactions, any action to recover from an error or report it to the user program takes place after the end of the time-out period.

The operations within the burst should be such that the whole burst can be repeated if it is not completed successfully, for example they should not include 'Read and Clear' operations.

In both single transactions and bursts of transactions the preset time for time-out depends on the number of crates, the logical byte delays and the byte frequency.

5.3 Error recovery

Error recovery procedures are attempts to achieve the intended effect of single CMD/RPY transactions, or bursts of transactions, which have failed. These procedures may be performed by the SD in a way that is invisible to the User Program.

The analysis of recieved messages supports three means of error recovery:-

- repeating a single command. This simple procedure can only be applied safely when the original command has not been executed, due to errors in transmitting the Command message.

- using the procedures based on the Delayed Error (DERR) status and Re-read Data feature of SCC-L2, see Section 15.6 of EUR 6100e and IEEE Std 595. These can be applied when the original Command has been executed but the Reply message has been corrupted.

- repeating a burst of transactions. The procedures based on the DERR status cannot be applied to bursts of transactions. Some duplication of commands is inevitable, and data may be lost if the burst included operations that read data destructively.

6. Block Diagram of the Serial Driver

The block diagram of the SD (Fig.2) shows the data and status information paths within the Serial Driver. The structure shown is independent of its implementation in hardware or software.

The Serial Driver consists of two parts:

The first part is related to the Serial Highway, and contains:

- the Transmitter
- the Receiver and
- the first stage of Message Analysis

In this part there is no interaction between the transmitting and receiving part. Therefore the first stage of message analysis relies completely on the individual received messages, independently from the transmitted Command and other messages received (analysis independent of context and type of transaction, see Fig. 3 a, 3b, 3c, columns 1 to 9).

The second part can consist, for example, of:

- a Command handler
- a Reply handler
- an Error handler
- a Demand handler
- a Burst handler
- and a System monitor.

In this part there is a close interaction between the transmitting and receiving parts. Therefore the second (final) message analysis of a received message includes the context of transmitted Commands and other received messages. This message analysis including context is described in section 9. As a result of this final message analysis one of the following actions are performed:

- Data and parameter transfer
- Response to Demands
- Recovery from errors
- Discarding garbage.

The block diagram of the Serial Driver also shows the interconnection between the Serial Driver and the User Program via System Services which are system dependent and may contain:

- Queue
- CAMAC Address Check
- Data Buffer Address Check
- CAMAC Device protection against unauthorized user access
- Data and parameter transfer between SD and User Program
- Response to an event representing a LAM or an error.

7. SD Command Handler and Transmitter

The Serial Driver Command Handler (Fig. 6) may operate the Serial Highway with either single Command/Reply transactions or bursts of transactions. The Serial Driver assembles each Command message from the Header to the SUM byte, adds the appropriate number of SPACE bytes and terminates the message by an END byte.

The number of SPACE bytes is implementation dependent. It may be:

- a function of the Command transmitted,
- or a fixed, safe number for the system,
- or an indefinite number, generated until the Reply is received.

The last method of terminating a Command/Reply transaction needs interactions between the receiving and transmitting part of the Serial Driver. But note that upstream crates receive fewer WAIT bytes and thus have fewer opportunities to transmit Demand messages, or to re-synchronise.

NOTE: commands for system reconfiguration (Bypass, Loop Collapse) must set Time-out for more than 100mS and must force the Command Handler to generate SPACE bytes during this time (see section 13).

The Command message may be assembled partly in the Command Handler (e.g. to determine the number of SPACE bytes) and partly in the Transmitter (e.g. to generate parity).

The Serial Driver Transmitter generates WAIT bytes if no Commands are transmitted :

- 3 WAIT bytes minimum are required for switching out each delay buffer in the SCC's. Generation of WAIT bytes by the SD is especially required for SCC's near to the output port of the SD.

- 3 WAIT bytes minimum are required for re-synchronisation of an SCC-L2 operating in bit-serial mode (see section A1.5.1 of EUR 6100e and IEEE Std 595).

In bit-serial mode the byte framing must contain one stop bit. There may be additional pause bits.

8. Message Analysis, Summary and First Stage

Each message received by the SD is first analysed individually, without regard to the context in which it occurs. The intermediate result from this analysis is passed to a second stage which analyses the message in a context including any previous messages occurring in the same transaction. The final decision identifies the message and the action to be taken by the SD.

8.1 Summary

The analysis of messages in single CMD/RPY transactions is shown in basic form in Fig 3a and in an extended form in Fig 3b. The analysis of messages in a burst of transactions is shown in Fig 3c. In these diagrams the decisions associated with normal error-free operations are indicated by shading.

In each case the first stage of analysis is similar and is shown in the left-hand block in the diagrams. Here, characteristics of the message (Cols 1-6) that are independent of its context lead to decisions that identify the message in terms of the types defined in Section 4. These intermediate decisions (Cols 7,8,9) are passed to the second stage of analysis shown on the right-hand side of each diagram.

The basic second stage of analysis for single transactions is on the right-hand side of Fig 3a. In the upper block the decisions from the first stage (Col 10) and two context-independent characteristics of the message (Cols 11 and 12) lead to final decisions that identify the message (Cols 21,22) and indicate the appropriate action to be taken by the SD (Col 23). The lower block shows the very simple treatment of incomplete transactions that are terminated by time-out.

In the absence of errors, Demand messages occurring at any time are identified by the first stage, and the action (to respond to the demand) is indicated by the second stage (Class 1). Truncated Command messages are identified by the first stage and, if they occur within a transaction, are discarded by the second stage (Class 6), which expects that a Reply message will follow. Write/Control/Read Reply messages are identified by the first stage and, if they occur within a transaction and have the expected header byte, lead to second stage decisions that the transaction has been completed successfully (Classes 2a and 3b).

Some errors result in the SD receiving Error Reply messages or Complete Command messages. These are identified by the first stage and, if they occur within a transaction, lead to second stage decisions that the command has not been accepted by any SCC (Classes 4 and 5), so that the command can be repeated. Other errors result in messages during the transaction that cannot be identified by the first stage, or messages occurring outside the transaction. These are discarded by the second stage. A special class of errors results in Reply messages with unexpected header bytes (Classes 2b and 3a), and the second stage indicates that the SD should inform the system monitor and user program. When a transaction is not completed, but is terminated by time-out, the second stage indicates that an undefined error should be reported to the user program.

The extended second stage analysis for single transactions is on the right-hand side of Fig 3b. The analysis leading to the decisions for classes 1-5 is similar to Fig 3a, but the analysis for Classes 6-8 is more detailed and leads to some actions using the Delayed Error (DERR) status and Re-read Data feature of SCC-L2. This extended analysis takes into account not only the characteristics of the current message (Cols 10-13) but also the transmitted command (Col 14) and any previous messages received during the transaction (Cols 15-19). Decisions 6a-7e resulting from one message are taken into account in the analysis of later messages in the same transaction (as columns 15-19).

The second stage analysis for a burst of transactions is on the right-hand side of Fig 3c, which is generally similar to Fig 3a. The main difference is in Col 20. If the actual number of replies is less than the expected number, the normal analysis of Reply messages leads to the decision 'Command executed successfully' (Classes 2x and 3x) and the action to wait for more replies. If the actual number is equal to the expected number, the decision is 'Burst completed successfully' (Classes 2y and 3y). The actions for Classes 4 and 5 do not repeat the command.

8.2 First Stage of Analysis

The SD Receiver checks the framing and performs byte synchronisation (by receiving a WAIT byte) in bit-serial mode, and discards WAIT bytes and other additional Delimiter bytes (see flow diagram Fig. 7). A FIFO buffer of one byte (or more) allows some flexibility between the internal clock and the received byte clock.

The criteria applied to message analysis in the first stage rely only on the content of the received message, and are independent of context (see Fig. 3a, 3b, 3c, columns 1 to 6). These criteria are:

		see Fig.3a, 3b, 3c, column
- Byte parity	(Pb)	1
- Column parity	(Pc)	2
- Message length	(L)	3
- Message identifier	(MI)	4
- Transmission Error bit	(ERR)	5
- Delimiter	(END byte)	6

Truncated Command messages are not fully protected by the Error Detection Code. Identification of these messages is strengthened by testing the Delimiter byte for the defined END byte pattern. The ENDSUM byte of Reply and Demand messages does not have a unique pattern.

For the Complete Command (type 5) the pattern of the Space bytes is not checked because the relevant information in this message is terminated by the SUM byte.

The message analysis performed at this stage is independent of the organisation of Serial Highway operations as individual transactions or bursts. It can be implemented either by inspecting each byte of a message as it is received, or by collecting all bytes of a message in a buffer and then performing the checks. The final decision about the message must be made after the end of the message has been received, to ensure that different implementations come to the same result.

The first stage analysis leads to the decisions shown in Figs. 3a, 3b and 3c, columns 8 and 9. The messages are analysed into the seven types described in section 4.

9. Second stage of Message Analysis

The criteria used for this stage of message analysis include the context of the message relative to the transmitted Command and other received messages.

The message analysis depends on whether the Serial Highway operations are organised as single transactions or bursts of transactions and, in the first case, whether undefined messages are analysed to identify cases where error recovery procedures based on DERR status can be used. Thus, three forms of second stage analysis are described below in sections 9.1, 9.2 and 9.3.

9.1 Single CMD/RPY Transactions, basic analysis

This basic form of second stage analysis does not analyse undefined messages. It supports simple error recovery actions that can be applied when the original command has not been executed, due to errors in transmitting the Command message. It is shown in Fig 3a, and uses the criteria shown in columns 10 to 12:

- Message types
- Transaction status
- Received Header = Transmitted Header

The first and second stage analyses lead to the decisions shown in Fig. 3a, columns 21 and 22:

* Demand message received
* Transaction completed successfully
+ Transaction completed at wrong SCC
+ Command not executed by SCC
+ Command not accepted by any SCC
+ Garbage received
* Truncated Command message received
+ Undefined message
+ Undefined error.

(* normal, + abnormal)

These criteria and decisions are defined in more detail below.

9.1.1 Criteria

The criteria for the basic second stage analysis of single transactions, without further analysis of undefined messages, are based on the following conditions:

- Message Types (Fig. 3a, Col 10)
  These are the seven message types identified by first stage analysis (see Section 4).

- Transaction status (Col 11)
  This condition indicates whether a transaction is in progress (P), or not in progress (N). When a transaction is not completed, the end of the time-out period (T) causes certain decisions to be taken.

- Received Header = Transmitted Header (Col 12)
  This condition indicates whether the Header byte of the received message matches, in all 8 bits, the Header byte of the transmitted Command message. In the analysis of Reply and Truncated Command messages this is evidence that the correct SCC has accepted the Command message.

9.1.2 Decisions

The second stage analysis of single transactions, without further analysis of undefined messages, uses the above criteria sequentially in any order, or simultaneously, and leads to the following decisions, shown in columns 21 and 22 of Fig. 3a:

* Demand message received (Class 1)
  The result of the first stage analysis is accepted, without applying further criteria, and independently of the condition 'Transaction status'. The actions appropriate to the demand are performed after any current transaction has been completed.

* Transaction completed successfully (Classes 2a, 3b)
  A Reply message, identified by the first stage as Type 2 or 3, has been received while a transaction is in progress, and has the expected Header byte. It is classified as the expected reply without applying other criteria, because of the strong protection given by the Error Detection code.

  The status and, in the case of a Read reply, the data are passed to the user program.

+ Undefined CAMAC operation executed (Classes 2b, 3a)
  A Reply message, identified by the first stage as Type 2 or 3, has been received while a transaction is in progress, but does not have the expected Header byte. It can be assumed that either the Reply was corrupted or the Command was executed at a wrong crate. However, the latter is unlikely due to the strong protection given by the error protection code.

  The user program and System Monitor are informed.

+ Command not executed by SCC (Class 4)
An error Reply message, identified by the first stage as Type 4, has been received while a transaction is in progress. This indicates that the Command message has been accepted by an SCC (it is irrelevant whether this is the correct SCC) and not executed, because of a transmission error in the Command message.

The transaction can be repeated, with a limit on the number of repetitions.

+ Command not accepted by any SCC (Class 5)
A Complete Command message (type 5) has been received while a transaction is in progress. This indicates that the Command message has not been accepted by any SCC, but this may be due to temporary conditions such as lack of synchronism or a transmission error in the Command message. It is unnecessary to analyse the reason for the failure.

The transaction can be repeated, with a limit on the number of repetitions.

* Truncated Command message received (Class 6)
A Truncated Command message (Type 6) has been identified by the first stage while a transaction is in progress. No further second stage analysis is performed and the Truncated Command message is discarded (but see Section 9.2.2 for analysis and actions when undefined messages are being processed).

+ Undefined message received (Class 7)
An Undefined message (Type 7) has been identified by the first stage while a Transaction is in progress. The Undefined message is discarded (but see Section 9.2.2 for analysis and actions when Undefined messages are being processed).

+ Undefined error (class 8)
At the end of the time-out period the first stage analysis has not identified a Reply message or a Complete Command message, or no message at all has been received.

There is insufficient evidence to show whether the command has been executed (but the reply lost), or has not been executed (eg. the command message lost). The user program is informed that the transaction has not been completed.

+ Garbage received
All messages, except Demands, received when no transaction is in progress are analysed as garbage, and are discarded.

A certain amount of garbage is expected depending on noise in the environment. It is recommended that the System Monitor should be informed if an abnormal amount of garbage is received.

(* normal, + abnormal)

9.2 Single CMD/RPY Transactions, extended analysis

This optional extended form of second stage analysis performs further processing of Truncated Command messages and undefined messages. It supports further error recovery procedures based on the Delayed Error (DERR) status which can be applied when the command has been executed but the Reply message has been corrupted. It is shown in Fig.3b.

If the analysis of the received messages does not show whether the Command has been executed or not, a recovery procedure can be attempted in which the Delayed Error bit gives this information. The use of DERR is safe only if there is a reasonable indication that the original Command has addressed the intended SCC, so that the state of the DERR-bit corresponds to the current transaction.

The required indication can be obtained if all messages within the time-out period are inspected and classified. The only indications at the SD that a Command has been accepted by an SCC are the receipt of the Truncated Command and/or the Reply message.

An error recovery procedure using the DERR bit must not be performed more than once, because the recovery operation (Read Status or Reread) may change the state of the DERR bit in the SCC. (The DERR bit corresponds to the last operation in the SCC).

If in a Read Status or Reread operation the DERR bit is found to be DERR = 1, the original Command has not been executed and it can be repeated.

If the Reply message to this repeated command is again an undefined message of type 8a-d this indicates that the message has been accepted by the intended SCC - although doubtful whether executed or not. In this case it is correct to perform another Read Status or Reread operation. This sequence Command - Reread/Read Status - repeat original Command - Reread/Read Status - ... is correct but has to be limited.

The extended second stage analysis uses the additional criteria shown in Fig. 3b columns 13 to 17 while the transaction is in progress:

- Received message length (L)
- Read Command transmitted
- Expected Truncated Command received
- Corrupt Command received
- Corrupt Reply received

The first and extended second stage analyses lead to the more detailed decisions of class 6 and 7 shown in Fig. 3b in the columns 21 and 22 while the transaction is in progress:

* Expected Truncated Command received
+ Other message with L=2
+ Corrupt Write/Control/Error Reply received
+ Corrupt Read Reply received
+ Corrupt Command (not accepted by any SCC)
+ Other message with L > 2 .

(* normal, + abnormal)

The extended second stage analysis uses the additional criteria shown in Fig. 3b in column 18 and 19 when the transaction has been terminated by Time-out:

- Other message with L = 2
- Other message with L > 2

The first and exetended second stage analysis lead to the additional final decisions when the transaction is completed by Time-out shown in Fig. 3b in the column 21 and 22:

+ Reply lost
+ Corrupt Read Reply (expected) received
+ Corrupt Write/Control/Error Reply (expected) received
+ Corrupt Complete Command received
+ Undefined Error.

(+ abnormal)

The additional criteria and decisions are defined in detail in Sections 9.2.1 and 9.2.2 .

9.2.1 Criteria

The additional criteria for extended second stage message analysis are based on the following conditions:

- Received message length (Fig. 3b, Col 13)
  The length of the undefined message is used to classify it as Command Form (length equal to that of the transmitted Command message) or Reply Form (length 3 or 7 bytes) or Other message with L = 2.

- Read Command transmitted (Col 14)
  The transmitted Command included a Read Function, so that a 7-byte Reply or an Error Reply are expected.

- Expected Truncated Command received (Col 15)
  The messages received while the transaction is in progress have already included one classified as Class 6a (see Section 9.2.2).

- Corrupt Command received (Col 16)
  The messages received while the transmission is in progress have already included one classified as class 7c (see Section 9.9.2).

- Corrupt Reply received (Col 17)
  The messages received while the transmission is in progress have already included one classified as class 7a or 7b (see Section 9.2.2).

- Other messages with L=2 received (Col 18)
  The messages received while the transaction is in progress have already included one classified as Class 6b, 6c or 7d (see section 9.2.2) in place of, or in addition to, the expected Truncated command.

- Other message with L > 2 received (Col 19)
  The messages received while the transaction is in progress have already included one or more with length more than two bytes that have either not had the appropriate length for Command Form or Reply Form, or are in addition to another Command Form or Reply Form message (Class 7e).

9.2.2 Decisions

The first and extended second stage analyses for single transactions lead to the following additional decisions while the transaction is in progress, shown in Fig. 3b in columns 18 and 19.

* Expected Truncated Command received (Class 6a)
The first stage analysis has identified a Truncated Command message (Type 6) while the transaction is in progress, and the Header byte matches that of the transmitted Command message. This is evidence that the Command message has been accepted by the correct SCC. This is used as a condition in the analysis of undefined messages (see column 15) when the transaction is completed by Time-out.

+ Other message with L=2 received (Classes 6b, 6c, 7d)
The first stage analysis has identified a Truncated Command message while the transaction is in progress, and this either does not have the expected Header or is an additional Truncated Command with the expected Header. In case of class 7d the first stage has identified an Undefined message with a length of two bytes. This is evidence of certain types of malfunction, and is used in the analysis of undefined messages (see Column 18) when the transaction is completed by Time-out.

+ Corrupt Write/Control/Error Reply received (class 7a)
The first stage analysis has identified an Undefined message (Type 7) while the transaction is in progress and it is the first received 3-byte message where the header matches.

+ Corrupt Read Reply received (Class 7b)
The first stage analysis has identified an Undefined message (Type 7) while the transaction is in progress and it is the first received 7-byte message where the header matches and a Read Command has been transmitted.

+ Corrupt Command received (Class 7c)
The first stage analysis has identified an Undefined message (Type 7) while the transaction is in progress and it is the first received message where the length is equal to the transmitted Command message length.

+ Other message with L > 2 received (Class 7e)
The first stage analysis has identified an Undefined message (Type 7) while the transaction is in progress and this message cannot be further classified into class 7a to 7d.

(* normal, + abnormal)

The first and extended second stage analysis for single transactions lead to the following additional decisions when the transaction is completed by Time-out shown in Fig. 3b in the lower sections of columns 21 and 22:

+ Reply lost (Classes 8a, 8b)
The expected Truncated Command message has been received, but the time-out period has elapsed without a good Reply having been received. No other message with L = 2 must have been received, in order to make sure that the expected Truncated Command message received is the only one which can be interpreted as such. It is reasonable to assume that the command has been accepted but the Reply has been lost or corrupted. The error recovery procedure described in EUR 6100e (IEEE Std 595) section 15.6 can be used, and the appropriate action is to read the Status Register of the SCC if the Command was Write or Control (Class 8a) and to Reread the data if the Command was Read (Class 8b).

+ Corrupt Write/Control/Error Reply received (class 8d)
The only Undefined message with L > 2 received while the transaction is in progress has been a Reply Form. The expected Truncated Command has not been received. The evidence suggests that the Undefined message is a corrupted Reply in response to a Write/Control command. The appropriate error recovery action is to read the Status register of the SCC.

+ Corrupt Read/Error Reply received (Class 8c)
As the preceding decision, but a Read command has been transmitted so that the evidence suggests a corrupted Read Reply. The appropriate error recovery action is to Reread the data from the SCC.

+ Corrupt complete Command message received (Class 8e)
The only Undefined message with L > 2 received while the transaction has been in progress is a Command Form. The expected Truncated Command has not been received. The evidence suggests that the Command has not been accepted by any SCC. The appropriate error recovery action is to repeat the transaction.

+ Undefined Error (Classes 8f and 8g)
The transmission has been completed by Time-out and the recieved messages do not correspond to classes 8a to 8e, or no messages at all have been received. There is insufficient evidence to show whether the Command has been executed or not. The user program is informed that the transaction has not been completed.

(+ abnormal)

9.3 Bursts of transactions

The second stage of message analysis for a burst of transactions (Fig. 3c) is generally similar to the basic analysis of single transactions (section 9.1), but error recovery is restricted to repeating the whole burst. Recovery based on the DERR status cannot be used, bacause the DERR status of each transaction is overwritten by any subsequent transaction to the same SCC. Recovery by repeating particular failed transactions is not recommended because the effect of a transaction may depend on its context within the burst. Thus bursts of transactions can only be used when these restrictions can be accepted.

9.3.1 Criteria

In addition to the conditions in section 9.1.1 there is the condition:

- Expected number of replies received (Fig. 3c, Col 20)
This indicates whether the number of Read/Write/Control Replies received (including the current message, if appropriate) is equal to the total number of commands in the burst.

9.3.2 Decisions

The first and second stage analyses of messages in a burst of transactions lead to the following decisions, as shown in Fig. 3c, Columns 21 and 22:

* Demand message received (Class 1)
As in Section 9.1.2, but the demand is serviced after the burst has been completed.

* Command executed successfully (Classes 2x, 3x)
The received Reply message does not complete the expected number of replies. Wait for more replies.

* Burst completed successfully (Classes 2y, 3y)
The received Reply message completes the expected number. The status and data from all replies in the burst are transferred to the user program.

+ Undefined CAMAC operation executed (Classes 2z, 3z)
The Header byte of the received Reply message does not match that of the corresponding transmitted Command message. The burst of transactions is abandoned. After the time-out period has elapsed (so that all messages resulting from the burst have been received) all received messages except Demands are discarded, and the burst can be repeated.

+ Command not executed by SCC (Class 4)
  An Error Reply has been received. The burst of transactions is abandoned, and can be repeated after the time-out period.

+ Command not accepted by any SCC (Class 5)
  A complete Command message has been received. The burst of transactions is abandoned and can be repeated after the time-out period.

* Truncated Command message received (Class 6)
  All received Truncated Commands are discarded, and not analysed.

+ Undefined message received (Class 7)
  All received Undefined messages are discarded, and not analysed.

+ Not enough replies (Class 8)
  The time-out period has elapsed before the expected number of replies has been received. The burst of transactions is abandoned, all received messages except Demands are discarded, and the burst can be repeated.

+ Garbage received
  All messages, except Demands, received when no Transaction is in progress are analysed as garbage and are discarded.

(* normal, + abnormal)

10. Demand handling

The Demand Handler should provide a secure way of handling LAM's by means of simple calls from the user program. The details of LAM handling at the hardware level should be hidden from the user, but the facilities provided will depend on the hardware and software support available (for example whether or not a LAM Grader is used, operating system facilities for task scheduling etc.). The following sections define the operations of the Demand Handler in various system configurations.

The actions of the Demand Handler upon receiving a Demand message from the first stage of analysis are:

- to identify its source
- to respond to the Demand
- to inform the corresponding user program that the demand has occurred.

A LAM Grader provides the first level of LAM handling by hardware. Systems with LAM Graders permit:

- fast identification of the LAM source (e.g. by coded station number in the SGL-field, comparable to a vectored interrupt),

- high system security because standard hardware functions are defined for masking individual LAM's. The Demand Handler itself can thus have the ultimate responsibility for handling LAM's.

However, for some systems the cost of LAM Graders is not justified. For systems without LAM Graders:

- the user program has the ultimate responsibility for handling all the LAM's in the system,
- LAM servicing is less efficient because the system can only disable Demands at the crate level, hence a LAM locks out other LAM's from the same crate until it is serviced,
- An additional Read operation is required for LAM identification in the Crate (comparable to a polling procedure).

The effect of a LAM locking out other LAM's until it is serviced can be avoided if the Demand Handler knows the individual Disable function code for each LAM. This allows the LAM to be disabled and the crate demand to be re-enabled immediately by the Demand Handler (see 10.2 below).

10.1 Identification of the source of a Demand

For systems with LAM Graders, the identification is greatly simplified if the bit pattern in the SGL field of the Demand message represents the station number of the module generating the LAM. This facility can be provided by the recommended LAM Grader (see Appendix A).

Systems without LAM Graders require one additional READ operation (Read Local LAM Pattern, see WARNING below) to the appropriate SCC to obtain the same station number information.

In both systems (with or without LAM Graders) further identification of the source of the LAM within the module (LAM at a subaddress, LAM at a bit position, or other non-standard possibilities) can be performed at the system level only by means of tables in the Demand Handler, giving the further access procedure at the module level. The entries in this table can be conveyed from the user program by declaration statements. If this facility is not incorporated in the Demand Handler, the user program itself must perform a search within the module.

WARNING:

Modules conforming to EUR 4100e 1969 (gating the L-Signal with Dataway Busy) must not be used in conjunction with SCC-L1 (generating Busy during Read Local LAM Pattern). This problem does not occur with SCC-L2, or if SCC-L1 is used with new modules.

10.2 System response to a Demand

In systems with LAM Graders using the LAM mask register, the system response is to clear a bit in the LAM Grader mask register, corresponding to the SGL number in the Demand message, thus masking that particular LAM. The LAM Grader can be enabled immediately, so that a new Demand message from a different LAM in the crate can be initiated (see Table 1, mode 1), or Demand message initiation could be enabled after the user has serviced the LAM to avoid nesting interrupt service routines (see Table 1, mode 2).

For some applications the mask operation can be omitted (see Table 1, mode 3); however, a long interrupt service routine may lead to generation of a Hung Demand. Such a Hung Demand must not interrupt the current interrupt service routine, to avoid servicing the same LAM twice (see 10.7 below).

If the Demand Handler finds that no user program is assigned to service the LAM, the LAM should be masked out at the LAM Grader, and an 'unexpected LAM' error message should be generated.

In systems without LAM Graders, the Crate Demand must be disabled until the identified LAMs are cancelled (disabled or cleared) in order to avoid the generation of Hung Demands (see section 10.7). This problem arises because an SCC has only one DMI-line (see EUR 6100e, 1976 / IEEE Std. 595, 1976 section 14.2.4) which may remain activated due to other LAMs which occur before the original LAM has been cancelled.

The disadvantage of disabling the Crate Demand until the corresponding user program has cancelled the LAM (see Table 1, mode 4) can be avoided by disabling the LAM in the module by means of a procedure known to the Demand Handler. This allows the Crate Demand to be re-enabled immediately, and a new Demand message to be generated if further LAM's are present (see Table 1, mode 5). The Demand Handler must know (for example by declarations from the user program) how to disable the LAMs at module level. Modules which do not have a disable LAM feature cannot be serviced by this standard disable procedure in the Demand Handler, so in this case the user program must clear the LAM and re-enable the Crate Demand.

Systems without LAM Graders can only handle unexpected LAM's by disabling all demands from the offending crate. Recovery from this condition is only possible by manual intervention.

In systems with LAM Graders only the expected LAM's are unmasked.

10.3 Demand dispatching to the corresponding user program

After the Demand Handler has responded to the LAM so that it does not generate a Hung Demand, the appropriate information has to be passed to the corresponding user program. This can be accomplished via entries in a system table indicating for example the setting of event flags or the activation of user tasks.

10.4 Re-enabling Demand Message Initiation

After servicing a LAM the user program issues a request to re-enable demands from that LAM source.

In systems with LAM Graders, the response to this request depends on the mode of operation . If the mask register is used, the Demand Handler must set the appropriate mask bit (see Table 1, mode 1 and 2). If Demand nesting is not allowed (see 10.2 above) the Demand Handler must also re-enable Demand message initiation (see Table 1, mode 2). If the mask register is not used, the Demand Handler need only re-enable Demand message initiation (see Table 1 , mode 3).

In systems without LAM Graders, the Demand Handler may simply re-enable demands in the appropriate SCC, or it may read the Local LAM Pattern. If this is zero, it re-enables crate demands; if it is not zero, it proceeds as if a Demand message had been received (see Table 1, mode 4).

Alternatively, if the system performs the disable LAM function at the module level as described in 10.2 above, the response to the enable demand request is to re-enable the LAM in the module (see Table 1, mode 5). The crate demand will have been re-enabled in response to the Demand message.

10.5 Connection/Disconnection between a LAM and an Interrupt Service Routine

For correct operation, the user program must declare the connection between a LAM and the Interrupt Service routine before the LAM occurs. This allows the Demand Handler to distinguish between expected (declared) and unexpected (undeclared) LAM's.

The connect statement contains the LAM address, and the actions which the Demand Handler must perform on receipt of the corresponding Demand messages (e.g. for some implementations the function codes for Disable/Enable LAM). A disconnect statement from the user program declares the corresponding LAM as 'unexpected'.

10.6 Explanation of the flow charts 11a to 11c

Three examples of implementations are given in the flow charts 11a to 11c.

The flow chart 11a assumes the use of the recommended LAM Grader (see Appendix A). It offers security since the system has ultimate control of all LAM signals (only those LAMs which are declared in the user program are unmasked).

Systems using LAM Graders can re-enable Demand message generation in two ways:

- As soon as the corresponding bit in the LAM Grader is cleared, the mechanism for generating further Demand messages is enabled;
- Or further Demand message generation is disabled until the user has given the request to re-enable Demand message initiation after the LAM is serviced (see 10.2 above).

The first method allows the generation of further Demand messages from the same crate, even when the original LAM has not been serviced. The second method avoids nesting of interrupt service routines for one crate.

The flow chart 11b is for systems without LAM Graders. To maintain system security the user must ensure that all possible LAM's in a Crate can be serviced, since an unexpected LAM would lock up the whole LAM handling mechanism in a Crate.

The security of the system represented by the flowchart 11c also depends upon the user program servicing all possible LAM's. It is similar to the system represented by 11b, with the additional feature that the Demand Handler disables LAM's directly.

10.7 Hung Demand

The Hung Demand message mechanism is provided to recover a lost Demand. This can occur if a Demand message becomes corrupted, when there cannot be any response to that Demand by the Demand Handler. After a predetermined time, the LAM Grader or the SCC initiates a Hung Demand Message indicating that some LAM is "hanging".

Problems can occur if Hung Demands are not handled correctly. In the following example a LAM is serviced twice - once as the result of handling a Hung Demand and again as the result of a normal Demand message. When LAM 'A' occurs this initiates a Demand Message 'A', but the response to this is too slow and so a Hung Demand message is initiated before the Demand Handler masks LAM 'A' and re-enables Demand message initiation. In the meantime a second demand LAM 'B' has occurred, and Demand Message 'B' is initiated when DMI is re-enabled. The Demand Handler processes this Hung Demand by reading the local LAM pattern, which includes the bit corresponding to LAM 'B'. This information is passed to the user program, which services LAM 'B'. Later, the Demand Handler receives Demand message 'B' and so the same demand is processed again. This problem can be avoided if the action in response to a Hung Demand does not depend on reading the local LAM pattern.

10.8 Recovery of lost Demand messages

For handling Demand messages via a LAM Grader, the following features are proposed:

When a normal Demand message (with GL-field < 31) has been generated by an SCC no further normal Demand messages will be initiated by the LAM Grader (described in Appendix A.5) until the initiation of further Demand messages has been re-enabled. The response to a Demand message is to clear the corresponding bit in a LAM mask register (thus disabling the LAM which has caused the Demand message) and to re-enable the initiation of further Demand messages. If an unmasked LAM is present a new Demand message is initiated. Disabling the LAM and enabling further Demand messages may be combined in one CAMAC command or they may be two different CAMAC commands from the Demand Handler to the LAM Grader (see section 10.6 and Appendix A.5).

If a LAM initiates a Demand message and is then not disabled within a predetermined time (given by a Repeat Timer) a Hung Demand message is initiated and a "Hung Demand Transmitted" flag (HDT) is set automatically in the LAM Grader. This flag inhibits the initiation of further normal Demand messages. Only additional Hung Demands may be initiated by the Repeat Timer.

The HDT flag is cleared by a command from the Demand Handler to the LAM Grader (see section 10.6 and Appendix A.5), thereby re-enabling the generation of normal Demand messages. If an unmasked LAM is present, a new Demand message is initiated. BUT NOTE: if the HDT flag is not set when the 'clear HDT flag' command is received, this command must not have any influence on the Demand message initiation. This is because more than one Hung Demand may be initiated in sequence, causing more than one Clear HDT flag instruction from the Demand Handler, but only the first of these clears the flag, and initiates a new normal Demand message if a LAM is present. (If the following clear HDT flag instructions are not ignored, this normal Demand message will be repeated illegally).

SUMMARY:

The interaction of LAM Grader and Demand Handler is described in section 10.6. If a normal Demand message is lost, the corresponding LAM is not disabled a Hung Demand is initiated. The Demand Handler clears the HDT flag, re-enabling Demand messages. Thus the Demand message corresponding to the "hanging" LAM is repeated.

If a Hung Demand is lost in addition, the Hung Demand is repeated after a time determined by the Repeat Timer.

If a Hung Demand is generated due to overload of the computer, without loss of the normal Demand message, clearing the HDT flag enables the initiation of the next Demand message only, since the unserviced Demand will precede the Hung Demand in the demand queue. Therefore the corresponding LAM will have been masked before the Hung Demand is serviced (see Table 1, mode 1 and 2).

In systems not using the LAM mask register the generation of Hung Demands due to computer overload must be avoided, otherwise the occurrence of one LAM can be transferred to the user program twice. Therefore the user program must have high enough priority to respond to a LAM in time (see Table 1, mode 3).

For handling Demand messages without using LAM Graders the recovery of lost Demand messages is performed by acting on a Hung Demand message in the same manner as on a normal Demand message. No problem of Hung Demands due to overload of the computer arise if the Demand Handler disables the LAM in the module using knowledge of the specific function code (see Table 1, mode 5). In systems where the Demand Handler does not disable the LAM in the module the Demand Handler must have high enough priority to disable the Crate Demand in time (see Table 1, mode 4) to avoid Hung Demands due to computer overload, otherwise the occurrence of one LAM can be transferred to the user program twice.

TABLE 1: PRINCIPLES OF HANDLING INDIVIDUAL DEMANDS

FACILITIES AND MODE OF OPERATION ACTIONS AFTER A DEMAND/HUNG /DEMAND	MODE 1	MODE 2	MODE 3	MODE 4	MODE 5
	WITH LAM-GRADER			WITHOUT LAM-GRADER	
	MASK REGISTER USED BY OPERATING SYSTEM		MASK REGISTER NOT USED BY OPERATING SYSTEM	ONLY CRATE DEMAND DISABLE	DISABLE/ENABLE LAM AT MODULE LEVEL
	DEMAND MESSAGE INITIATED IMMEDIATELY BY DEMAND HANDLER	SUBSEQUENT DEMAND MESSAGES DEFERRED UNTIL THE USER HAS SERVICED THE CURRENT LAM			
ACTION FOR LAM SOURCE IDENTIFICATION, BY DMD HANDLER	NO ACTION NEEDED (IDENTIFICATION IN DEMAND MESSAGE)			READ LOCAL LAM PATTERN	
RESPONSE TO HUNG DMD, BY DMD HANDLER	UNLOCK LAM GRADER BY XEQ COMMAND			DISABLE CRATE DEMAND	DISABLE CRATE DEMAND + DISABLE LAM IN MODULE + ENABLE CRATE DEMAND
RESPONSE TO A REGULAR DEMAND, BY DEMAND HANDLER	MASK THE LAM IN LAM GRADER + REENABLE DEMAND INITIATION	MASK THE LAM IN THE LAM GARDER	—		
DISPATCH DMD TO USER BY DEMAND HANDLER	EG. : SET EVENT FLAG, ACTIVATE TASK				
USER INTERRUPT SERVICE	EVENTUAL SEARCH FOR THE LAM IN THE MODULE, IF NEEDET + CLEAR LAM IN MODULE				
REENABLE DMD. MESSAGE INITIATION BY REQUEST FROM USER	UNMASK THE LAM IN LAM GRADER	UNMASK THE LAM IN THE LAM GRADER + REENABLE DMD MESSAGE INITIATION	REENABLE DMD MESSAGE INITIATION	REENABLE CRATE DMD	REENABLE LAM IN MODULE

11. System Services

In addition to the user (application) program and the Serial Driver, facilities referred to here as the 'System Services' are required. They may be parts of an operating system, or could be partly incorporated in the user program in applications without an operating system. The System Services are implementation dependent, therefore no detailed descriptions are given, only principles are discussed.

The following are the main facilities already listed in section 6.

- Queue

To provide the flexibility of asynchronous input/output together with the execution of normal instructions, the system must contain an algorithm to manage queues where device handler requests are buffered.

- CAMAC Address Check

In the case of CAMAC, it is good practice to perform address checks to increase system reliability as far as possible. In systems with static addresses at run time the address check could be done during assembly. In systems that allow dynamic address modification, these checks must be performed at run-time, at the expense of execution speed of the program.

A reasonable address check is a plausibility check of the CAMAC address, i.e. whether the CAMAC C,N,A,F lies in the permissible range.

- Data Buffer Address Check

Another check is to test whether the user buffer address lies in the appropriate data space (memory protection facility).

- CAMAC Device Protection Against Unauthorized User Access

To enable the System Services to block illegal accesses to CAMAC devices, the user program must declare the CAMAC space belonging to its corresponding process.

- Response to an event representing a LAM or an error

On the occurrence of a LAM or an error the System Services pass the appropriate information to the corresponding user program. The association between the LAM and the service program can be static, or it can be changed dynamically by a statement in the user program (connecting/disconnecting a LAM or an error with/from its interrupt service).

NOTE: If a user program stops or is aborted, all declared entities must be disconnected.

- System error

The system should incorporate a general routine to handle system errors. For example, in the case of a fatal error a message could be printed on the operator console, and the Serial Driver inhibited from processing further CAMAC requests.

- Data and parameter transfer between the Serial Driver and the user program

In some systems, address parameters passed from the user program must be mapped into physical hardware addresses.

12. User Interface

This section summarises the communication (data and parameter paths) between the user program and the system software, represented in Fig. 2 as System Services. This interface depends on facilities provided in the System Services and in the Serial Driver as well as in the user program, represented by the appropriate language. The aims of the following descriptions are not to define any specific language semantics or syntax, but to give guidlines for implementations.

12.1 Declaration facilities

For CAMAC device protection against unauthorized user access it is necessary to declare the used CAMAC address space for each user program. For user programs which want to respond to LAM's or errors (from the transactions performed by the SD) it is necessary to declare the correlation between a LAM or Error and the service routine in the user program. For the identification of the LAM origin in cases where LAM Graders are used, it is necessary to declare the Crate address and the GL-number. In systems without LAM Graders the crate address and the station number must be declared. For those systems not using LAM Graders but disabling LAM's in the module it is necessary to declare the specific Disable and Enable LAM function code.

12.2 User Requests

A user request is either for data transfer between CAMAC devices and the user program, or for the execution of control functions in CAMAC devices. The user request may contain the following parameters:

- CAMAC device address
- CAMAC function
- Write data
- Memory location address for Read data
- Memory location address for transaction status
- Type of data format representing the data in the CAMAC device and in the memory location
- Type of transaction (single CMD/RPY or burst of transactions)
- Reconfiguration Command (see Section 13.2)
- LAM address/origin (used for Demand message initiation after LAM service)

All parameters can be transfered directly or indirectly by a memory location address (pointer to parameter or to a parameter field if appropriate). A burst of transactions requires parameter fields:

- CAMAC device address (can be reduced to one parameter)
- CAMAC function (can be reduced to one parameter)
- Write data
- Memory location address for Read data
- Memory location for transaction status

The Block Transfers (explained in Supplement to EUR 4100e, IEEE Std 683) can be executed by means of preprocessing in a Block Transfer Handler, which transfers the user request for the Block Transfer into either a series of single CMD/RPY transactions or into a burst of transactions. A Block Transfer can be transformed into a burst of transactions in cases where the Block Transfer can be repeated, and a burst handler is implemented in the SD.

13. System Initialisation

System initialisation must only be performed under the control of the SD, otherwise user requests to initialise parts of the serial system could lead to fatal errors.

In principle the system initialisation can be divided into two parts:

- the 'cold start' of the whole serial CAMAC system after power on, and

- system re-initialisation requests to restart individual external processes from the user program or the operator.

13.1 Initialisation from a cold start

After power-on a sufficient number of WAIT bytes must be generated in order to synchronise the SCC's and the receiving part of the SD.

The initialisation procedure for the Serial system is based on an individual initialisation of each crate known to the SD. This knowledge can be conveyed either by predefining a number of crates with fixed crate addresses, or by a dynamic declaration procedure e.g. a summary of all crate addresses given by an interactive communication with the operator or by a system configuration file.

The initialisation procedure from a cold start for each crate depends on whether or not the system is equipped with LAM Graders.

The first Command sent to the appropriate SCC will set it into an unbypass condition, and the second will set all relevant status bits in the SCC status register. Two separate Commands are recommended because EUR 6100e or IEEE Std 595 only permits operation on the other bits of the status register when restoring the unbypass state (see EUR 6100e or IEEE Std 595 Section 12.4.2).

This setting of status bits in the SCC is performed with Write group 2 (F17 N30 A0) and data. For the unbypass Command the data is equal to zero. For the second Command the data depends on whether or not the system is equipped with LAM Graders.

In systems with LAM Graders, data 403 (octal) represents:

- Generate Z (masks all LAM's in the LAM Grader)
- Generate Clear
- Inhibit off
- Enable Demand
- Loop Collapse
- Dataway on-line

In addition to the Commands to the SCC, the Demand Message Generation at the LAM Grader as described in Appendix A must be enabled by ENB (F26 at A1).

In systems without LAM Graders, data 3 (octal) is as above except for:

- Disable Demand

13.2 Request For Initialisation

For some applications it is useful to allow a partial re-initialisation of the Serial system on user or operator requests, which means that an individual external process is restarted or that the Serial Highway configuration is changed dynamically (e.g. for the maintenance of peripherals).

System reliability can only be guaranteed when re-initialisation or reconfiguration (bypass, loop collapse) requests are supervised by the system (see Sytem Services, Section 11) which means that individual users or programs can only access their declared process peripherals.

In cases where different user programs share one crate, system reliability can only be guaranteed when the execution of dangerous requests is trapped (e.g. Crate Initialise).

NOTE:

When system reconfiguration commands (reset bypass, set loop collapse) are performed, the SD must wait more than 100ms before transmitting the next Command, and must generate a sufficient number of SPACE bytes (see EUR 6100e or IEEE Std 595 sections 12.4.2 and 12.4.3). For example, the reconfiguration requests from the user can be flagged (see section 12) to enable the SD to change the number of SPACE bytes and the time-out. Commands for system initialisation are handled like normal single CMD/RPY transactions. In cases when an initialisation request leads to an undefined error (see section 9) it should be signalled as a system error.

It should be noted that, in any case, spurious errors occur during the CMD/RPY transaction which performs the system reconfiguration, as a consequence of the relay switching. Therefore, in cases of undefined error as the result of these transactions, the easiest way of recovering is to read the status register of the SCC to obtain information about its current state.

14. Interaction between Serial Driver Tasks

The Serial Driver flow chart (Fig. 5) describes the three basic states of the Serial Driver: Within single CMD/RPY transactions, bursts of transactions and outside transactions. The different simultaneous actions of the SD are represented by independent tasks, which are

- Transmitter
- Receiver
- Command Handler
- Demand Handler
- Reply Handler
- Error Handler
- Burst Handler

It shows the conditions under which the different entry and exit points of the specific Handler (given in more detail in the flow charts) are selected.

It is assumed that the Transmitter generates WAIT bytes if it sends no commands.

The sequence for error-free single CMD/RPY transactions is indicated by the heavy lines, starting at the top right-hand corner of the diagram.

The Command Handler (see Fig. 6) is entered if the previous decision did not require a response to a demand, and if a new request has been issued by the user program, and if this request calls for a single transaction. The appropriate Command message is transmitted, and the SD-receiver (see Fig. 7) processes the Truncated Command message. The sequence leaves the SD-receiver at exit 1 and enters the first stage of analysis (see Fig. 8). This gives the intermediate decision 'Truncated Command message', exit 1, and the second stage of analysis is performed in the Error Handler (see Figs. 10a and 10b). This gives the final decision at exit 3 to wait for the Reply message. The SD-receiver processes the received Reply message, exit 2, and the first stage analysis is entered again. This time the decision is 'Read/Write/Control Reply', exit 4. The second stage analysis is in the Reply Handler (See Figs. 9a and 9b) and gives the decision 'Transaction Completed Successfully', and the sequence returns from exit 1 to the beginning.

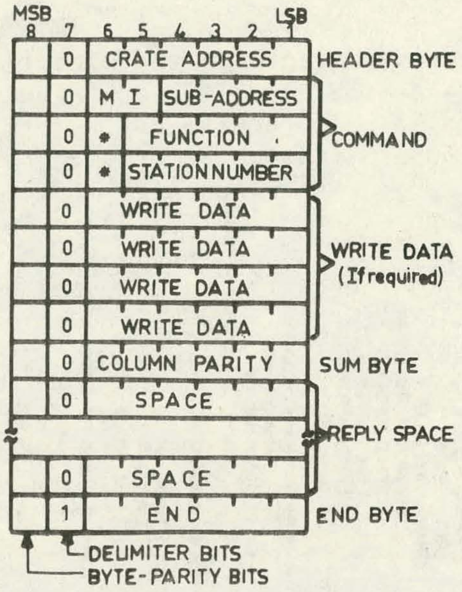


Fig.1-1 Command Message :Field Assignments
* Reserved Bits
MI Message Identifier field

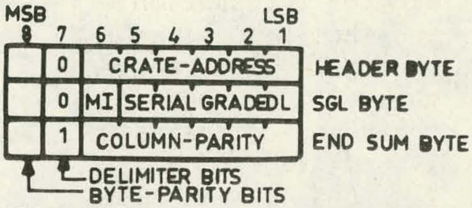


Fig 1.4 Demand Message :Field Assignments
MI Message Identification Field

Message	MI-field	
	M2	M1
Command	0	0
Reply	0	1
Demand	1	–

TABLE 1.5

Contents of Message Identification Field

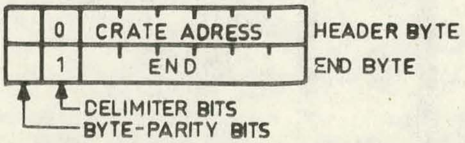


Fig.1.2 Truncated Command Message : Field Assignments

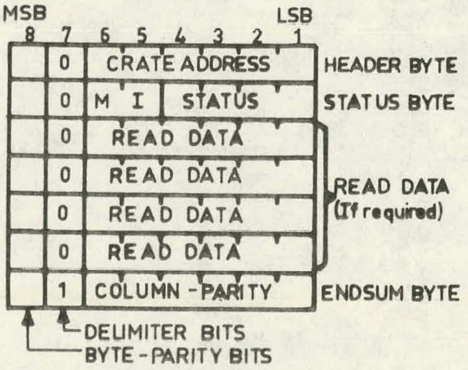


Fig 1.3 Reply Message :Field Assignments
MI Message Identier field

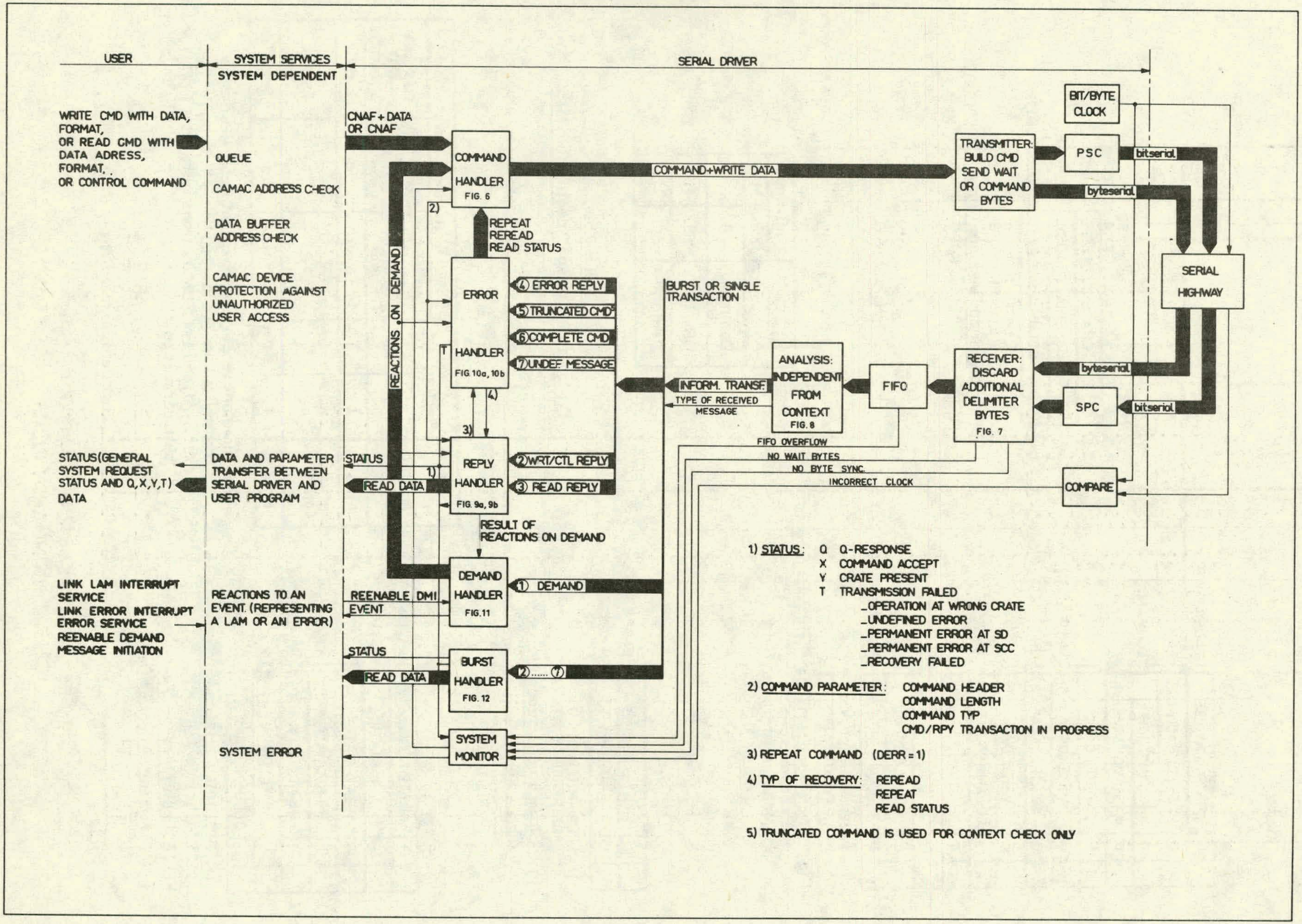
Operation	Function Field		Number of Bytes		
	F16	F8	COMMAND from Header to SUM inclusive	REPLY from Header to ENDSUM inclusive	COMMAND/ REPLY TRANSACTION
Read	0	0	5	7	12*
Control	0 1	1 1	5	3	8*
Write	1	0	9	3	12*

* Minimum length, assuming that Reply Header is transmitted by SCC as first SPACE byte is received, and ENDSUM is transmitted as END byte is received.

TABLE 1-6

Length of Command/Reply Transactions

Fig.1 : SERIAL HIGHWAY MESSAGE TYPES



1) STATUS: Q Q-RESPONSE
X COMMAND ACCEPT
Y CRATE PRESENT
T TRANSMISSION FAILED
_OPERATION AT WRONG CRATE
_UNDEFINED ERROR
_PERMANENT ERROR AT SD
_PERMANENT ERROR AT SCC
_RECOVERY FAILED

2) COMMAND PARAMETER: COMMAND HEADER
COMMAND LENGTH
COMMAND TYP
CMD/RPY TRANSACTION IN PROGRESS

3) REPEAT COMMAND (DERR = 1)

4) TYP OF RECOVERY: REREAD
REPEAT
READ STATUS

5) TRUNCATED COMMAND IS USED FOR CONTEXT CHECK ONLY

Fig. 2: BLOCK DIAGRAM OF A SERIAL DRIVER

FIRST STAGE								
CRITERIA						RESULT		
current message								
PARITY					(END-BYTE)		transf.to2.stage	
P_b	P_c	L	MI	ERR	DELIMITER	DECISION	TYPE	CTH. INFORM.
1	2	3	4	5	6	7	8	9
OK	OK	3	1-	-	-	DEMAND	1	Crate Addr. SGL-Field
OK	OK	3	01	0	-	WRITE/ CONTROL REPLY	2	Crate Addr. Field, Status-Field
OK	OK	7	01	0	-	READ REPLY	3	Crate Addr. Status-Field Read Data
OK	OK	3	01	1	-	ERROR REPLY	4	-
OK	OK	-	00	-	-	COMPLETE COMMAND	5	-
OK	-	2	-	-	YES	TRUNCAT. COMMAND	6	Crate Address Field
MESSAGE DOES NOT MATCH CRITER.						UNDEFIN. MESSAGE	7	Crate Addr. Field, Length

SECOND STAGE													
CRITERIA											RESULT		ACTION TO BE TAKEN
current message				☒									
TYPE FROM 1ST STAGE	TRANS-ACTION STATUS	RECVD. HEAD.= TRANSM. HEADER									DECISION	CLASS	
10	11	12	13	14	15	16	17	18	19	20	21	22	23
I											DEMAND MESSAGE	1	Respond to DEMAND after Transaction has been terminated
2	P	YES									TRANSACT. COMPLET. SUCCESSFULLY	2a	Transfer Status to User.
		NO									UNDEFINED CAMAC OPERATION EXECUTED	2b	Inform System Monitor
3	P	NO										3a	
		YES									TRANSACT. COMPLET. SUCCESSFULLY	3b	Transfer Status and Data to User
4	P	-									COMMAND NOT EXECUTED BY ANY SCC	4	Repeat Command #
5	P	-									COMMAND NOT ACCEPTED BY ANY SCC	5	
2-7	N										GARBAGE		Discard and Wait for more messages
6	P	-									TRUNCATED COMMAND	6	
7	P	-									UNDEFINED MESSAGE	7	

AFTER TIME-OUT: ALL MESSAGES DURING TRANSACTION					
T	-		UNDEFINED ERROR	8	Transfer Status to User

Legend see page 40

Fig. 3A: BASIC MESSAGE ANALYSIS FOR SINGLE COMMAND/REPLY TRANSACTIONS

Showing the second stage analysis without further analysis of Undefined Messages

FIRST STAGE								
CRITERIA						RESULT		
current message								
PARITY					(END-BYTE)		transf.to2.stage	
P_b	P_c	L	MI	ERR	DELIMITER	DECISION	TYPE	OTH. INFORM.
1	2	3	4	5	6	7	8	9
OK	OK	3	1-	-	-	DEMAND	1	Crate Addr. Field, SGL-Field
OK	OK	3	01	0	-	WRITE/ CONTROL REPLY	2	Crate Addr. Field, Status-Field
OK	OK	7	01	0	-	READ REPLY	3	Crate Addr., Status-Field Read Data
OK	OK	3	01	1	-	ERROR REPLY	4	-
OK	OK	-	00	-	-	COMPLETE COMMAND	5	-
OK	-	2	-	-	YES	TRUNCAT. COMMAND	6	Crate Addr. Field
MESSAGE DOES NOT MATCH ABOVE CRITERIA						UNDEFIN. MESSAGE	7	Crate Addr. Field, Length

SECOND STAGE														
CRITERIA											RESULT		ACTION TO BE TAKEN	
current message				☒	previous message during transaction									
TYPE FROM 1ST STAGE	TRANS-ACTION STATUS	RECVD. HEAD.= TRANSM. HEADER	RECVD. MESSG. LENGTH	READ CMD TRANSM.	EXPECT. TR.CMD MESSG. RECVD.	CORRUPT CMD RECVD.	CORRUPT RPY RECVD.	OTHER MESSAGE L = 2	L > 2	-	DECISION	CLASS		
10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1											DEMAND MESSAGE	1	Respond to Demand after Transaction has been terminated	
2	P	YES									TRANSACT. COMPLET. SUCCESSFULLY	2A	Transfer Status to User	
		NO									UNDEFINED CAMAC OPERATION EXECUTED	2B	Inform System Monitor	
3	P	NO										3A		
		YES									TRANSACT. COMPLET. SUCCESSFULLY	3B	Transfer Status and Data to User	
4	P	-									COMMAND NOT EXECUTED BY ANY SCC	4	Repeat Command #, +	
5	P	-									COMMAND NOT ACCEPTED BY ANY SCC	5		
2-7	N										GARBAGE		Discard	
6	P	YES			NO						EXPECT.TRUNC. CMD	6A	Remember Decision (condition for col..) and Wait for more messages	15
					YES						OTHER MESSAGE WITH L = 2	6B		18
		NO			-							6C		
7	P	YES	3	-		-	NO				CORRUPT W/C/E-RPY	7A		17
			7	YES		-	NO				CORRUPT READ RPY	7B		
		-	=CMD	-		NO	-				CORRUPT COMMAND	7C		16
		-	2	-		-	-				OTH.MESSG.: L = 2	7D		18
		ABOVE CRITERIA NOT SATISFIED									OTH.MESSG.: L > 2	7E		19

AFTER TIME-OUT: ALL MESSAGES DURING TRANSACTION											
T		NO	YES	NO	-	NO	-		REPLY LOST	8A	Read Status +
		YES	YES	NO	-	NO	-			8B	Re-read +
		YES	NO	NO	YES	-	NO		CORRUPT R/E-RPY	8C	
		NO	NO	NO	YES	-	NO		CORRUPT W/C/E-RPY	8D	Read Status +
		-	NO	YES	NO	-	-		CORRUPT COMPL. CMD	8E	Repeat CMD # +
		ABOVE CRITERIA NOT SATISFIED							UNDEFINED ERROR	8F	Transfer Status to User +
		NO MESSAGE RECEIVED								8G	

Legend see page 40

Fig. 3B: EXTENDED MESSAGE ANALYSIS FOR SINGLE COMMAND/REPLY TRANSACTIONS

Showing the second stage analysis with further analysis of Undefined Messages, leading to error recovery actions

FIRST STAGE									SECOND STAGE													
CRITERIA						RESULT			CRITERIA											RESULT		ACTION TO BE TAKEN
current message									current message				☒	previous message during transaction								
PARITY					(END-BYTE) DELIMITER	DECISION	transf.to2.stage		TYPE FROM 1ST STAGE	TRANS-ACTION STATUS	RECVD. HEAD.= TRANSM. HEADER								EXPECT. NUMBER OF RPY'S RECVD.	DECISION	CLASS	
P_b	P_c	L	MI	ERR			TYPE	OTH.INFORM.														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
OK	OK	3	1-	-	-	DEMAND	1	Crate Addr. Field, SGL-Field	1											DEMAND MESSAGE	1	Respond to DEMAND after Burst
OK	OK	3	01	0	-	WRITE/ CONTROL REPLY	2	Crate Addr. Field, Status-Field	2	P	YES								NO	COMMAND EXECUT. SUCCESSFULLY	2X	Wait for more Replies
																			YES	BURST COMPLETED SUCCESSFULLY	2Y	Transfer Status and Data to User
											NO								-	UNDEFINED CAMAC OPERATION EXECUT.	2Z	Abandon, Wait for Time-Out,Repeat Bu
OK	OK	7	01	0	-	READ REPLY	3	Crate Addr. Status-Field Read Data	3	P	YES								NO	COMMAND EXECUT. SUCCESSFULLY	3X	Wait for more Replies
																			YES	BURST COMPLETED SUCCESSFULLY	3Y	Transfer Status and Data to User
											NO								-	UNDEFINED CAMAC OPERATION EXECUT.	3Z	Abandon, Wait for Time-Out,Repeat Bu.
OK	OK	3	01	1	-	ERROR REPLY	4	-	4	P	-								-	COMMAND NOT EXECUTED BY ANY SCC	4	Abandon, Wait for Time-Out, Repeat Burst
OK	OK	-	00	-	-	COMPLETE COMMAND	5	-	5	P	-								-	COMMAND NOT ACCEPTED BY ANY SCC	5	
									2-7	N										GARBAGE		Discard
OK	-	2	-	-	YES	TRUNCAT. COMMAND	6	Crate Address Field	6	P	-								-	TRUNCATED COMMAND	6	Discard
MESSAGE DOES NOT MATCH CRITER.						UNDEFIN. MESSAGE	7	Crate Addr.Field, Length	7	P	-								-	UNDEFINED MESSAGE	7	

AFTER TIME-OUT: ALL MESSAGES DURING TRANSACTION						
T	-		NO	NOT ENOUGH REPLIES	8	Abandon, Wait for Time-Out, Repeat Burst

Legend see page 40

Fig. 3C: MESSAGE ANALYSIS FOR BURSTS OF COMMAND/REPLY TRANSACTIONS

Column		
1	P_b	Byte parity
2	P_c	Column parity
3	L	Message length
4	MI	Message identifier
5	ERR	Transmission Error bit
6	END Byte	Delimiter
7	DECISION	Result of first stage message analysis
8	TYPE of message transferred to second stage	
9	Other information transferrred to second stage	
10	TYPE of current message transferred from first stage	
11	Transaction status :	
	P	Transaction in progress
	N	Transaction not in progress
	T	Transaction terminated by Time-out
12	Received Header = Transmitted Header ? (of current message)	
	YES/NO	Criteria applied
	-	Criteria not applied
13	Received message length ? (of current message)	
14	Read Command transmitted ?	
15	Expected Truncated Command message received ?	
16	Corrupt Command received ?	
17	Corrupt Reply received ?	
18	Other message with length L = 2	
19	Other message with lenght L > 2	
20	Expected number of Replies received ?	
21	DECISION	Result of second stage message analysis
22	CLASS of message as result of analysis	
23	ACTION to be taken after message analysis	
	#	indicates "with limit on number of repetition"
	+	not appropriate if Error Recovery in progress
	Shadowed fields : Result of message analysis when no error occurred	

Legend to Fig. 3a, 3b and 3c

FROM STATUS FIELD OF REPLY

X	Command-Accepted bit
Q	Q-response bit
ERR	ERR-bit (indicates a transmission error of the Command)
DERR	DERR-bit (indicates the ERR-bit form the previous transaction)

FROM READ FIELD OF REPLY

DSX	Delayed Command-Accepted bit
DSQ	Delayed Q-response bit

ERROR HANDLER PARAMETERS

RCV	Error Recovery (1 = active, 0 = not active)
RCVTYP	Type of recovery Command (1 = Reread, 0 = Read Status Command)
HDRREC	Expected Truncated Command received
HDRFOR	Other message with L = 2
RPYFOR	Corrupt Reply (Reply Form)
CMDFOR	Corrupt Command (Command Form)
UWMES	Other message with L > 2

STATUS OF TRANSACTION

Y	Crate present
T	Transmission failed

Key to the flow charts

REPEAT COUNTER

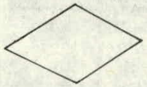
IREC	Error-Reply Counter
ICCC	Complete Command Counter
IRDC	Repeat "DERR = 1" Counter

TRANSACTION PARAMETERS

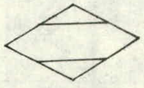
CMDTYP	Type of transmitted Command : 1 = Read Command 2 = Write Command 3 = Control Command
CMDHDR	Header byte of the transmitted Command
CMDLNG	Number of bytes in the transmitted Command
NCMD	Actual number of transmitted commands in the burst
LGTH	Number of commands in the burst
NRPY	Actual number of received replies in the burst
Time-out Clock 1	- Monitoring the Transaction
Time-out Active	- transaction in progress
Time-out Clock 2	- Monitoring the system status

RECEIVER PARAMETERS

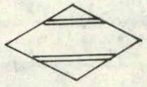
DLMTR	Indication of the type of the previous byte (1 = delimiter byte, 0 = non-delimiter byte)
BYTSYN	State of byte synchronism (0 = no byte synch, >0 = byte synch (No. of received bits)
L	Number of bytes in the received message
Pb	Byte parity
Pc	Column parity
MI	Message identifier
RECHDR	Header byte of the received message



Represents a decision made on the actual set parameter



Represents a decision made on the previous set parameter



Represents a decision made on the transaction parameter

Fig.4: LEGEND TO Fig.3A,3B,3C AND KEY TO THE FLOW CHARTS.

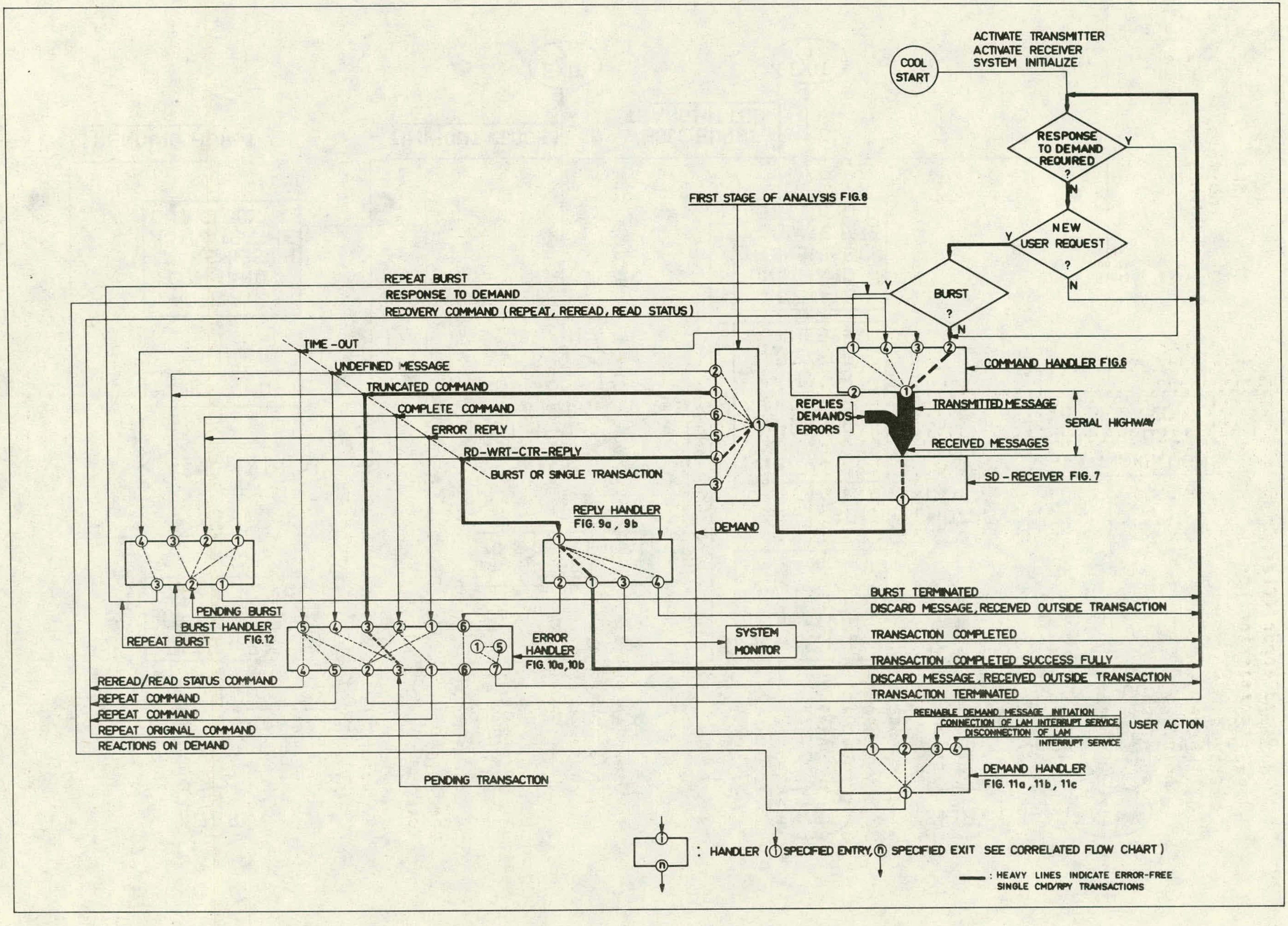


Fig.5 : SERIAL DRIVER FLOW CHART

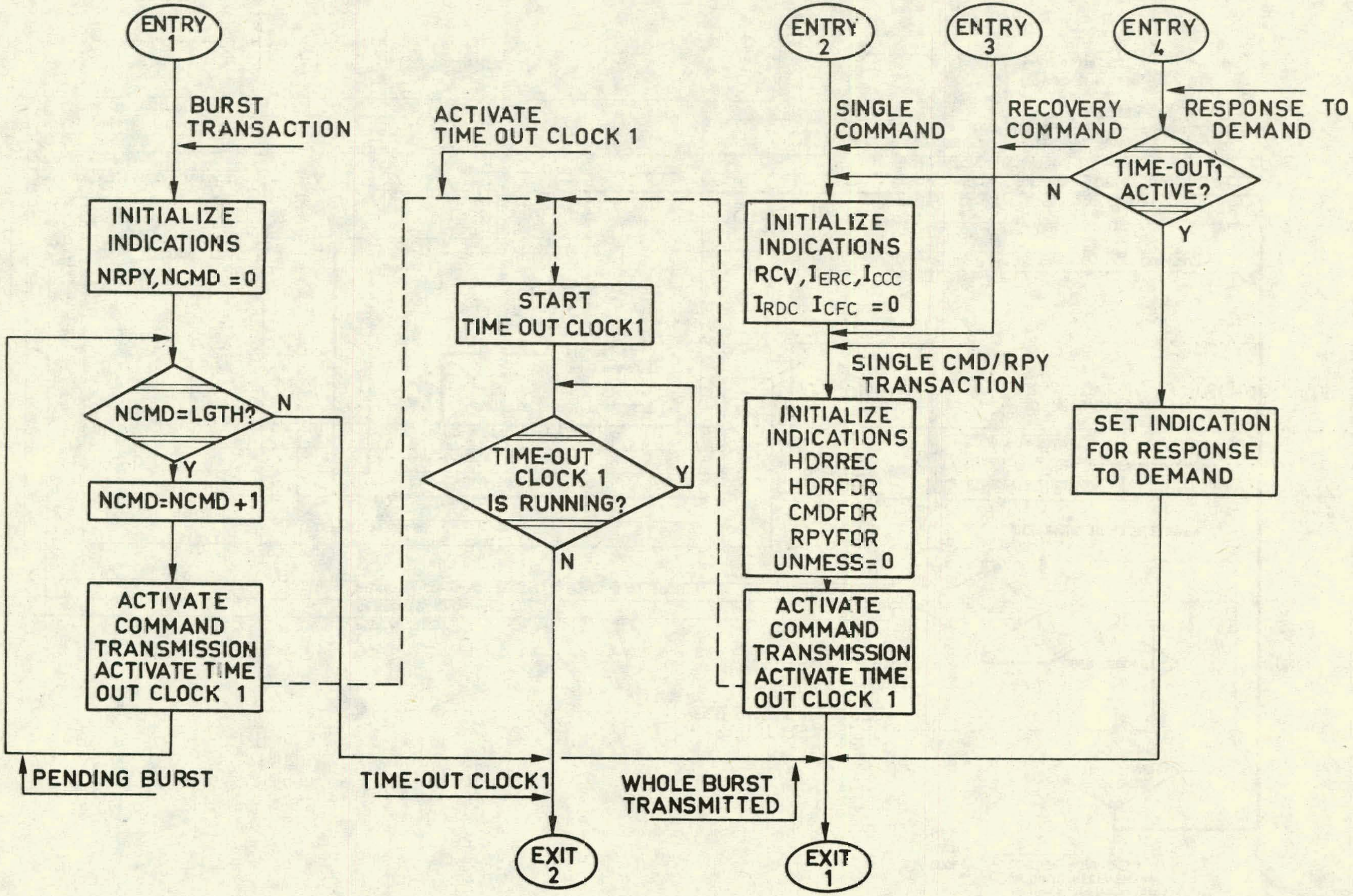
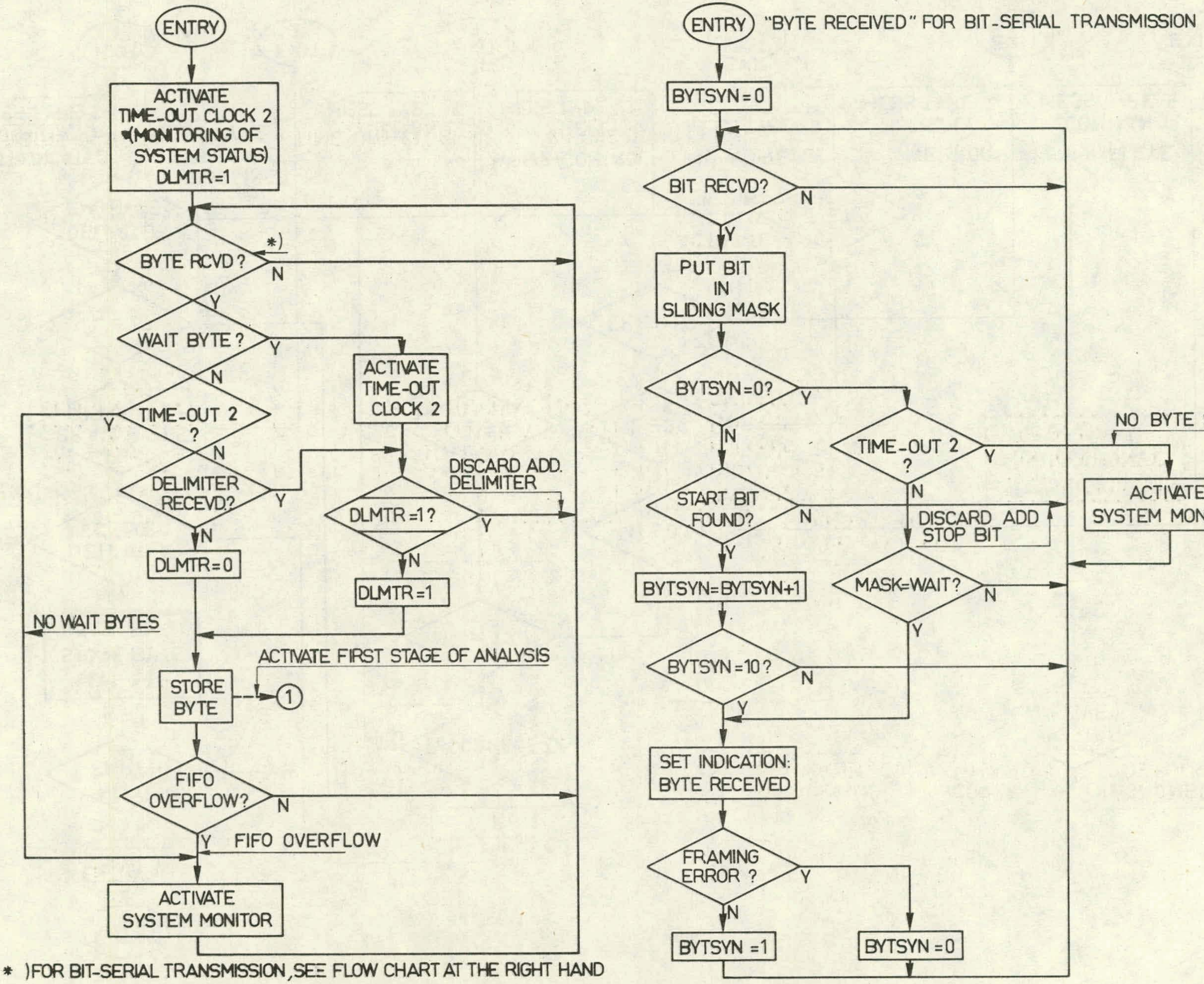

ENTRY 1
BURST TRANSACTION
INITIALIZE INDICATIONS NRPY,NCMD =0
NCMD=LGTH?
N
Y
NCMD=NCMD+1
ACTIVATE COMMAND TRANSMISSION ACTIVATE TIME OUT CLOCK 1
PENDING BURST
ACTIVATE TIME OUT CLOCK 1
START TIME OUT CLOCK1
TIME-OUT CLOCK 1 IS RUNNING?
Y
N
TIME-OUT CLOCK1
EXIT 2
ENTRY 2
ENTRY 3
ENTRY 4
SINGLE COMMAND
RECOVERY COMMAND
RESPONSE TO DEMAND
TIME-OUT1 ACTIVE?
N
Y
INITIALIZE INDICATIONS RCV,I_{ERC},I_{CCC} I_{RDC} I_{CFC} =0
SINGLE CMD/RPY TRANSACTION
INITIALIZE INDICATIONS HDRREC HDRFOR CMDFOR RPYFOR UNMESS=0
ACTIVATE COMMAND TRANSMISSION ACTIVATE TIME OUT CLOCK 1
SET INDICATION FOR RESPONSE TO DEMAND
WHOLE BURST TRANSMITTED
EXIT 1

Fig.6 : COMMAND HANDLER



*)FOR BIT-SERIAL TRANSMISSION, SEE FLOW CHART AT THE RIGHT HAND

Fig. 7: SD RECEIVER
DISCARDING OF ADDITIONAL
DELIMITER BYTES

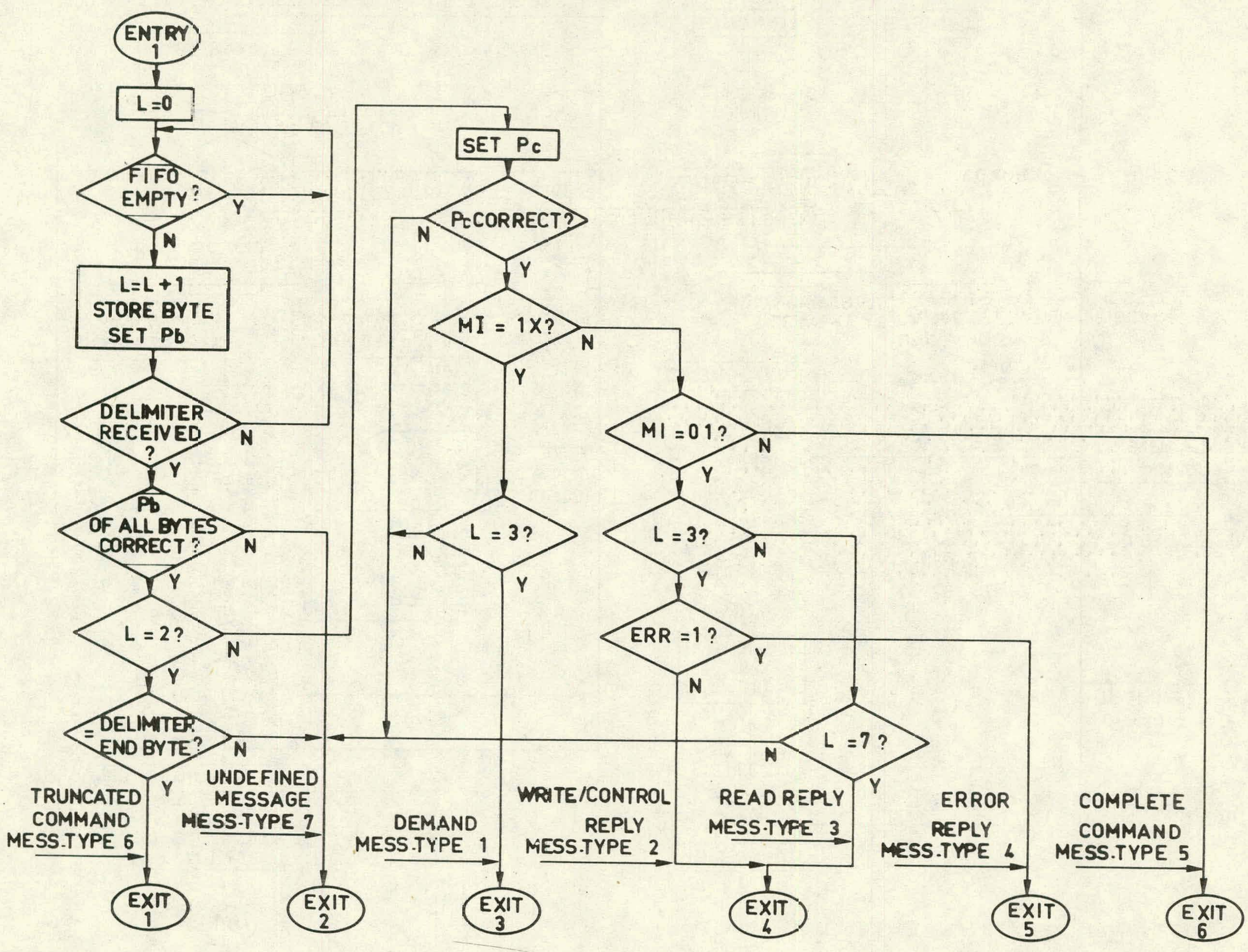


Fig.8 : EXAMPLE OF ANALYSIS OF RECEIVED MESSAGE
(FIRST STAGE OF ANALYSIS)

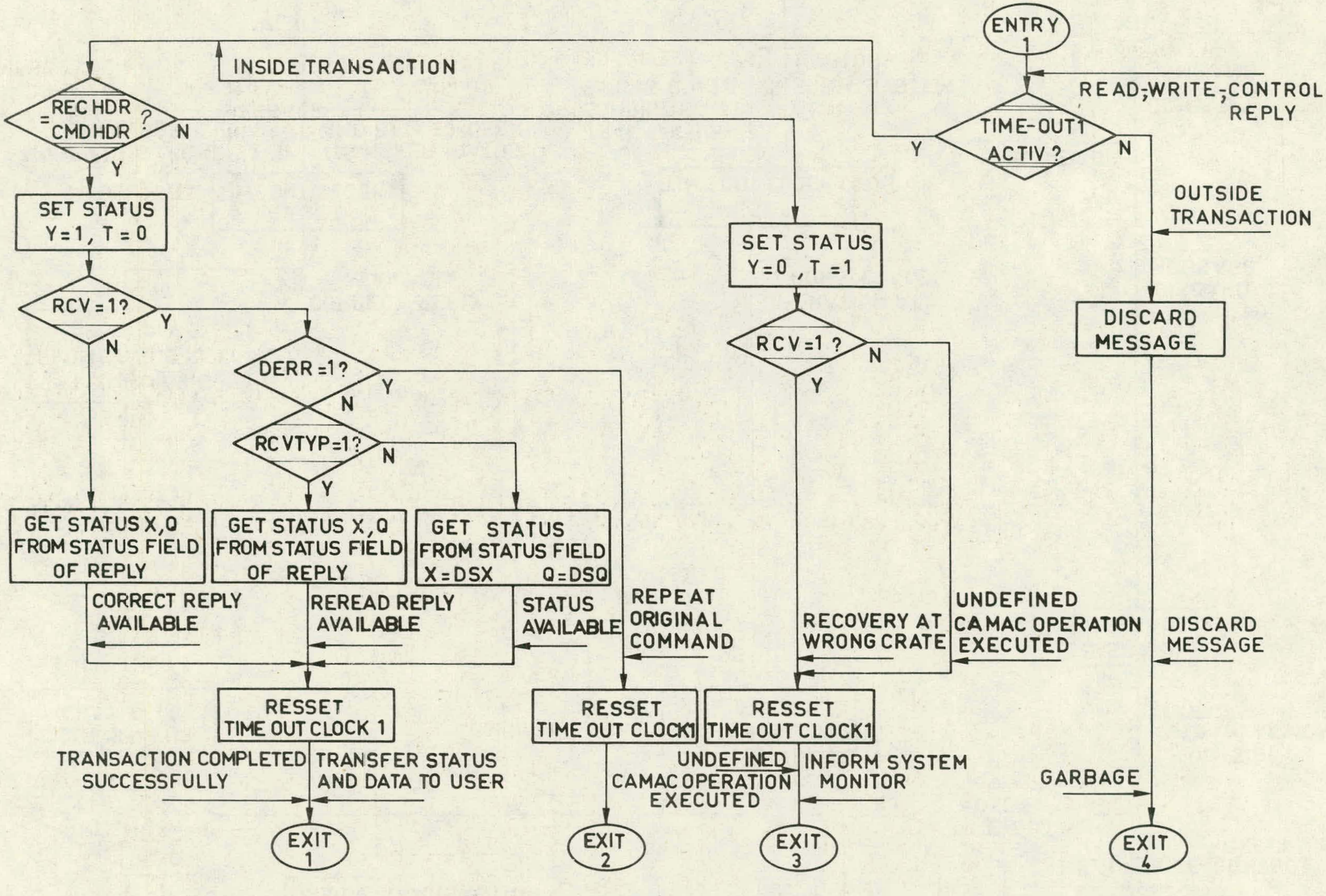


Fig. 9A: REPLY HANDLER (BASIC)
USED IN A SYSTEM WITHOUT ERROR RECOVERY
BASED ON THE DERR-BIT.

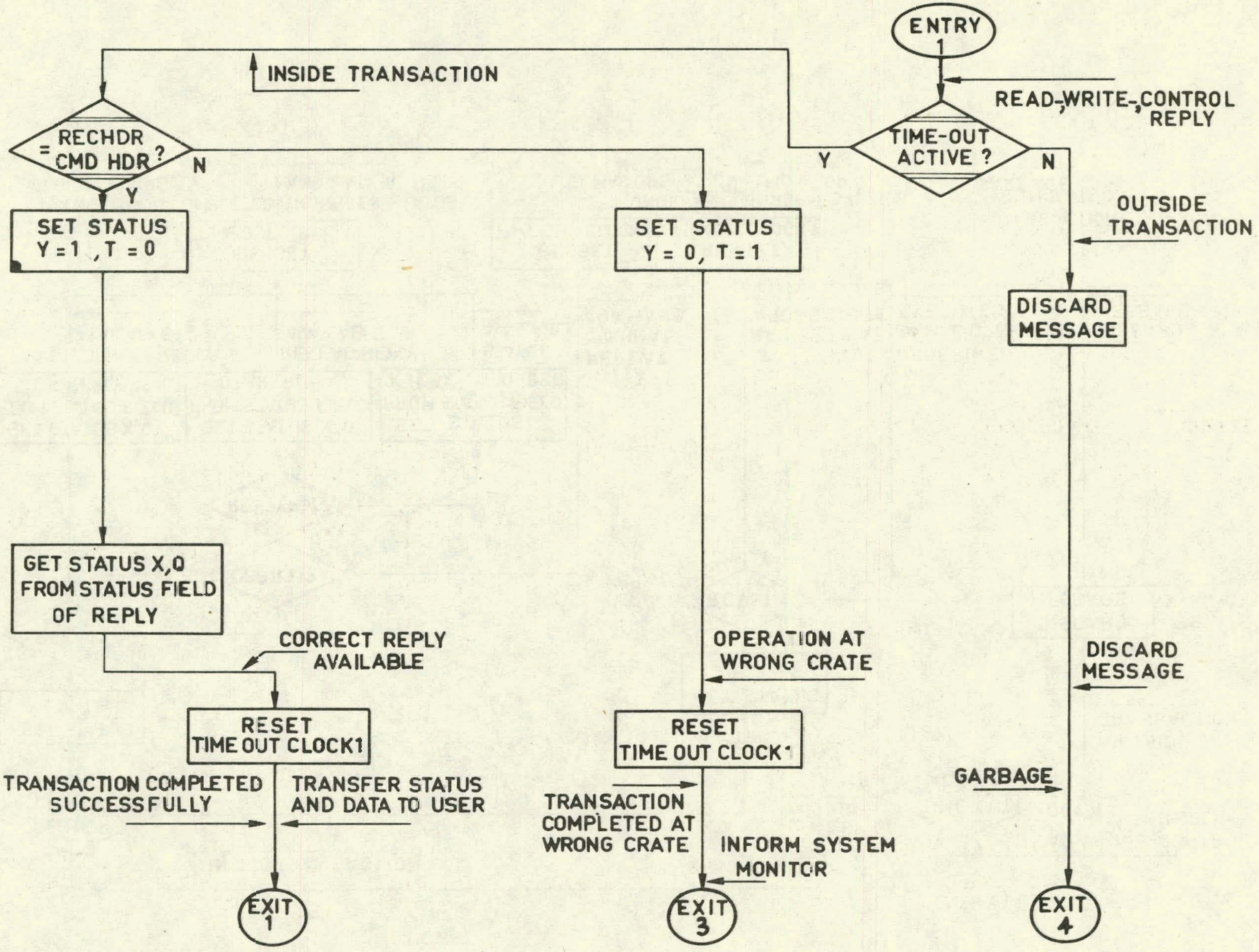


Fig.9B: REPLY HANDLER (EXTENDED) USED IN A SYSTEM WITH ERROR RECOVERY BASED ON THE DERR-BIT.

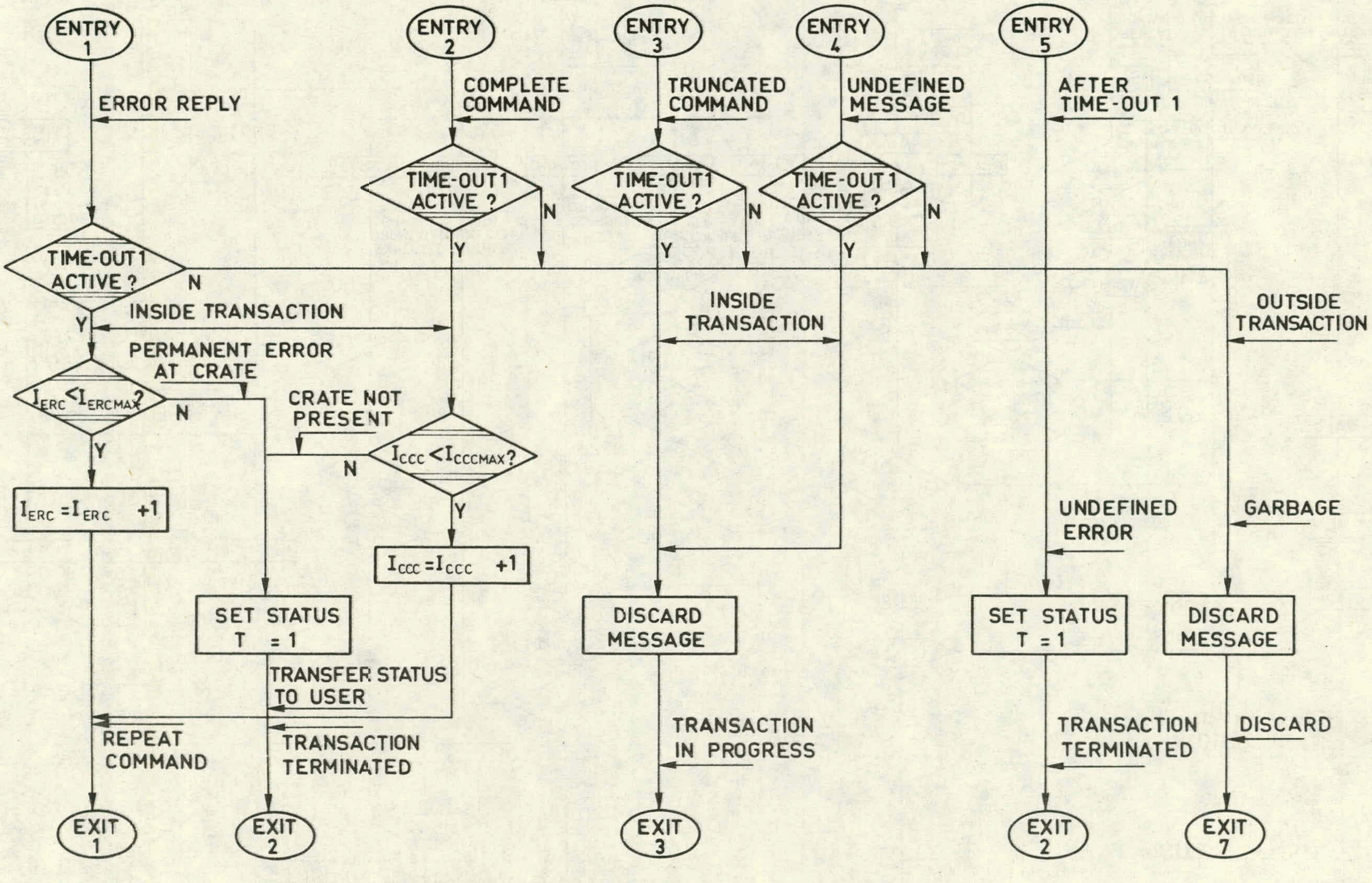


Fig. 10A: ERROR HANDLER (BASIC)
USED IN A SYSTEM WITHOUT ERROR RECOVERY
BASED ON THE DERR-BIT.

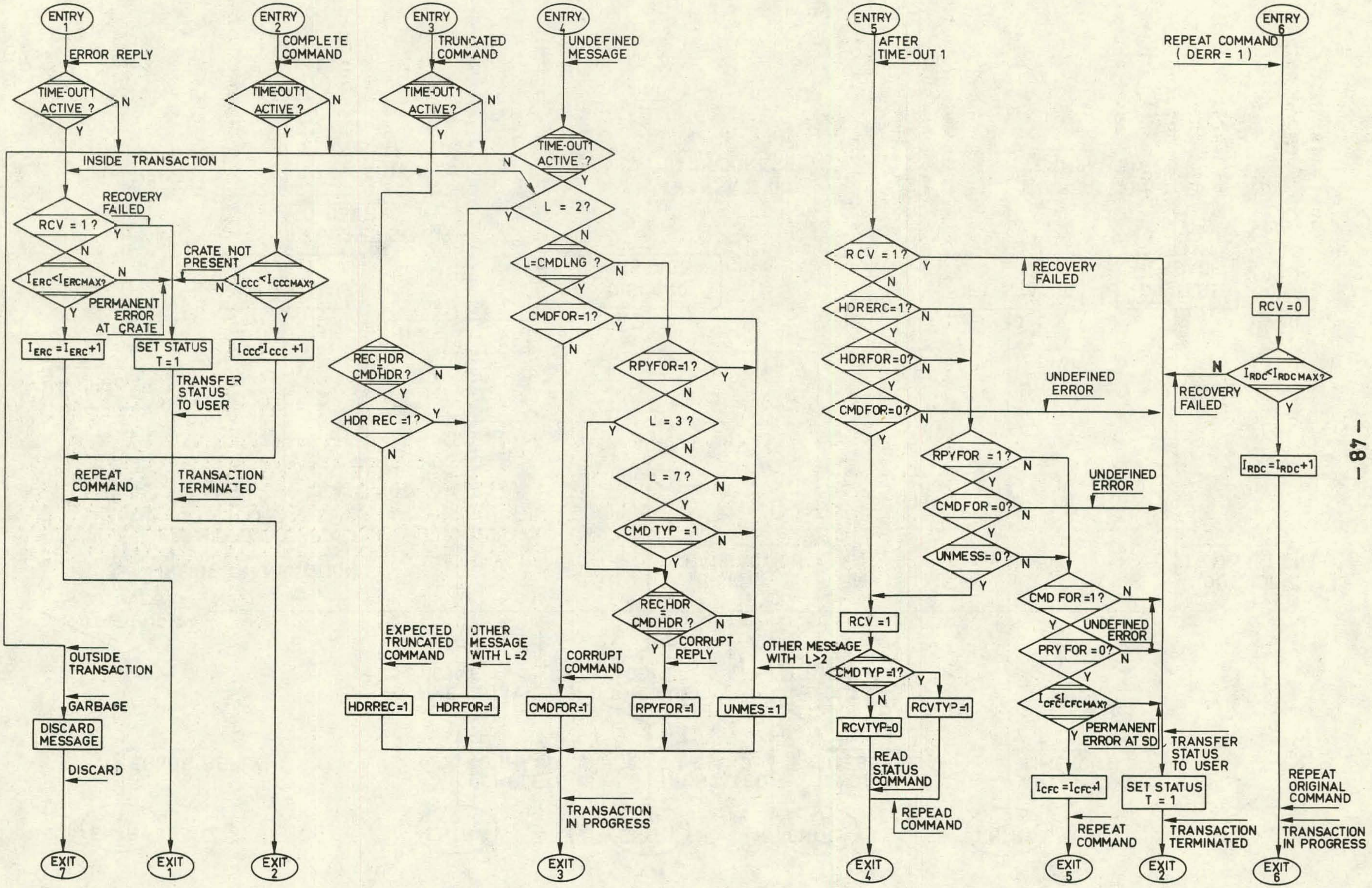
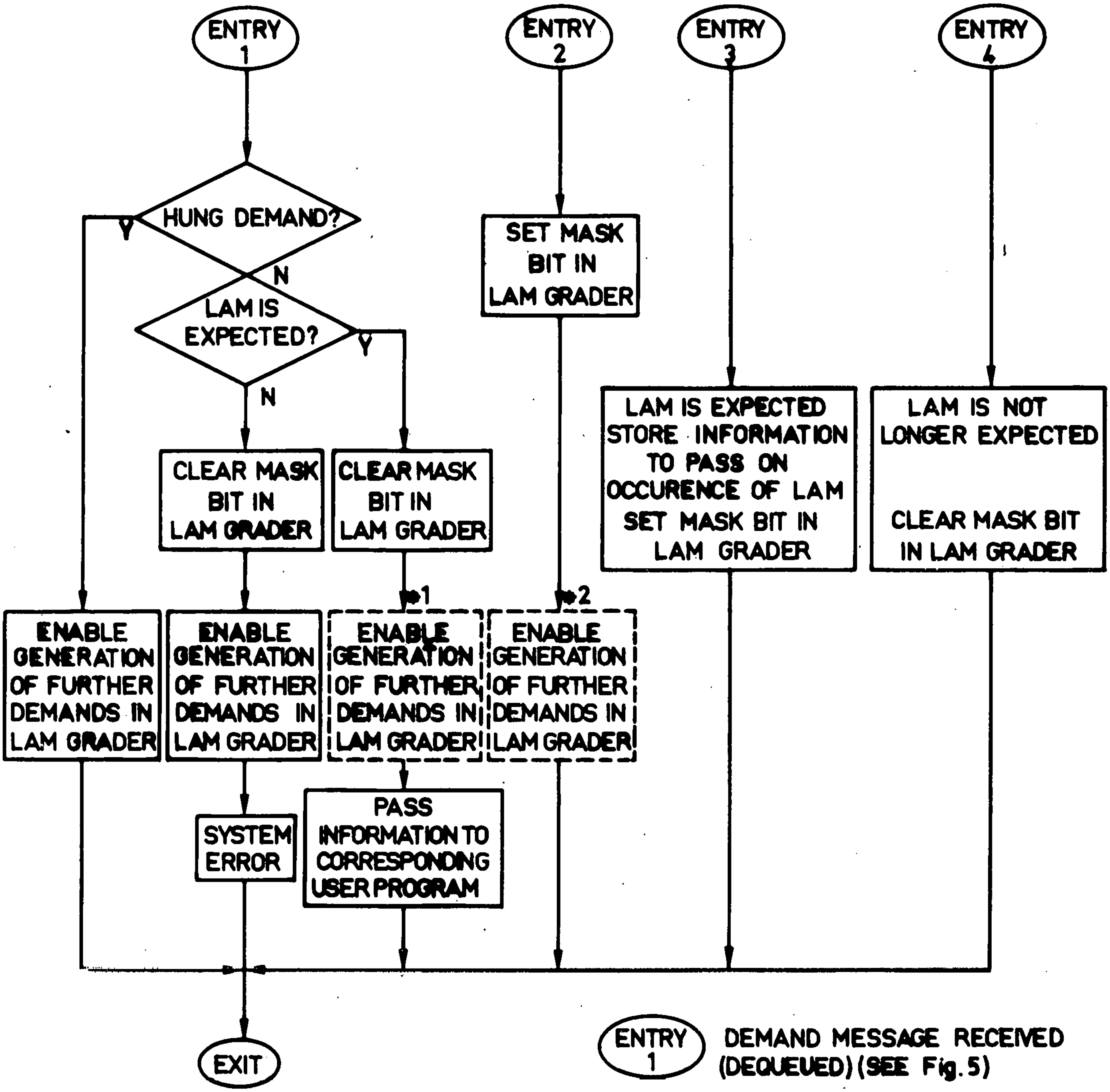


Fig. 10B: ERROR HANDLER (EXTENDED)

USED IN A SYSTEM WITH ERROR RECOVERY BASED ON THE DERR BIT AND FURTHER ANALYSIS OF UNDEFINED MESSAGES AND THE TRUNCATED COMMAND


ENTRY 1
ENTRY 2
ENTRY 3
ENTRY 4
HUNG DEMAND?
Y
N
LAM IS EXPECTED?
Y
N
SET MASK BIT IN LAM GRADER
CLEAR MASK BIT IN LAM GRADER
CLEAR MASK BIT IN LAM GRADER
LAM IS EXPECTED STORE INFORMATION TO PASS ON OCCURENCE OF LAM SET MASK BIT IN LAM GRADER
LAM IS NOT LONGER EXPECTED CLEAR MASK BIT IN LAM GRADER
*1
*2
ENABLE GENERATION OF FURTHER DEMANDS IN LAM GRADER
ENABLE GENERATION OF FURTHER DEMANDS IN LAM GRADER
ENABLE GENERATION OF FURTHER DEMANDS IN LAM GRADER
ENABLE GENERATION OF FURTHER DEMANDS IN LAM GRADER
SYSTEM ERROR
PASS INFORMATION TO CORRESPONDING USER PROGRAM
EXIT

ENTRY 1 DEMAND MESSAGE RECEIVED (DEQUEUED) (SEE Fig.5)

ENTRY 2 REENABLE DEMAND MESSAGE INITIATION (USER INITIATION)

ENTRY 3 CONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)

ENTRY 4 DISCONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)

*NOTE : IF ONLY BOX 1 IS USED, NESTING OF INTERRUPT SERVICE ROUTINE IS POSSIBLE -

IF ONLY BOX 2 IS USED, NO NESTING OF INTERRUPT SERVICE ROUTINE IS POSSIBLE

Fig.11 A : DEMAND HANDLER

DEMAND HANDLER WITH LAM GRADER

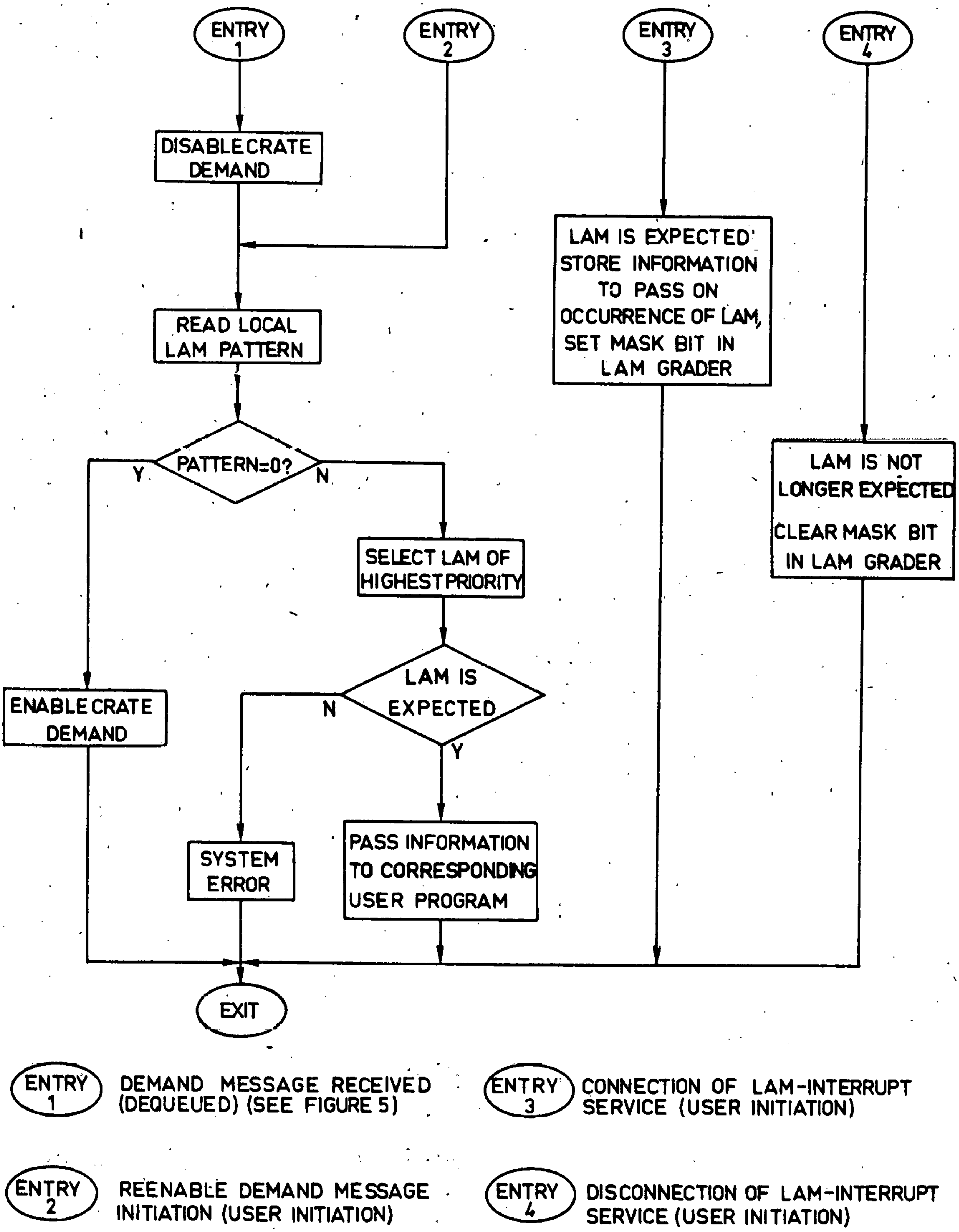
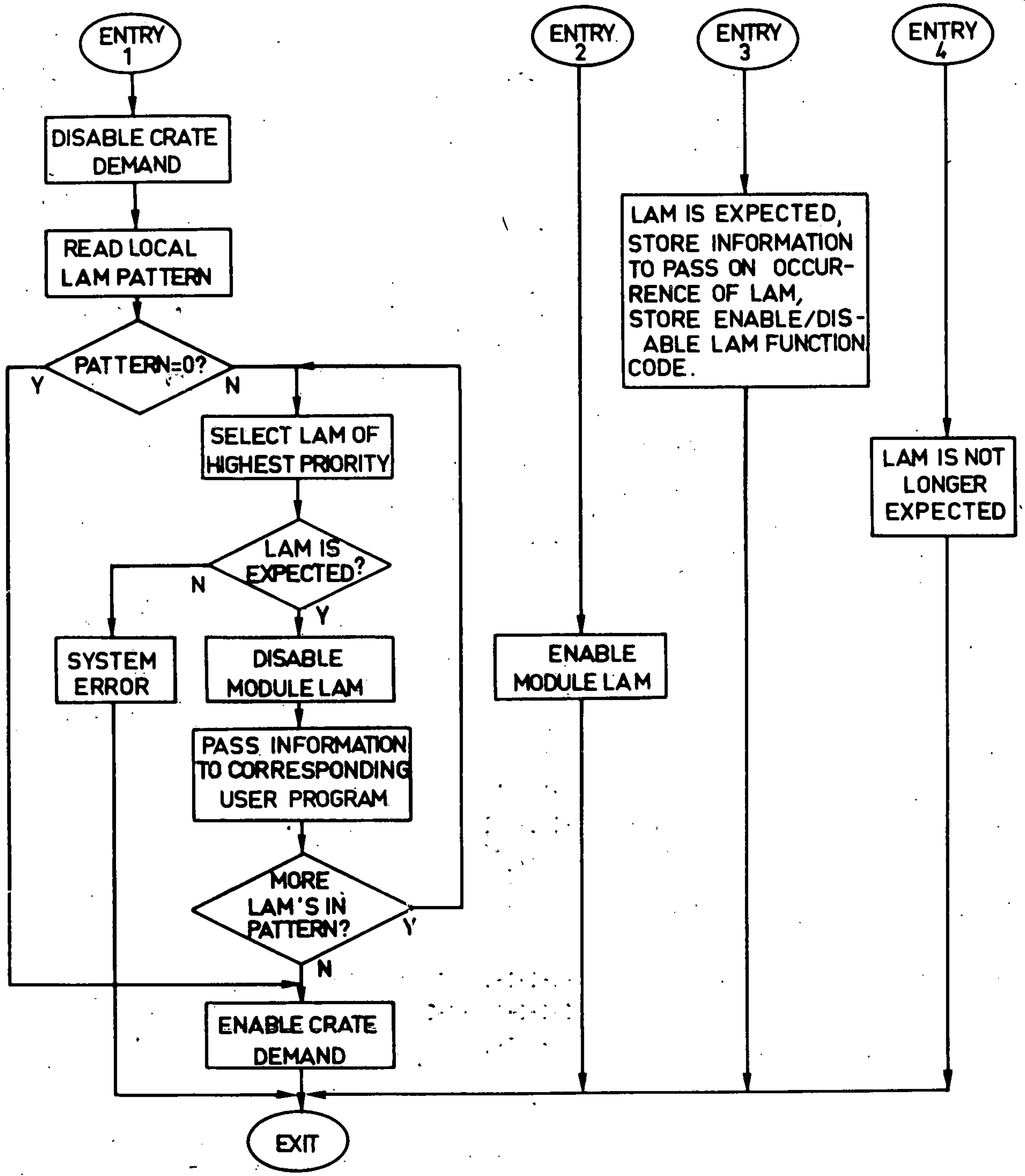

ENTRY 1
ENTRY 2
ENTRY 3
ENTRY 4
DISABLE CRATE DEMAND
READ LOCAL LAM PATTERN
PATTERN=0?
Y
N
SELECT LAM OF HIGHEST PRIORITY
LAM IS EXPECTED
N
Y
ENABLE CRATE DEMAND
SYSTEM ERROR
PASS INFORMATION TO CORRESPONDING USER PROGRAM
LAM IS EXPECTED: STORE INFORMATION TO PASS ON OCCURRENCE OF LAM, SET MASK BIT IN LAM GRADER
LAM IS NOT LONGER EXPECTED
CLEAR MASK BIT IN LAM GRADER
EXIT
ENTRY 1 DEMAND MESSAGE RECEIVED (DEQUEUED) (SEE FIGURE 5)
ENTRY 3 CONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)
ENTRY 2 REENABLE DEMAND MESSAGE INITIATION (USER INITIATION)
ENTRY 4 DISCONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)

Fig. 11 B: DEMAND HANDLER

DEMAND HANDLER WITHOUT LAM GRADER


ENTRY 1
DISABLE CRATE DEMAND
READ LOCAL LAM PATTERN
PATTERN=0?
Y
N
SELECT LAM OF HIGHEST PRIORITY
LAM IS EXPECTED?
N
Y
SYSTEM ERROR
DISABLE MODULE LAM
PASS INFORMATION TO CORRESPONDING USER PROGRAM
MORE LAM'S IN PATTERN?
Y
N
ENABLE CRATE DEMAND
EXIT
ENTRY 2
ENABLE MODULE LAM
ENTRY 3
LAM IS EXPECTED, STORE INFORMATION TO PASS ON OCCURRENCE OF LAM, STORE ENABLE/DISABLE LAM FUNCTION CODE.
ENTRY 4
LAM IS NOT LONGER EXPECTED

ENTRY 1 DEMAND MESSAGE RECEIVED (DEQUEUED) (SEE FIGURE5)

ENTRY 3 CONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)

ENTRY 2 REENABLE DEMAND MESSAGE INITIATION (USER INITIATION)

ENTRY 4 DISCONNECTION OF LAM-INTERRUPT SERVICE (USER INITIATION)

Fig.11C: DEMAND HANDLER

DEMAND HANDLER WITHOUT LAM GRADER, WITH ADDITIONAL DISABLE MODULE LAM

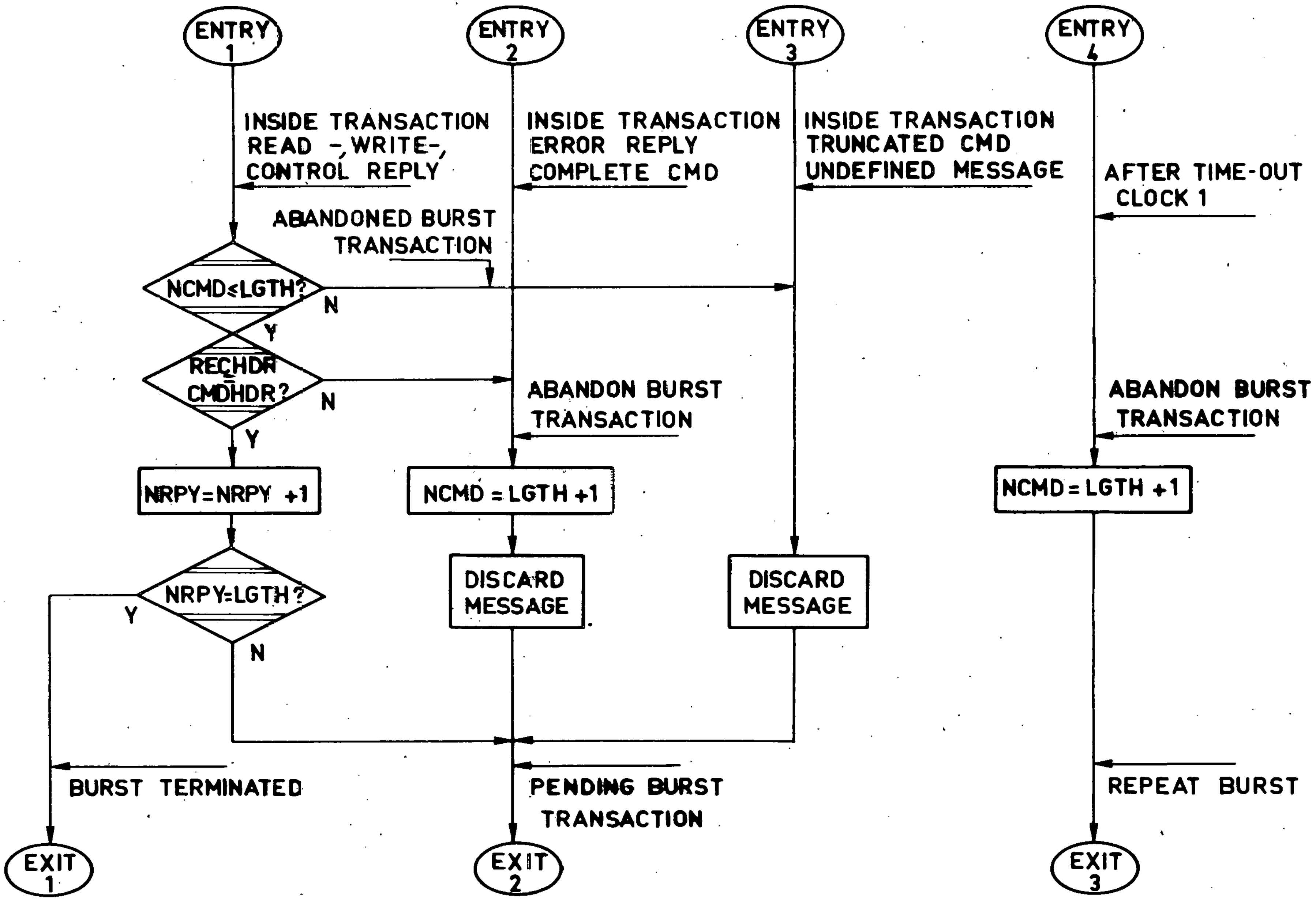


Fig. 12 : BURST HANDLER

APPENDIX A

LAM GRADERS FOR THE SERIAL CRATE CONTROLLER TYPE L2

A1. Introduction

The CAMAC Serial Highway provides a method of demand identification that is not to be found in the Branch Highway (EUR 4600e, IEEE Std 596). The Serial Driver (SD) receives demands by means of Demand Messages from the Serial Crate Controller (SCC). The SCC is able to generate three-byte messages comprising Crate Address, Serial Graded LAM (SGL) Number and a Checksum. Thus the SD is able to identify the location of the module that is generating the LAM, without recourse to CAMAC read or test operations. In order to generate the SGL field in the Demand message a five-bit pattern is presented to the rear connector of the SCC. These bits can of course be any bit pattern, or simply the Station Number of the module that is generating the LAM.

This description is intended as a guide to those implementing LAM Graders to be used in Serial Systems. The features described are those that should be found in any LAM Grader used for the Serial System and, while designers may add features that they consider are desirable, a LAM Grader containing only the features described would form a viable unit.

A2. Features

A2.1 L Status

While Dataway Busy is not present all modules are free to assert L signals, and the incoming L signals from the SCC are loaded into a 24-bit L Status register. When the LAM Grader receives Busy from the Dataway or Demand Busy from the SCC it maintains the previous contents of this register. If Busy occurs (e.g. from an auxiliary controller) while a Demand message is being transmitted, some modules may be forced to remove their L signals, but the immediately previous L status is available from the register in order to determine the SGL field of the message.

A2.2 L Mask

A 24-bit register is provided to mask the output from the L Status register ('1' enables, '0' disables). The register is cleared by Initialise and CL2 (F11). It is also accessed by BS2 (F20 A13) and BC2 (F22 A13). See section A3 for explanation of BS2 and BC2.

A2.3 L Request

The logical AND of the L Status and the L Mask registers forms the L Request 'register'.

A2.4 Encoded L Request

The output from the L Request 'register' is encoded to form a five-bit number which may be read via the Dataway by RD1 (F0 A0). This five-bit number is used for the SGL field of the Demand message. Where more than one bit is present on the encoder input the lowest bit number will take preference. Thus, if no patching of incoming L signals is employed, the SGL Number indicates the left-most L signal in the Crate.

A2.5 Demand Message Generation

If there is a bit present at the output from the encoder, the Demand Message Initiate (DMI) is passed to the SCC. The Demand message, with the SGL number, is then transmitted on the serial highway. When a Demand message has been initiated the LAM Grader automatically suspends the generation of further Demand messages (except Hung Demands) until unlocked by XEQ (F25 at A2). This prevents generation of unwanted Demand messages that could occur in certain circumstances involving the clearing of a LAM.

When a Hung Demand message has been initiated the LAM Grader suspends the generation of further Demand messages (except Hung Demands) until unlocked by XEQ (F25 at A3), and meanwhile ignores F25 at A2.

After the LAM Grader has been initialised (Z) the suspension and interlocking features described above must allow Demand message generation.

The Hung Demand Timer is reset (STIM removed) when there are no LAM's present at the Decoder, providing that a Hung Demand has not occurred. In addition it is reset by XEQ at A3.

Demand message generation can be enabled and disabled by the use of ENB (F26 A1) and DIS (F24 A1). The ENB operation does not affect the Demand message suspension interlocks. These can be unlocked immediately from any state by XEQ at both A2 and A3. Alternatively, the LAM Grader can be allowed to go to the Hung Demand state and then unlocked in the normal way.

The ENB and XEQ commands should remove DMI for a sufficient period of time to allow a new Demand message to be generated when a LAM is already present. In order to prevent ambiguous Demand message generation when more than one Hung Demand has been received by the SD, the XEQ at A3 command must be ignored if a Hung Demand is not present (e.g. it has already been cleared by a previous XEQ at A3).

A2.6 Using the LAM Grader

LAM's can be handled at the SD by one of two methods. Either they can be serviced one at a time, or they can be masked off in the LAM Grader as they occur and queued in the SD to be serviced later.

In the first method, following receipt of a Demand message the LAM is serviced and the LAM Grader is reactivated by XEQ at A2. This allows further Demand message generation and subsequent Demand messages are handled in the same way. If a Hung Demand message is transmitted the LAM Grader will suspend generation of further Demand messages, except for further Hung Demand messages, until an XEQ command at A3 is performed. Following this command the LAM Grader will give a message corresponding to the left-most LAM in the crate (if any).

Protection against Hung Demand messages arising from delay in servicing the LAM may be provided by masking off the LAM with BC2 at A13 until the LAM is serviced, when it can be masked on with BS2 at A13.

In the second method, following the receipt of a Demand message the relevant bit in the LAM Grader is masked off and the LAM Grader reactivated. 'This is done by BC2 at A13 followed by XEQ at A2.

To simplify this reactivation the command BC2 at A11 with Write data W1 to W5 corresponding to the SGL field of the Demand message has the same effect as BC2 at A13 and XEQ at A2. Following a Hung Demand message (with SGL field = 31) the command BC2 at A11 with Write data 31 has the same effect as XEQ at A3. Thus for this method of operation BC2 at A11 with data equal to the received SGL number is always the response to a Demand message. Following the operation to service the LAM itself, the relevant mask bit can be set with BS2 at A13.

A2.7 Internal LAM

The Serial LAM Grader should be provided with its own internal LAM for test purposes. This is set by either front panel push button or Lemo connector. This LAM is transmitted on the L line of the Dataway connector and is gated with and cleared by CLM (F10 A0). It is tested with TEST (F8 A0). An additional Lemo connector at the front panel gives an output when the LAM is set.

A2.8 Recommended Commands For Basic Features

Feature	Address	Commands
Internal LAM	A0	CLM (F10) TLM (F8)
DMD MSG Generation	A1	DIS (F24) ENB (F26)
Reset MSG Suspension	A2	XEQ (F25)
Reset Hung Demand	A3	XEQ (F25)
Encoded L Register	GP1 A0	RD1 (F0)
L Mask Register	GP2 A13	CL2 (F11) BC2 (F22)* BS2 (F20)*
L Mask Register	GP2 A11	BC2 (F22)#

NOTES:

* Write data W1-W5 determines which bit of the register is set by BS2 and cleared by BC2.

# If the Write data W1-W5 is in the range 1-30 this determines which bit of the register is cleared. The command also has the effect of XEQ at A2.

If the Write data W1-W5 has value 31 the command has the effect of XEQ at A3.

A2.9 Additional Features

If a Serial LAM Grader has features additional to those described, the function code and the subaddresses should be in accordance with those specified for LAM features in modules in EUR 4100e, IEEE Std 583. Specifically, it is strongly recommended that the L Status and L Request register should be addressed at Gp2 A12 and A14 respectively. Where these registers are accessed by the Dataway as 24-bit words then L1 should correspond to R1/W1 and L21 to R21/W21 etc.

In order to make the LAM Grader useful in a parallel highway system conforming to EUR 4600e and IEEE Std 596, the Selected LAM Present signal can be used as the equivalent of External Demand. Provision should be made for accepting the Controller Addressed signal from Crate Controller Type A1 as an equivalent of the Station Number address of the LAM Grader. The encoded LAM number can be read via the Dataway by RD1 (F0 A0). In order to identify the crates with active demands, if the LAM Grader has a demand present it should respond to a multi-crate operation RD1 (F0 A1) by generating an output on one of the lines R1 to R7, corresponding to its crate address in the parallel system. The SGL Encoder connector on the LAM Grader is not pin-for-pin compatible with the LAM Grader connector on Crate Controller Type A1, but can be suitably cross connected.

A2.10 Initialisation

The LAM Grader responds to the Initialise operation by clearing its mask register, clearing its LAM, disabling message generation and removing any message generation suspension.

A3. Treatment of LAM numbers

The SD receives Demand messages in which LAM's are identified by binary codes. The Demand Handler should be able to use these numbers in controlling corresponding features of the LAM Grader. Therefore the two non-standard function codes F20 and F22 are used to set and clear individual bits of the LAM Mask register. A five-bit number on the write lines (W1-W5) is decoded in the LAM Grader to select the bit to be set or cleared by the Bit Set Group2 (BS2) or Bit Clear Group2 (BC2) commands. Thus the Demand Handler is concerned only with LAM Numbers and is not concerned with bit positions in a 24-bit word.

A4. Description of the block diagram

The Block Diagram (Fig. A2) is intended as an aid in understanding the operation of the features described in appendix A. It is not suggested that the diagram represents an actual implementation. The following text may also be read in conjunction with the flow diagram (Fig. A3).

The 24 L signals are loaded into a LAM Status register before being presented to a 24-bit LAM Mask. All Mask bits can be cleared simultaneously by initialise or a clear operation, and each bit can be set or cleared individually.

Following the Mask, the encoder selects the highest priority LAM and presents it as a 5-bit encoded pattern to the 5 SGL lines. This pattern can also be read via the Dataway. The OR of the 5 SGL signals is used to indicate that an active LAM is present.

The Enable/Disable flip-flop is used to give overall control of the LAM Grader and allows active LAM's to assert Selected LAM Present (SLP) at the SCC.

The sequence of operation of the interlock circuitry is as follows, starting with all flip-flops reset:

The assertion of SLP sets the Timer Start flip-flop which asserts Start Timer (STIM) to the SCC. The SCC responds by asserting Time-Out (TIMO) which causes the LAM Grader to set its Demand Transmitted flip-flop and to assert Demand Message Initiate (DMI), thereby causing the SCC to transmit a Demand message.

The Demand Transmitted flip-flop prevents the Timer Start flip-flop being set again until the LAM Grader has received a Command that acknowledges the Demand message (XEQ at A12). The removal of SLP will however reset the Timer Start flip-flop, thus stopping the Hung Demand Timer in the SCC.

When the Timer in the SCC times out, TIMO is removed for 200nS. This removal of TIMO while STIM is present is used to set the Hung Demand Transmitted flip-flop in the LAM Grader, and a Hung Demand message is initiated on the re-assertion of TIMO.

The Hung Demand Transmitted flip-flop maintains STIM until the Hung Demand is acknowledged by the command XEQ at A3. During this time further Hung Demand messages are transmitted at the end of each time-out period.

The SCC does not detect the removal of STIM until some time after Dataway Busy is removed, see Fig. 14-1 in EUR 6100e and IEEE Std 595. The Reset Timer flip-flop is used to ensure that, following an XEQ Command (at A2 or A3 as appropriate), the timer in the SCC will always be stopped. If an active LAM is still present then the Timer will be re-started immediately.

		METHOD OF USING LAM GRADER		
		1	2	3
FURTHER DEMANDS ALLOWED FROM SAME CRATE WHILE CURRENT DEMAND IS SERVICED		YES	NO	NO
MASK REGISTER USED BY SD DEMAND HANDLER		YES	YES	NO
RISK OF HUNG DMD'S DUE TO LATE DMD SERVICING REDUCED BY		QUICK LAM MASKING AND DEMAND REACTIVATION BY DMD HANDLER	QUICK LAM MASKING BY DMD HANDLER AND QUICK LAM SERVICE BY USER PROGRAM IN CASE OF ADDITIONAL UNMASKED LAM'S IN ONE CRATE	QUICK LAM SERVICE BY USER PROGRAM
RECOVERY FORM HUNG DMD'S		UNAMBIGUOUS	UNAMBIGUOUS	ONE LAM MAY CAUSE DOUBLE SERVICE

COMMAND TO THE LAM GRADER FROM THE SERIAL DRIVER DEMAND HANDLER	ON REGULAR DEMAND	BC2:F(22).A(11)(DATA 1-30)
		=
		BC2:F(22).A(13)(DATA 1-30)
		+ XEQ:F(25).A(2)
			BC2:F(22).A(13)(DATA 1-30)	-

	ON HUNG DEMAND	BC2:F(22).A(11)(DATA=31)
		=
		XEQ:F(25).A(3)
			XEQ:F(25).A(3)	XEQ:F(25).A(3)

COMMAND TO THE LAM GRADER INNITIATED BY THE USER	ON REGULAR DEMAND	BS2:F(20).A(13)(DATA 1-30)	BS2:F(20).A(13)(DATA 1-30)
			+ XEQ:F(25).A(2)
				XEQ:F(25).A(2)

	ON HUNG DEMAND	-	-	-

Fig.A1: METHODS OF USING LAM GRADER TO HANDLE AN INDIVIDUAL DEMAND

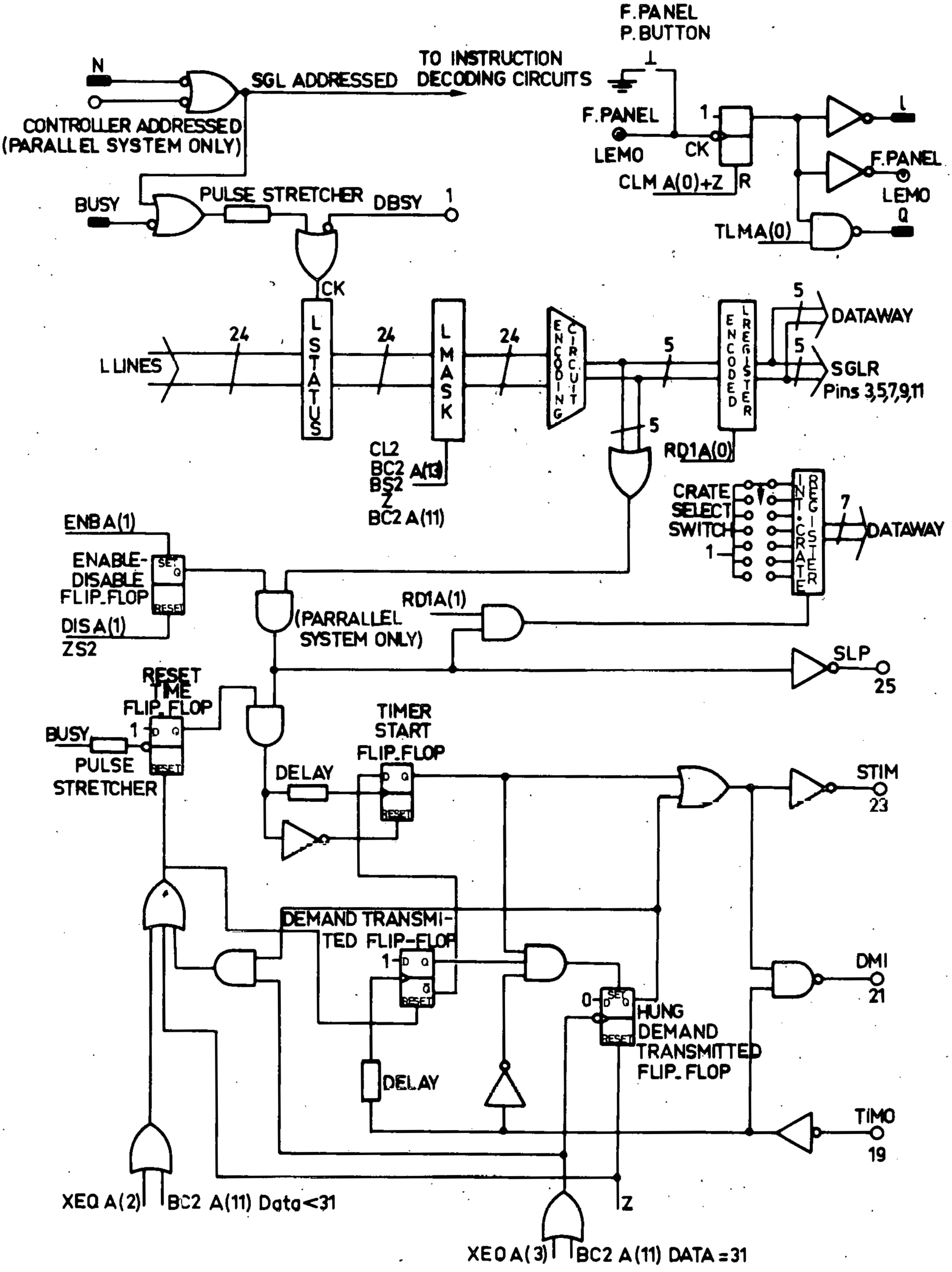
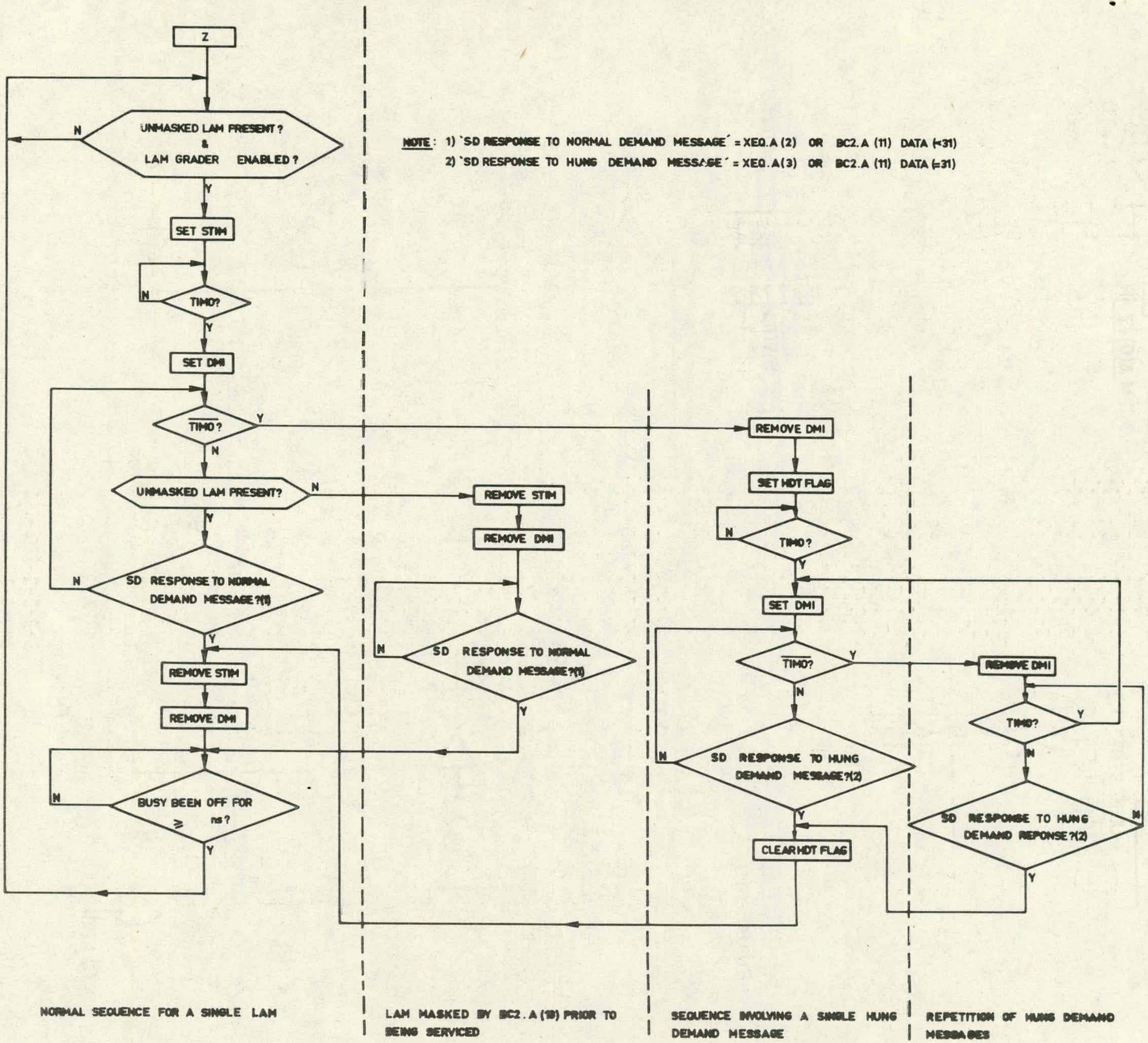


Fig. A2: SERIAL LAM GRADER RECOMMENDATIONS


Z
UNMASKED LAM PRESENT ?
&
LAM GRADER ENABLED ?
N
Y
SET STIM
TIMO?
N
Y
SET DMI
TIMO ?
Y
N
UNMASKED LAM PRESENT?
N
Y
SD RESPONSE TO NORMAL DEMAND MESSAGE ?(1)
N
Y
REMOVE STIM
REMOVE DMI
BUSY BEEN OFF FOR ≥ ns ?
N
Y
REMOVE STIM
REMOVE DMI
SD RESPONSE TO NORMAL DEMAND MESSAGE ?(1)
N
Y
REMOVE DMI
SET HDT FLAG
TIMO ?
N
Y
SET DMI
TIMO?
Y
N
SD RESPONSE TO HUNG DEMAND MESSAGE?(2)
N
Y
CLEAR HDT FLAG
REMOVE DMI
TIMO?
Y
N
SD RESPONSE TO HUNG DEMAND REPONSE ?(2)
N
Y
NORMAL SEQUENCE FOR A SINGLE LAM
LAM MASKED BY BC2. A (10) PRIOR TO BEING SERVICED
SEQUENCE INVOLVING A SINGLE HUNG DEMAND MESSAGE
REPETITION OF HUNG DEMAND MESSAGES

NOTE: 1) 'SD RESPONSE TO NORMAL DEMAND MESSAGE' = XEQ.A (2) OR BC2.A (11) DATA (<31)
2) 'SD RESPONSE TO HUNG DEMAND MESSAGE' = XEQ.A(3) OR BC2.A (11) DATA (=31)

Fig.A3: SEQUENCE OF OPERATION

☆ U.S. GOVERNMENT PRINTING OFFICE: 1978— 261-324/140