Cotectic

Solid
Solutions

PSS-04-256
4-255 Data Bus Protocol Extensions

By Chris Plummer, Cotectic Ltd.

May 10th 1997.

ANNOTATED DRAFT
Revision 1.2.f
[To be ratified as PSS-04-256]

NOTE

This is a preliminary draft copy for information and review purposes only. It is not to be used as a definition or a specification.

Abstract: This document defines a set of protocol extensions for use with the PSS-04-255 Data Bus. The protocol extensions consist of interrogation and response word formats which facilitate data distribution, acquisition, and transfer across the bus.

This Annotated draft version of the document is issued for the purposes of review by the OBDH standards panel. All of the annotations are in italics and most of them will be removed from the final issue of the standard.

Foreword

The 4-201 Data Handling System (DHS) standard specifies a set of services, protocols, and interfaces to be provided by onboard data handling systems for unmanned spacecraft. The standard comprises a set of documents, each of which is dedicated to one aspect of the 4-201 DHS. The diagram below shows the relationships between these documents and the following paragraphs describe the role of each of them.

- PSS-04-201
  4-201 Data Handling System Standard
  - PSS-04-251
    Specification for the 4-201 DHS Transfer and User Layers
  - PSS-04-252
    Specification for the 4-201 DHS Data Link Layer
  - PSS-04-253
    Specification for the Electrical User Interfaces for use in 4-201 Data Handling Systems
  - PSS-04-254
    Serial Communication Link Specification for 4-201 Data Handling Systems
  - PSS-04-255
    4-255 Data Bus Specification
  - PSS-04-256
    4-255 Data Bus Protocol Extensions
  - PSS-04-257
    4-255 Data Bus Modem Specification

PSS-04-201 4-201 Data Handling System Standard This document specifies the reference model to be used to describe 4-201 Data Handling Systems. It also establishes common terminology to be used in 4-201 DHS descriptions.

The reference model describes a layered architecture. All of the other documents in the series specify parts of particular layers. For example, PSS-04-251 specifies the services to be provided by the 4-201 DHS transfer and user layers.

PSS-04-251 Specification for the 4-201 DHS Transfer and User Layers.This document specifies the services which are to be provided by the transfer and user layers of a 4-201 Data Handling System.

PSS-04-252 Specification for the 4-201 DHS Data Link Layer. This document specifies the services which must be provided by the Data Link Layer of a 4-201 DHS. This standard includes the definitions of primitives used to access the Data Link services.

PSS-04-253 Specification for the Electrical User Interfaces for use in 4-201 Data Handling Systems. This document specifies a number of electrical interfaces to sensors and actuators onboard a spacecraft. These interfaces include serial digital interfaces, analogue interfaces, and discrete digital interfaces.

PSS-04-254 Serial Communication Link Specification for 4-201 Data Handling Systems. This document specifies the electrical and physical characteristics of a set of serial communication links for use in 4-201 Data Handling Systems.

PSS-04-255 4-255 Data Bus Specification. This document defines a multiple access data interchange bus for 4-201 Data Handling Systems.

PSS-04-256 4-255 Data Bus Protocol Extensions This document specifies a number of protocols for use over the 4-255 Data Bus.

PSS-04-257 4-255 Data Bus Modem Specification This document specifies the characteristics of modems for use on the 4-255 Data Bus.

Contents

1. Introduction.

This standard defines a number of protocol extensions for use with the 4-255 Data Bus specified in the ESA PSS-04-255 document. Each protocol extension provides interrogation word formats which extend the generic capabilities defined in the body of the PSS-04-255 standard. Furthermore, the protocol extensions are designed to support the use of other standards in the PSS-04-201 Data Handling System series. Protocol Extensions 1 and 2, for example, can be used to support the Electrical User Interfaces defined in the PSS-04-253 document.

The separation of sets of interrogation formats into a number of protocol extensions, rather than attempting to define a single instruction set with many options, offers several advantages. Among these are:

- The protocol extensions need not all be defined at the same time. Protocol extensions for immediate requirements have been defined here. Other protocol extensions may be introduced in the future without compromising the stability of this standard or of any other protocol extensions.
- For a given implementation, only the relevant protocol extensions need be used. This should significantly reduce the complexity of many system designs.
- System profiling and documentation is significantly simplified since the wide variety of options and inevitable non-compliances that would be associated with a single instruction set are not encountered in the protocol extension approach.

1.1. Applicable Documents.

The following documents are applicable documents to this standard:

1) PSS-04-201 4-201 Data Handling System Standard.
2) PSS-04-255 4-255 Data Bus Specification.

1.2. Reference Documents.

The following documents are reference documents to this standard:

1) CCSDS 301.0-B-2 Time Code Formats, April 1990.
2) PSS-04-257 4-255 Data Bus Modem Specification.

2. Scope.

Each protocol extension in this standard consists of:

- a set of named instructions,
- the specified behaviour of those instructions, and
- a definition of the encoding of the instructions and their responses into 4-255 Data Bus interrogation and response words.

[illegible]

- Protocol Extension 1 (PE1) - Data Acquisition and Pulse Instruction Protocol,
- Protocol Extension 2 (PE2) - Register Load Instruction Protocol,
- Protocol Extension 3 (PE3) - Basic Control Instruction Protocol,
- Protocol Extension 4 (PE4) - Remote Bus Interface Protocol,
- Protocol Extension 5 (PE5) - Serial Time Distribution Protocol,
- Protocol Extension 6 (PE6) - Virtual Circuit Protocol,
- Protocol Extension 7 (PE7) - Memory Access Protocol.

The protocol extensions may include definitions of error conditions which must be detected by the protocol. However, the means of signalling such errors to higher layers and the definition of actions to be taken in response to those errors are beyond the scope of this document. Information relating to inter-layer signalling can be found in the specifications for those layers, such as PSS-04-251, the Specification for the 4-201 DHS Transfer and User Layers.

2.1. PSS-04-256 Document Structure.

The remainder of this document is organised as follows:

- Section 3 contains definitions of terms and conventions which are used in this document. These terms and conventions constitute a formal part of the 4-256 Data Bus Protocol Extensions specification.

- Section 4 contains a discussion of some system level aspects which should be considered when using protocol extensions. This section does not constitute a formal part of the 4-255 Data Bus Protocol Extensions specification

- Section 5 contains general requirements concerning the use of Protocol Extensions and their implementation on 4-255 Data Bus terminals. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 6 contains the requirements applicable to Protocol Extension 1, the Data Acquisition and Pulse Instruction Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 7 contains the requirements applicable to Protocol Extension 2, the Register Load Instruction Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 8 contains the requirements applicable to Protocol Extension 3, the Basic Control Instruction Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 9 contains the requirements applicable to Protocol Extension 4, the Remote Bus Interface Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 10 contains the requirements applicable to Protocol Extension 5, the Serial Time Distribution Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 11 contains the requirements applicable to Protocol Extension 6, the Virtual Circuit Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Section 12 contains the requirements applicable to Protocol Extension 7, the Memory Access Protocol. This section constitutes a formal part of the 4-255 Data Bus Protocol Extension specification.

- Appendix 1 contains recommendations for the allocation of Protocol Extension 1 instruction codes to specific interfaces defined in PSS-04-253, the Specification for the Electrical User Interfaces for use in 4-201 Data Handling Systems.

- Appendix 2 contains recommended techniques to extend the range of registers which can be addressed using Protocol Extension 2.

- Appendix 3 contains a compliance pro-forma for 4-255 Data Bus protocol extensions specified in this document.

3. Definitions and Conventions.

3.1. Definitions.

Compliant: An implementation is said to be compliant with this standard if it meets all of the mandatory requirements and any dependent requirements associated with options which have been selected.

Format: The organisation of data within a data unit. E.g. the arrangement of bits within a word.

Ignore: A Remote Terminal ignores an interrogation if it does not perform the action requested in that interrogation and does not generate a response to that interrogation. Interrogations that are received with errors are ignored.

Mandatory Requirement: A requirement that is always imposed.

Optional Requirement: A requirement that is optional, i.e. may be applied to certain implementations but not to others. An implementation that does not implement a given optional requirement can still be compliant with this standard.

Option Dependent Requirement: A requirement that is mandatory if certain optional requirements are implemented.

Procedure: A sequence of steps performed in order to accomplish a task.

Protocol: A set of rules and formats that determine the behaviour of the entities in the performance of a given function.

Recommendation: A statement containing advice for designers. Recommendations should be followed whenever possible.

Reject: A Remote Terminal rejects an interrogation if it is unable to perform the requested action. The Remote Terminal must generate a response to a rejected interrogation. Rejected interrogations are received without error.

Reserved: A bit or bit-field which is marked "reserved" in this standard has been reserved for use by this or other standards in the 4-201 DHS series. Users must not use these bits.

Unassigned: A bit or bit-field which is marked "unassigned" in this standard has no assigned function and is free to be assigned for user specific purposes.

3.2. Abbreviations.

The following abbreviations are used in this document:

BC	Bus Controller.
bps	Bits per second.
BM	Bus Monitor.
BTB	Block Transfer Bus.
CTMS	Central Time Management System.
DHS	Data Handling System.
DMA	Direct Memory Access.
ET	Elapsed Time.
LTMS	Local Time Management System.
MAP	Memory Access Protocol.
PE	Protocol Extension.
PLA	Programmed Listening Address.
RBI	Remote Bus Interface.
RT	Remote Terminal.
TA	Terminal Address.
VC	Virtual Circuit.
VCI	Virtual Circuit Identifier.

3.3. Conventions.

The following conventions apply throughout this document:

3.3.1. Requirement Numbering and Naming.

All of the requirements explicitly stated in this standard are numbered. The numbering scheme used ensures that the numbered requirements can be identified as being part of this standard.

A requirement number for this standard has the form:

256-n-t (where t = M, O, or D)

or

256-PEx-n-t (where t = M, O, or D)

The leading 256 indicates this standard, PSS-04-256. In the second form above, PEx indicates that the requirement applies to protocol extension x only. In the first form (without the PEx part), the requirement is generic and is not related to any specific protocol extension. The n is the sequence number of the requirement, i.e. the nth generic requirement or the nth requirement associated with protocol extension x. The t represents a final character indicating the type of the requirement. This character is M for mandatory requirements, O for optional requirements, or D for option dependent requirements.

Requirements also have a name that follows the number. The name indicates the subject of the requirement. So, for example, a requirement might be numbered and named as follows:

256-PE1-23-M Protocol Extension 1 Mandatory Requirement

Optional requirements always begin with the word ‘optionally’ to emphasise that they may not always be imposed on a given implementation. Option dependent requirements are mandatory on implementations which elect to apply the option on which they are dependent.

This standard also includes recommendations. These are numbered and named in the same way as requirements but the character R is used in place of M, O, or D, e.g.

256-PE1-7-R Protocol Extension 1 Recommendation

A compliance pro-forma for the protocol extensions defined in this document is provided in Appendix 3. This contains all the requirements and recommendations of this document, identified by their numbers and names, and allows requirement dependencies to be traced. By completing this pro-forma, the implementer not only states his compliance to the standard but also characterises his implementation by indicating which options and recommendations have been selected.

3.3.2. Bit Numbering Convention.

The most significant bit of an n-bit field shall be:

- Numbered bit 0 (zero),
- The first bit transmitted,
- The leftmost bit on a format diagram.

The least significant bit of an n-bit field shall be:

- Numbered bit n-1,
- The last bit transmitted,
- The rightmost bit on a format diagram.

This is illustrated in Figure 1.

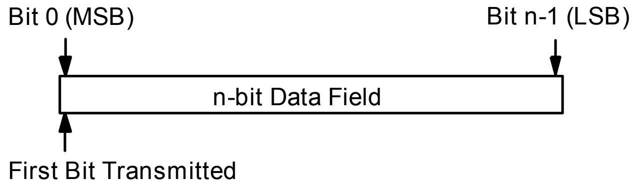

Figure 1 - Bit Numbering Convention.

4. Principles of 4-255 Data Bus Protocol Extensions.

This section discusses a number of aspects of protocol extension use that should be considered when designing 4-255 Data Bus systems and associated terminal equipment.

4.1. Protocol Extension Concepts.

The 4-255 Data Bus protocol extensions are designed to support specific tasks such as transferring blocks of data or the distribution of spacecraft time parameters. Each protocol extension defines:

- The encoding of instructions and their responses which are exchanged between entities across the 4-255 Data Bus,
- The behaviour of the communicating entities with respect to those instructions.

The entities which communicate using a protocol extension are located at different terminal addresses. Controlling entities are located in the Bus Controller and communicate with one or more controlled entities located in Remote Terminals using a given protocol extension as shown inFigure2.

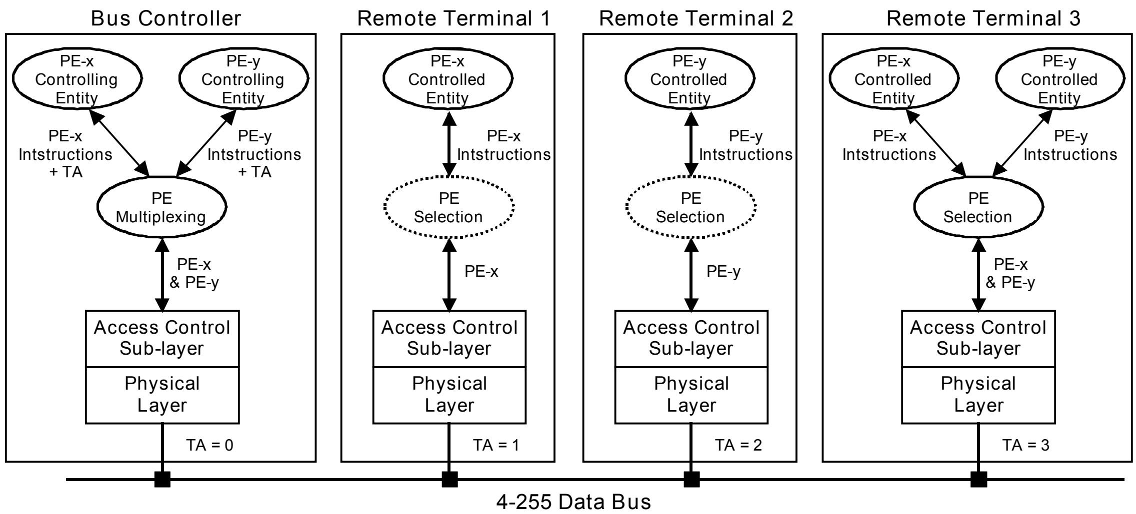

Figure 2 - Use of Protocol Extensions in a 4-255 Data Bus System.

In Figure 2, the Bus Controller hosts PE-x and PE-y controlling entities. Instructions generated by these entities are multiplexed onto the 4-255 Data Bus. The PE-x entity in the Bus Controller controls PE-x entities located in Remote Terminals 1 and 3. To send a PE-x instruction to the PE-x entity in Remote Terminal 1, the PE-x controlling entity must provide the terminal address of Remote Terminal 1 and the instruction, which are then assembled into a single 4-255 Data Bus interrogation word. This interrogation is received by Remote Terminal 1 and the instruction delivered to the PE-x controlled entity. Remote Terminal 3 hosts both PE-x and PE-y controlled entities. Therefore, received interrogations must be routed to the appropriate entity by a protocol extension selection function.

4.2. Choosing a Protocol Extension.

The choice of which protocol extension to use in a given application is influenced by two main factors:

- The suitability of the protocol extension to the function,
- The protocol extension compatibility.

Some protocol extensions are highly specific. For example PE-5, the Serial Time Distribution Protocol, is designed for the distribution of time information to all terminals on the 4-255 Data Bus. Other protocols are more general purpose such as PE-2, the Register Load Instruction Protocol, and PE-7, the Memory Access Protocol. These both perform similar functions and the choice between them will probably be based on other considerations such as the Remote Terminal characteristics.

Where more than one protocol extension will be used on the 4-255 Data Bus, compatibility issues, which are discussed in detail in section 0, must be considered.

The following paragraphs describe the intended applications for the protocol extensions defined in this document:

Protocol Extension 1, The Data Acquisition and Pulse Instruction Protocol , provides instructions for the direct acquisition of data from sensors connected to Remote Terminals and for the issuing of command pulses from Remote Terminals. While this protocol extension is primarily intended for non-intelligent Remote Terminals, it may still be used effectively on other types of Remote Terminal.

Protocol Extension 2, The Register Load Instruction Protocol , is intended for non-intelligent Remote Terminals and provides a mechanism to load a restricted number of registers or memory locations. Special techniques to extend the range of registers or memory locations which can be accessed are described in Appendix 2, however, the use of PE-4 or PE-7 may be more appropriate where a vastly extended addressable range is required.

Protocol Extension 3, The Basic Control Instruction Protocol , is suitable for all types of Remote Terminal and provides a set of commonly needed control instructions for Remote

Terminal operation. This protocol extension prevents the proliferation of Remote Terminal specific instruction sets and encoding schemes. The encoding of the Basic Control Instructions ensures compatibility with the other protocol extensions defined in this document, and allows terminal specific instructions to be incorporated when needed.

Protocol Extension 4, The Remote Bus Interface Protocol, is primarily intended for intelligent and packet Remote Terminals and supports the exchange of blocks of data, such as data packets, between 4-255 Data Bus terminals. Block transfers are managed using a DMA technique which allows the Remote Terminal processor to operate asynchronously with respect to the 4-255 Data Bus.

Protocol Extension 5, The Serial Time Distribution Protocol, provides an extremely efficient and transparent means of distributing spacecraft time information, and other parameters, throughout the spacecraft. A single broadcast bit in the 4-255 Data Bus interrogation word is used to do this. The information conveyed in this protocol includes time and synchronisation data which allows a coherent time reference to be maintained at each 4-255 Data Bus terminal.

Protocol Extension 6, The Virtual Circuit Protocol, provides a virtual circuit capability between 4-255 Data Bus Remote Terminals. The data exchanged over the virtual circuits can be completely unformatted and the amount of data is limited only by the lifetime of the virtual circuit. The bandwidth of each virtual circuit is governed by the Bus Controller, which allocates bus slots for virtual circuit use. Remote Terminals connected via a virtual circuit are not obliged to transfer data in every slot allocated to the virtual circuit, the Bus Controller can therefore allocate bus slots to virtual circuits to make use of unassigned 4-255 Data Bus bandwidth.

Protocol Extension 7, The Memory Access Protocol, is primarily intended for intelligent Remote Terminals and enables a Bus Controller to access the Remote Terminal memory and data buffers. It could also be used in non-intelligent Remote Terminals in place of the Register Load Instruction Protocol to provide an extended range of addressable registers. This protocol is particularly efficient for the exchange of short, variable length messages and may be used in conjunction with the Remote Bus Interface Protocol to manage packet exchanges.

4.3. Protocol Extension Compatibility.

Protocol extension compatibility must be considered in two senses:

- The compatibility of more than one protocol extension at a given Remote Terminal address, i.e. terminal level compatibility,
- The compatibility between protocol extensions used among all of the terminals on a given bus system, i.e. system level compatibility.

4.3.1. Terminal Level Compatibility.

For protocol extensions to be compatible at a given Remote Terminal, that terminal must be able to distinguish between the instructions belonging to the different protocol extensions. Protocol extensions whose encoding schemes do not overlap, i.e. which do not use the same instruction codes, are compatible at a Remote Terminal. However, protocol extensions which each use a given instruction code to mean a different thing, have overlapping encoding schemes and are therefore incompatible at a Remote Terminal.

Table 1 shows how the protocol extension to which a received 4-255 interrogation belongs can be identified. Based on the information in this table, a protocol selection function can be implemented to route received interrogations to the appropriate protocol extension entity in a Remote Terminal.

Interrogation Word Bits→	Broadcast Pulses 3-4-5	Mode Identifier 12-13-14	Instruction Group Identifier 15-16-17-18
PE1 Data Acquisition and Pulse Instruction Protocol	Not Used	000	0000
PE2 Register Load Instruction Protocol	Not Used	≠ 000	Part of data (not used as group identifier)
PE3 Basic Control Instruction Protocol	Not Used	000	0001
PE4 Remote Bus Interface Protocol	Not Used	001	≠ 1111
		101	1xxx
		110	Part of data (not used as group identifier)
		111	
PE5 Serial Time Distribution Protocol	BCP-0 (bit-3) Used	Not used by PE-6	
PE6 Virtual Circuit Protocol	Not Used	000	1111
PE7 Memory Access Protocol	Not Used	001	1111
		010	Part of data (not used as group Identifier)
		011	
		100	
		101	0xxx

Table 1 - Identifying Interrogations from Protocol Extensions Defined in this Document.

The protocol extensions defined in this document are designed to be compatible. The only notable exception to this is that PE2 is not compatible with either PE4 or PE7 since they use the same instruction codes for different purposes. However, PE2 is primarily intended for use in non-intelligent Remote Terminals while PE4 and PE7 are primarily intended for use in packet and intelligent Remote Terminals, and non-intelligent Remote Terminals where the register addressing range of PE2 is not adequate. Table 2 indicates the compatibility of the protocol extensions defined in this standard at a given Remote Terminal address.

	PE1	PE2	PE3	PE4	PE5	PE6	PE7
PE1 Data Acquisition and Pulse Instruction Protocol	-	Yes	Yes	Yes	Yes	Yes	Yes
PE2 Register Load Instruction Protocol	Yes	-	Yes	No	Yes	Yes	No
PE3 Basic Control Instruction Protocol	Yes	Yes	-	Yes	Yes	Yes	Yes
PE4 Remote Bus Interface Protocol	Yes	No	Yes	-	Yes	Yes	Yes
PE5 Serial Time Distribution Protocol	Yes	Yes	Yes	Yes	-	Yes	Yes
PE6 Virtual Circuit Protocol	Yes	Yes	Yes	Yes	Yes	-	Yes
PE7 Memory Access Protocol	Yes	No	Yes	Yes	Yes	Yes	-

Table 2- Compatibility between Protocol Extensions at a Given Remote Terminal Address.

4.3.2. System Level Compatibility.

The system level compatibility of protocol extensions used on the 4-255 Data Bus determines which protocol extensions can be used across the whole data bus system. I.e. whether one protocol extension can be used between the Bus Controller and one Remote Terminal, and a different protocol extension can be used between the Bus Controller and another Remote Terminal. The nature of the 4-255 Data Bus is such that most protocol extensions will be compatible on the bus if they use the terminal data field of an interrogation since this field is only decoded by the specific terminal to which the interrogation is addressed. Many spacecraft already use more than one protocol extension on a single bus, for example those which have a mixture of RTU's and RBI based terminals.

However, problems can occur if interrogation broadcasting is used. Broadcast interrogations are effectively addressed to every Remote Terminal on the bus simultaneously and the terminal data fields of those interrogations may therefore be decoded by every Remote Terminal (although no terminals are permitted to respond). Therefore a Remote Terminal could receive an instruction belonging to an incompatible protocol extension if that instruction is broadcast.

This problem only occurs if protocol extensions which would be incompatible at a given Remote Terminal are implemented (see Table 2). If this is the case, conflicts can be avoided either by prohibiting the broadcast transmission of interrogations from the Bus Controller, or by selectively disabling broadcast reception in the Remote Terminals which may misinterpret broadcast interrogations. Broadcast reception disabling is a feature of 4-255 Data Bus modems which comply with PSS-04-257, the 4-255 Data Bus Modem Specification, and instructions to control this facility are defined in Protocol Extension 3, the Basic Control Instruction Protocol.

Protocol extensions which use the broadcast field of an interrogation may also pose a system compatibility problem. Since the broadcast field bits are decoded by all Remote Terminals, the use of a given broadcast field bit by a protocol extension makes that bit unavailable for any other use. For example, if the Serial Time Distribution Protocol (PE5) defined in this document is operated on a 4-255 Data Bus system, it requires exclusive use of the BCP0 broadcast field bit. BCP0 could therefore not be used for any other purpose.

4.4. Use of the Response Preamble Field.

The 4-255 Data Bus response preamble field comprises four bits of protocol control information as shown in Figure 3.

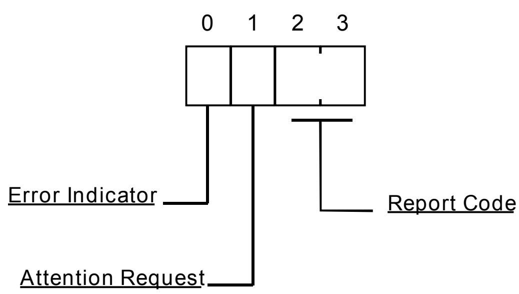

Figure 3 - The 4-255 Data Bus Response Preamble Field.

This field can be accessed and modified by entities in different protocol layers in order to signal a corresponding entity at the other end of a communication path. The use of the various components of the preamble field is as follows.

The Error Indicator bit is set to one to indicate that an error has been detected during the processing of the interrogation which invoked the response, or in the generation of the response itself. An error indicator bit value of one may imply that the response data field content is not reliable or that the response data field contains an error code instead of data.

The Attention Request bit is set to one to indicate that the Remote Terminal generating the response requires servicing by the Bus Controller. This bit may either be set as a result of the execution of the interrogation which invoked the response, or as a result of some other event within the Remote Terminal which requires Bus Controller attention.

The Report Code consists of two bits which can provide additional information about the response. If the error bit is set, the report code provides information to indicate the cause of the error. If the error bit is not set, the report code may provide information qualifying the response data field, e.g. to indicate the position in a formatted data unit such as first word, last word, data word etc.

The response preamble field is accessible to entities within different protocol layers of the 4-201 DHS Reference Model and these entities may not be directly concerned with the implementation of the protocol extension. Sections 0 to 0 describe the use of the response preamble field in more detail.

4.4.1. Undeliverable Instruction Response.

An undeliverable instruction response is generated when an interrogation is sent to a Remote Terminal but that Remote Terminal cannot deliver the interrogation to an appropriate protocol extension entity. This may occur if the Remote Terminal does not implement the protocol extension to which the instruction belongs or if the protocol extension entity is not operational. To indicate that an interrogation is undeliverable, the response preamble field error bit is set to 1 and the report code is set to 11, the response data field is set to zero, and the response parity bit is set to zero.

Figure 4 shows an example of how this response can be generated by a protocol selection function in a Remote Terminal which implements more than one protocol extension.

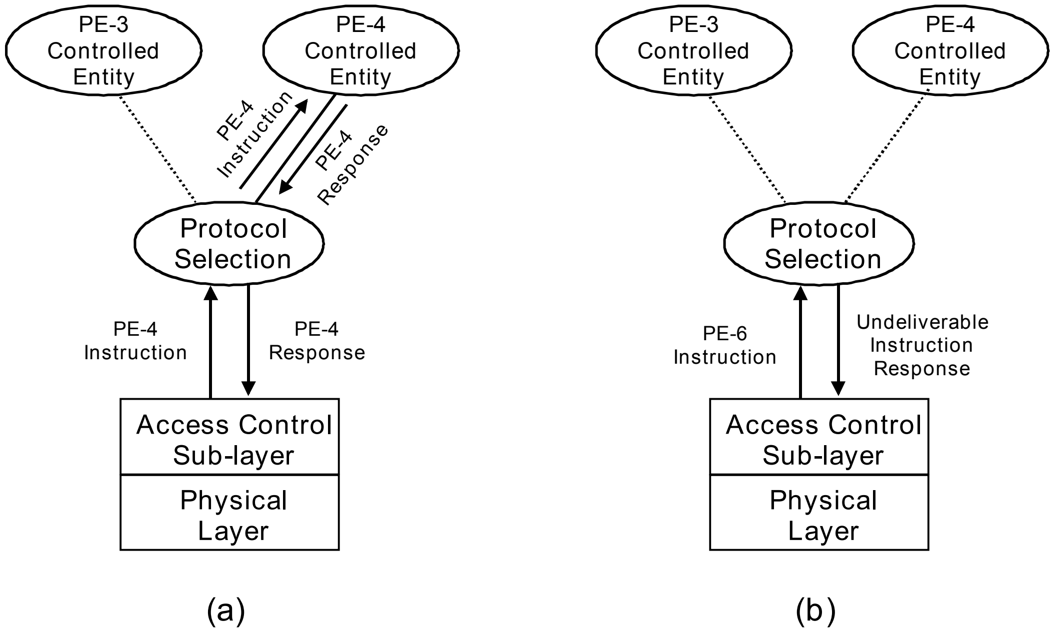

Figure 4 - Generation of Normal Response (a) and Undeliverable Instruction Response (b) by a Remote Terminal Implementing Multiple Protocol Extensions.

4.4.2. Late Response Report.

A late response report may be generated within the Access Control Sub-layer if a response to an interrogation is expected from a higher layer (e.g. a protocol extension entity) but is not received within the response deadline. To indicate a late response, the response preamble field error bit is set to 1 and the report code set to 11, the response data field is set to all ones, and the response parity bit is set to zero. If a response is subsequently received from the higher layer, it is discarded within the Access Control Sub-layer.

Figure 5 illustrates how a late response report is generated in the Access Control Sub-layer.

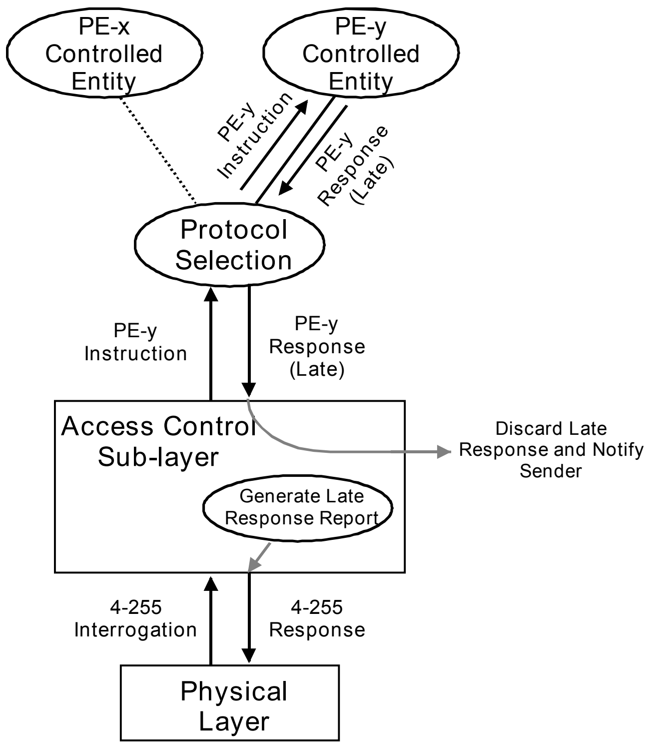

Figure 5 - Generation of a Late Response Report in the Access Control Sub-layer.

4.4.3. Report Codes Set by the Protocol Extension.

If an instruction can be delivered to the protocol extension entity, the response preamble field can be used by the protocol extension itself to indicate error conditions. For example, a PE3 Basic Control Instruction Protocol implementation may not implement all of the instructions specified. Therefore an instruction interpreter in a PE3 protocol entity could receive instructions which it does not recognise. PE3 specifies a means of reporting this occurrence using the response preamble field (specifically by setting the response preamble field error bit to 1 and the report code to 10).Figure 6 shows the generation of this report code.

Other protocol extensions include definitions of report codes to convey appropriate response qualifying information such as indicating the position of the word within a formatted data unit (first, last, continuation) or indicating the type of word (data, control).

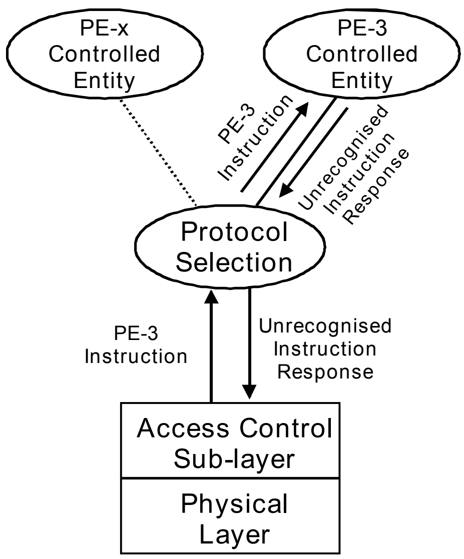

Figure 6 - Generation of an Unrecognised Instruction Response by a PE3 Implementation.

4.5. Response Preamble Field Attention Request Bit.

The command/response nature of the 4-255 Data Bus means that Remote Terminals can only request the attention of the Bus Controller if they are polled, i.e. if an explicit interrogation is sent to the Remote Terminal. With the increasing trend of multi-function Remote Terminals, it is desirable to allow individual functions in a Remote Terminal to request attention from the Bus Controller without having to be explicitly polled. This can be achieved using an attention request flag in the Remote Terminal and conveying the value of this flag in the attention request bit of the response preamble field.

When the response preamble field attention request flag is set to one it is interpreted by the Bus Controller to mean that some function in the Remote Terminal requires servicing. The Bus Controller can then explicitly poll the terminal to determine the function that requires servicing. This obviates the need for the Bus Controller to poll functions unnecessarily.

> Note: An efficient method for a Remote Terminal to identify the function requiring attention to the Bus Controller is to use the unassigned bits of a basic status response as a service request vector. The Bus Controller can then identify the service requestor by sending a single PE-3 RT_STATUS instruction.

The attention request bit should be set within the Access Control Sub-layer since this is the lowest layer which has access to the response preamble field. This ensures that the attention request bit is not overwritten if the preamble field is modified, for example if a late response report is generated. Therefore, if a protocol extension entity requires service from the Bus Controller it should do so via the Access Control Sub-layer and not by asserting the attention request bit itself. Figure 7 shows the insertion of the attention request flag at the Access Control Sub-layer.

Remote Terminals which maintain an attention request flag should clear that flag when they are reset. However, there is nothing to prevent that flag being re-asserted as soon as the terminal has completed its reset cycle so that the attention request bit of the first response received from a Remote Terminal is set.

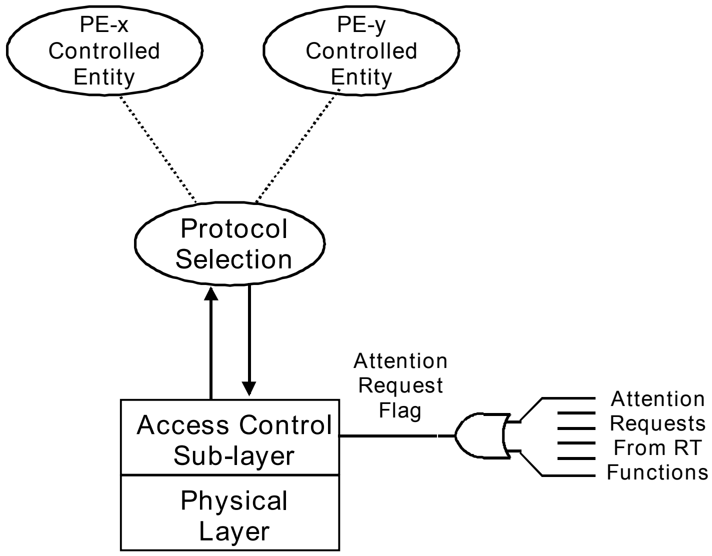

Figure 7 - Insertion of Attention Request Flag at the Access Control Sub-layer.

4.6. Use of the Response Parity Bit.

The response parity bit can be used to provide end-to-end protection for the data field of a 4-255 Data Bus response. The level of protection provided by the parity bit is minimal if the data is only being passed across the 4-255 Data Bus, since responses are already well protected by the Litton encoding used on the bus. However, if other low level links are used between the Access Control Sub-layer and the protocol extension entity, such as a serial link as shown in Figure 8, the parity bit offers protection for the response data across that link and may therefore provide significantly improved end-to-end error protection.

Protocol extensions which use the response data field, and can therefore take advantage of parity protection, are specified with two parity modes, even and none. When the parity mode is even, the response parity bit is set so that the sum of all the 1 bits in the response data field plus the response parity bit is even. When the parity mode is set to none the response receiving entity does not check for even parity so the response parity bit may be set to 1 or 0. The parity mode to be used between protocol extension entities must be agreed between both ends of the communication link, i.e. both communicating entities must use the same mode. The same parity mode need not be used for all end-to-end communications with a given protocol extension. For example a PE-x entity in the Bus Controller may use a parity mode of even when communicating with a PE-x entity in one Remote Terminal and a parity mode of none when communicating with a PE-x entity in another Remote Terminal. However, care should be taken when direct Remote Terminal to Remote Terminal communication is used and in most cases the same parity mode will be used by all entities using a given protocol extension.

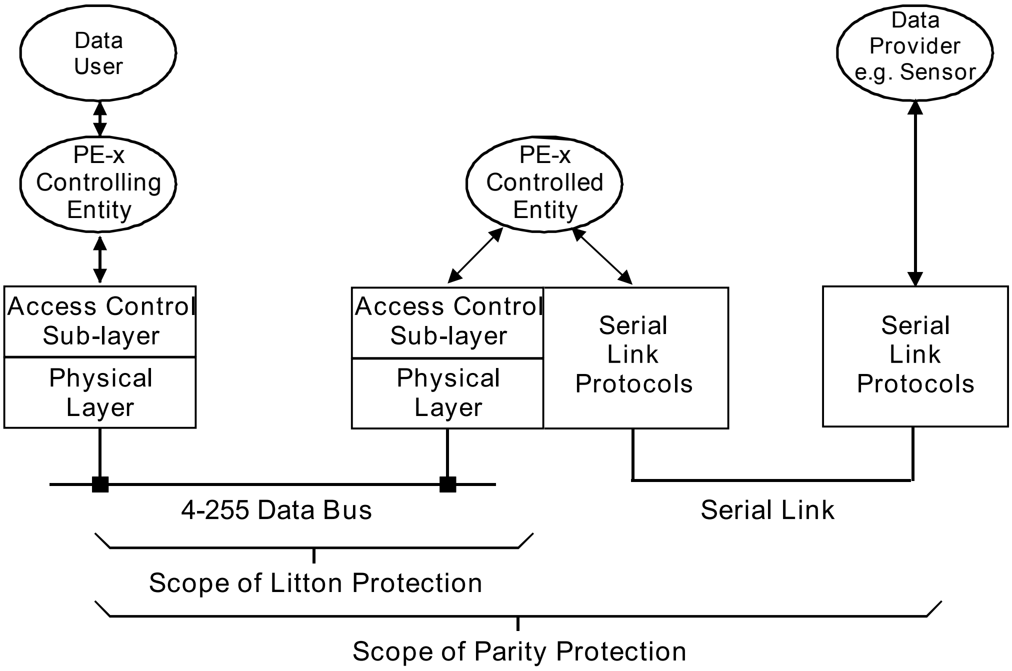

Figure 8 - End-to-end Parity Protection over Multiple Links.

When the parity mode between two protocol extension entities is set to none, the response parity bit is not modified or checked during protocol extension exchanges. The response parity bit can therefore be treated as an extra data bit which is carried by the protocol extension. This bit could be used to carry a parity bit generated within a higher layer and this may provide better protection than using parity at the level of the protocol extension. Figure 9 shows how parity protection applied to the protocol extension may still fail to protect data transferred between higher layer entities. Figure 10 shows how setting the parity mode between protocol extension entities allows parity protection to be applied at a higher protocol layer.

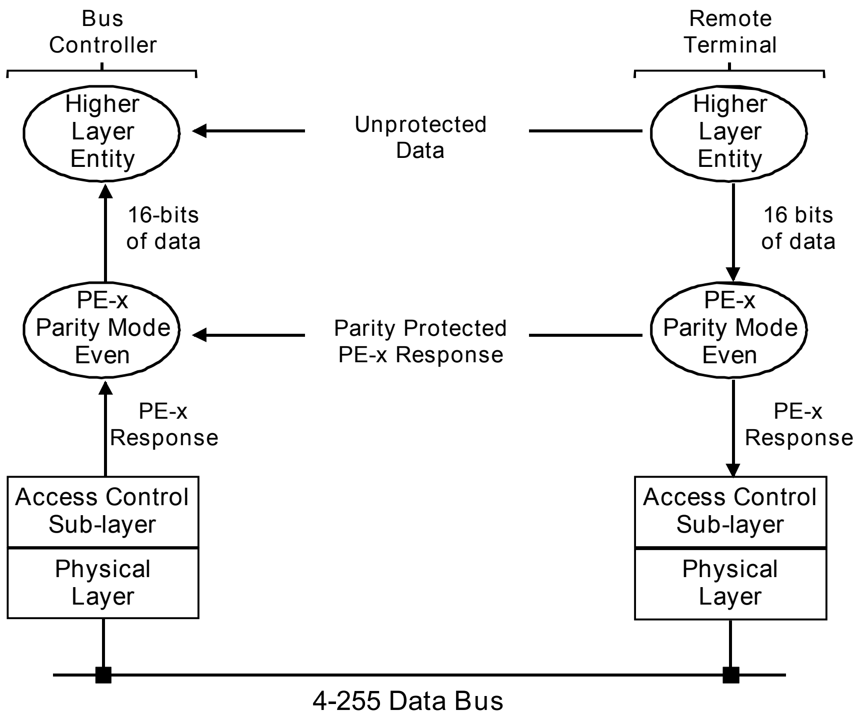

Figure 9 - Use of Even Parity at Protocol Extension Level.

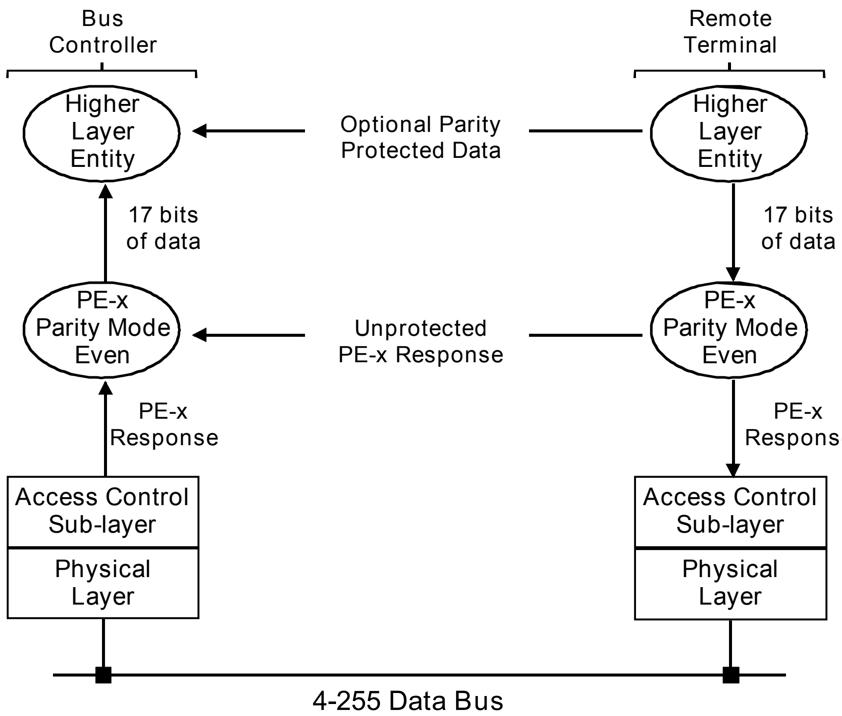

Figure 10 – Optional use of Parity in Higher Protocol Layers with PE Parity Mode Set to None.

4.7. Implications for the Bus Controller.

The use of 4-255 Data Bus protocol extensions has a number of special implications for the Bus Controller which are discussed here.

4.7.1. Handling of Special Responses.

The slot correlation function implemented in the Access Control Sub-layer of a 4-255 Data Bus Bus Controller ensures that the responses to protocol extension instructions are returned to the appropriate protocol extension controlling entity. However, the Bus Controller must examine the response to check for the special responses indicating an undeliverable instruction or a late response. Handling these special responses once they are detected may effect the system behaviour.

Normally, an undeliverable instruction response should be returned to the protocol extension controlling entity which provided the instruction. This protocol extension controlling entity will have provided the terminal address to which the instruction was sent and therefore expected a controlled entity at that location. If an interrogation containing an instruction to the controlled entity is not deliverable, it may imply either that that controlled entity is not functioning correctly, or that it is not present, and these situations should be handled by the controlling entity. However, an undeliverable instruction response can also imply a general failure in the addressed Remote Terminal, a situation which should not be handled by a single protocol extension controlling entity but may require the intervention of an overall system controller. Therefore, some consideration must be given to the best way of handling an undeliverable instruction response in a given 4-255 Data Bus system. The protocol extension controlling entity must always be notified that an instruction resulted in an undeliverable response report.

A late response report indicates that the entity to which a protocol instruction was sent could not generate a response within the response deadline imposed by the Access Control Sub-layer. This may be a single isolated event or it may imply a general problem in the addressed Remote Terminal that will result in frequent late response reports being generated. In either case, the response, including the preamble, should be returned to the controlling entity which can then act accordingly, e.g. by repeating the instruction or by enquiring the status of the controlled entity. To determine whether a Remote Terminal is persistently generating late response reports, the Bus Controller may need to record the number of such reports received per Remote Terminal over a given time interval.

4.7.2. Handling of the Attention Request Bit.

The attention request bit of the response preamble field is available for any function within a Remote Terminal to request the attention of the Bus Controller and is set at the Access Control Sub-layer in the Remote Terminal. The Bus Controller must therefore check this bit at the Access Control Sub-layer, i.e. before passing a response to the protocol extension controlling entity, and must perform the appropriate actions to service the Remote Terminal. The protocol extension controlling entity need not examine this bit in a received response.

4.7.3. Parity Modes.

A protocol extension controlling entity operating within a Bus Controller may be communicating with many controlled entities located in Remote Terminals. The response parity mode which is used between the protocol extension controlling entity and each controlled entity may be either ‘even’ or ‘none’. Since the parity mode used at a given controlled entity is fixed and mode information cannot be explicitly communicated between the controlled and controlling entities, the controlling entity must use the appropriate mode for exchanges with each controlled entity.

Typically, a controlling entity will maintain a table of attributes for each controlled entity with which it communicates, containing the terminal address of the Remote Terminal hosting the controlled entity together with other protocol extension specific attributes (such as address ranges, etc). This table must also include the response parity mode as an attribute.

5. General Protocol Extension Requirements.

The requirements contained in this section are general requirements relating to 4-255 Data Bus protocol extensions.

5.1. Protocol Extension Implementation on 4-255 Data Bus Terminals.

256-1-O Implementation of PE-1, Data Acquisition and Pulse Instruction Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 1, the Data Acquisition and Pulse Instruction Protocol.

256-2-O Implementation of PE-2, Register Load Instruction Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 2, the Register Load Instruction Protocol.

256-3-O Implementation of PE-3, Basic Control Instruction Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 3, the Basic Control Instruction Protocol.

256-4-O Implementation of PE-4, Remote Bus Interface Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 4, the Remote Bus Interface Protocol.

256-5-O Implementation of PE-5, Serial Time Distribution Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 5, the Serial Time Distribution Protocol.

256-6-O Implementation of PE-6, Virtual Circuit Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 6, the Virtual Circuit Protocol.

256-7-O Implementation of PE-7, Memory Access Protocol

Optionally, a 4-255 Data Bus terminal shall implement Protocol Extension 7, the Memory Access Protocol.

256-8-M Remote Terminal – Prohibited use of PE-2 with PE-4

A Remote Terminal shall not simultaneously implement both Protocol Extension 2, the Register Load Instruction Protocol, and Protocol Extension 4, the Remote Bus Interface Protocol.

256-9-M Remote Terminal – Prohibited use of PE-2 with PE-7

A Remote Terminal shall not simultaneously implement both Protocol Extension 2, the Register Load Instruction Protocol, and Protocol Extension 7, the Remote Bus Interface Protocol.

256-10-D Remote Terminal – Continued Servicing of PE-3 Instructions when Halted

A Remote Terminal which implements Protocol Extension 3, the Basic Control Instruction Protocol, shall continue to service PE-3 instructions when the terminal has been halted using an RT_HALT instruction.

> Note: This requires that the protocol stack, up to and including the PE-3 controlled entity, remains operational when the Remote Terminal is halted, and ensures that the Remote Terminal can be made to resume activity using a PE-3 RT_RESUME instruction.

256-11-M Remote Terminal – Protocol Selection Function

A Remote Terminal which simultaneously implements more than one protocol extension shall implement a protocol selection function which routes received interrogations to the appropriate protocol extension entity.

256-12-O Remote Terminal – Generation of Undeliverable Instruction Response

Optionally, a Remote Terminal shall generate an undeliverable instruction response if it cannot deliver a received interrogation to an appropriate protocol extension entity.

256-13-O Remote Terminal – Generation of Late Response Report

Optionally, a Remote Terminal which does not receive a response to a valid interrogation from the terminal equipment at the Access Control Sub-layer within the response deadline shall generate and transmit a late response report.

> Note: The 4-255 Data Bus Access Control Sub-layer is normally implemented by a Remote Terminal modem.

5.2. 4-255 Data Bus Responses.

256-14-M Response Word – Format

The response word shall use the 4-255 Data Bus standard response word format consisting of:

- A 4-bit response preamble field,
- A 16-bit response data field,
- A response parity bit.

The response word format is shown in Figure 11.

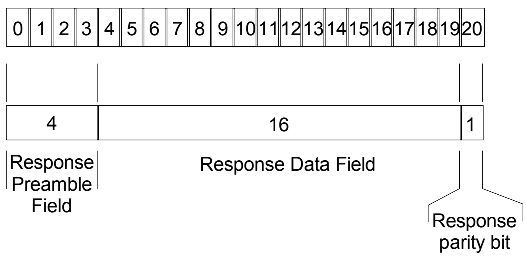

Figure 11 - 4-255 Data Bus Response Word Format.

256-15-M Response Preamble Field

The response preamble field occupies bits 0, 1, 2 and 3 of the response word and shall consist of:

- An error indicator bit,
- An attention request bit,
- A 2-bit report code.

The preamble field format is shown in Figure 12.

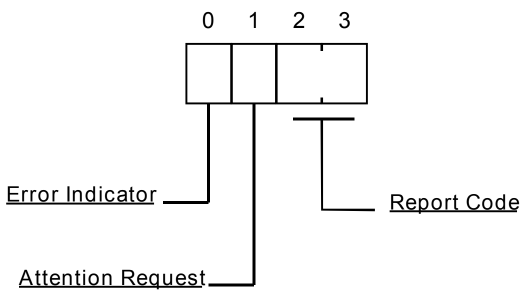

Figure 12 - Response Preamble Field Format.

256-16-M Response Preamble Field - Error Indicator

The response preamble field error indicator bit (bit 0) shall be set to one (1) to indicate that an error occurred during the execution of the instruction contained in the corresponding interrogation word. If no errors are detected, this bit shall be set to zero (0).

256-17-M Response Preamble Field - Attention Request

The response preamble field attention request bit (bit 1) shall be set to one (1) to indicate that the Remote Terminal that generated the response requires attention from the Bus Controller. Otherwise, this bit shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-18-M Response Preamble Field –Report Code

The response preamble field report code (bits 2 and 3) shall be used to provide qualifying information pertaining to the response data field and shall be interpreted in conjunction with the response preamble field error indicator.

256-19-M Response Data Field

The 16-bit response data field (bits 4 to 19 inclusive) shall be used to convey the Remote Terminal data which is generated in response to the corresponding interrogation.

Note: This can include error codes generated if the instruction contained in the interrogation cannot be correctly executed or if the interrogation cannot be delivered.

256-20-M Response Parity Bit

The response parity bit (bit 20) shall be used for either of the following:

- As an even parity bit covering the 16-bit response data field (PE response parity mode 'even'),
- As an additional (i.e. seventeenth) data bit in conjunction with the response data field (PE response parity mode 'none').

5.2.1. Undeliverable Instruction Response.

256-21-O Undeliverable Instruction Response

An Undeliverable Instruction Response shall comprise a 4-255 Data Bus response with the following response field values:

- Response preamble field error indicator (bit 0) set to 1,
- Response preamble field report code (bits 2 and 3) set to 11,
- Response data field (bits 4 to 19 inclusive) all set to 0,
- Response parity bit (bit 20) set to 0.

Note: The value of the response preamble field attention request bit is not specified as it is set independently of the response type.

5.2.2. Late Response Report.

256-22-M Late Response Report

A Late Response Report shall comprise a 4-255 Data Bus response with the following response field values:

- Response preamble field error indicator (bit 0) set to 1,
- Response preamble field report code (bits 2 and 3) set to 11,
- Response data field (bits 4 to 19 inclusive) all set to 1,
- Response parity bit (bit 20) set to 0.

Note: The value of the response preamble field attention request bit is not specified as it is set independently of the response type.

6. Protocol Extension 1 - Data Acquisition and Pulse Instruction Protocol.

This protocol extension defines the protocol for acquiring data from Remote Terminal user interface inputs and for distributing pulse signals from Remote Terminal user interface outputs.

Data acquisition using Protocol Extension 1 is on a word-by-word basis, i.e. each PE-1 data acquisition instruction is distinct from any preceding or following instructions and can acquire exactly one word of data.

6.1. Data Acquisition and Pulse Instructions - General Requirements.

This section defines a set of general requirements which must be met by Protocol Extension 1 entities.

6.1.1. Data Acquisition Instructions.

256-PE1-1-D Data Acquisition Instructions - Use

A data acquisition instruction shall be issued by the Bus Controller in order to acquire data from a specified data acquisition channel at the addressed Remote Terminal.

256-PE1-2-D Data Acquisition Instructions - Rejection Criteria

A Remote Terminal shall reject a data acquisition instruction if the channel identifier of the instruction does not match the channel identifier of any of the data input channels implemented on that Remote Terminal.

256-PE1-3-D Data Acquisition Instructions - Action on Receipt, Instruction Not Rejected

A Remote Terminal that receives a data acquisition instruction and does not reject it shall acquire a data value from the specified channel and shall generate a 4-255 Data Bus response.

Note: The acquired data value must be packed into a 16-bit data word.

256-PE1-4-D Data Acquisition Instructions - Action on Receipt, Instruction Rejected

A Remote Terminal that receives a data acquisition instruction and rejects it shall not acquire any data but shall generate a 4-255 Data Bus response.

256-PE1-5-D Data Acquisition Instructions - Response

The response to a data acquisition instruction shall contain the data acquired as a result of the data acquisition instruction and shall indicate whether the instruction was rejected and whether errors were detected during execution of the instruction.

256-PE1-6-O Data Acquisition Instructions – Optional Response Parity Mode

Optionally, the PE-1 data acquisition instruction response parity mode at a given Remote Terminal shall be 'Even'.

Note 1: A response parity mode of 'Even' means that the data acquisition instruction function (part of the PE-1 entity) in the Remote Terminal must generate and insert an even parity value into the response parity bit. This value is checked in the PE-1 entity receiving the response to verify the quality of the response data.

Note 2: By default, the PE-1 data acquisition instruction response parity mode at a Remote Terminal is 'None'.

6.1.2. Pulse Instructions.

256-PE1-7-D Pulse Instructions - Use

A pulse instruction shall be issued by the Bus Controller in order to provoke the generation of an output pulse on the specified output channel of the addressed Remote Terminal.

256-PE1-8-D Pulse Instructions - Mandatory Rejection Criteria

A Remote Terminal shall reject a pulse instruction if either of the following conditions apply:

- A pulse is already being output on the specified output channel,
- The channel identifier of the pulse instruction does not match the channel identifier of any of the pulse output channels implemented on that Remote Terminal.

256-PE1-9-O Pulse Instructions - Optional Rejection Criteria

Optionally, a Remote Terminal shall reject a pulse instruction if it is already outputting a pulse on any channel.

256-PE1-10-D Pulse Instructions - Action on Receipt, Instruction Not Rejected

A Remote Terminal that receives a pulse instruction and does not reject it shall output a pulse on the specified channel and shall generate a 4-255 Data Bus response.

256-PE1-11-D Pulse Instructions - Action on Receipt, Instruction Rejected

A Remote Terminal that receives a pulse instruction and rejects it shall not output a pulse on any channel but shall generate a 4-255 Data Bus response.

256-PE1-12-D Pulse Instructions - Response

The response to a pulse instruction shall contain an acknowledgement code and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the pulse instruction.

256-PE1-13-O Pulse Instructions – Optional Response Parity Mode

Optionally, the PE-1 pulse instruction response parity mode at a given Remote Terminal shall be ‘Even’.

> Note 1: A response parity mode of ‘Even’ means that the pulse instruction function (part of the PE-1 entity) in the Remote Terminal is required to generate and insert an even parity value covering the response data field into the response parity bit. The parity value is checked by the PE-1 entity in the Bus Controller to verify the quality of the data contained in the response data field.
>
> Note 2: By default, the PE-1 pulse instruction response parity mode at a Remote Terminal is ‘None’.

6.2. Data Acquisition and Pulse Instructions - Interrogation Word Formats.

In PE-1 the terminal data field of the 4-255 Data Bus interrogation word is divided into the following fields:

- A 3-bit mode identifier field,
- A 4-bit instruction group identifier field,
- A 1-bit instruction type identifier field,
- A 3-bit instruction code field,
- An 8-bit channel identifier field.

This format, within the complete 4-255 Data Bus interrogation word, is shown in Figure 13.

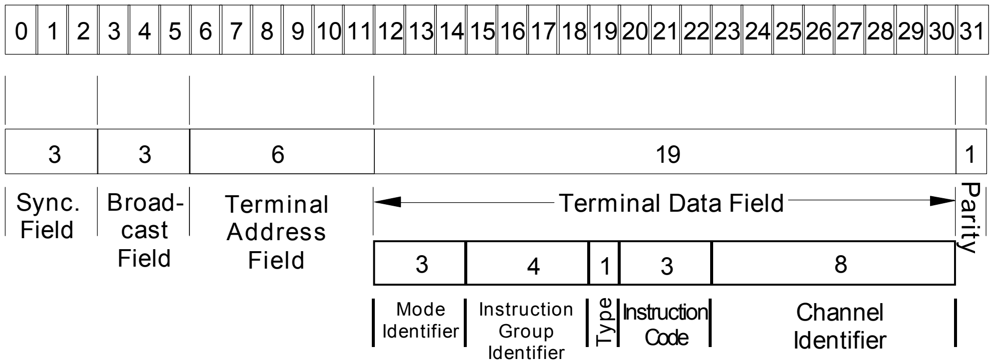

Figure 13 - Data Acquisition and Pulse Instruction Interrogation Word Format.

256-PE1-14-D Data Acquisition and Pulse Instructions - Mode Identifier Field

The mode identifier field occupies bits 12, 13 and 14 of the interrogation word and shall be set to all zeroes (000) to indicate that the interrogation is a data acquisition or a pulse instruction.

256-PE1-15-D Data Acquisition and Pulse Instructions - Instruction Group Identifier Field

The 4-bit instruction group identifier field occupies bits 15, 16, 17 and 18 of the interrogation word. This field shall be set to all zeroes (0000).

256-PE1-16-D Data Acquisition and Pulse Instructions - Instruction Type Identifier Field

The instruction type identifier bit, bit 19 of the interrogation word, shall be set to zero (0) to indicate a pulse instruction or one (1) to indicate a data acquisition instruction.

256-PE1-17-D Data Acquisition and Pulse Instructions - Instruction Code Field

The 3-bit instruction code field occupies bits 20, 21 and 22 of the interrogation word. This code shall be interpreted by the Remote Terminal as the channel instruction code and shall be used in conjunction with the instruction type identifier to determine the operation to be performed.

256-PE1-18-D Data Acquisition and Pulse Instructions - Instruction Code Assignments

The data acquisition and pulse instruction code assignments shall be as shown in Table 3.

Instruction Type Identifier	Instruction Code			Remote Terminal Operation
Bit 19	Bit 20	Bit 21	Bit 22	
0	0	0	0	Pulse Instruction Set 0
0	0	0	1	Pulse Instruction Set 1
0	0	1	0	Pulse Instruction Set 2
0	0	1	1	Pulse Instruction Set 3
0	1	0	0	Pulse Instruction Set 4
0	1	0	1	Pulse Instruction Set 5
0	1	1	0	Pulse Instruction Set 6
0	1	1	1	Pulse Instruction Set 7
1	0	0	0	Data Acquisition Instruction Set 0
1	0	0	1	Data Acquisition Instruction Set 1
1	0	1	0	Data Acquisition Instruction Set 2
1	0	1	1	Data Acquisition Instruction Set 3
1	1	0	0	Data Acquisition Instruction Set 4
1	1	0	1	Data Acquisition Instruction Set 5
1	1	1	0	Data Acquisition Instruction Set 6
1	1	1	1	Data Acquisition Instruction Set 7

Table 3 - Data Acquisition and Pulse Instruction Instruction Codes.

Note: Appendix 1 contains recommendations for the allocation of instruction sets to electrical user interface types.

256-PE1-19-D Data Acquisition and Pulse Instructions - Channel Identifier Field

The channel identifier field occupies bits 23 to 30, inclusive, of the interrogation word. This field shall contain an 8-bit code identifying the (user interface) channel on the Remote Terminal from which data is to be acquired or on which a pulse is to be output.

Under this encoding scheme, the instruction codes identify groups of channels. Each group can comprise up to 256 uniquely identifiable channels.

6.3. Data Acquisition and Pulse Instructions - Response Word Formats.

The response word used by Protocol Extension 1 complies with the 4-255 Data Bus 21-bit response word format shown in Figure 14.

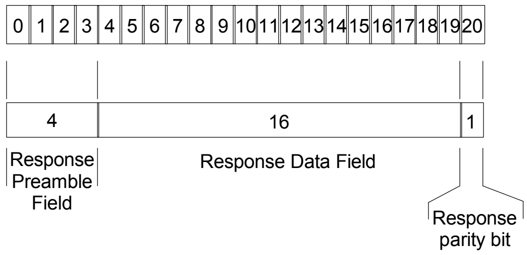

Figure 14 - Data Acquisition and Pulse Instruction Response Word Format.

256-PE1-20-D <u>Data Acquisition and Pulse Instructions - Response Preamble Field</u>

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. These four bits shall be used to indicate data quality, to report errors, and to request Bus Controller attention.

The format of the 4-255 Data Bus response preamble field for Protocol Extension 1 is shown in Figure 15.

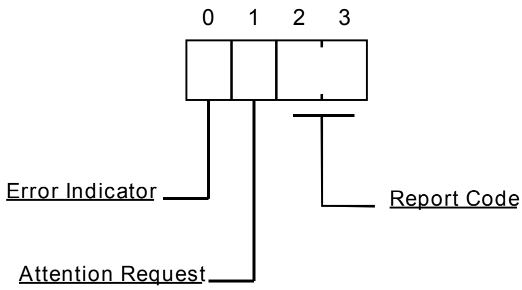

Figure 15 - Data Acquisition and Pulse Instruction - Response Preamble Field.

256-PE1-21-D <u>PE-1 Response Preamble Field - Error Indicator</u>

The response preamble field error indicator bit (bit 0) shall be set to zero (0) if a data acquisition or pulse instruction is not rejected and the requested operation is completed successfully. This bit shall be set to one (1) if the data acquisition or pulse instruction is rejected or if an error is detected during the execution of the requested operation.

256-PE1-22-D <u>PE-1 Response Preamble Field - Attention Request</u>

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE1-23-D <u>PE-1 Response Preamble Field - Report Code, No Error</u>

If no error occurs, the response preamble field report code field (bits 2 and 3) is unassigned and by default shall be set to zero (00). This field may be used for user defined report codes.

256-PE1-24-O <u>PE-1 Response Preamble Field - Report Code, Error</u>

Optionally, if an error occurs, the Remote Terminal shall set the response preamble field report code to indicate the type of error as per Table 4. If an error occurs and the Remote Terminal does not indicate the type of error, the preamble field report code shall be set to zero (00) to indicate an unspecified error.

Response Preamble Field				
Error Indicator	Attention Request	Report Code		Meaning
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	Data Acquisition Error
1	X	1	1	Reserved

Table 4 - Data Acquisition and Pulse Instruction - Response Preamble Field Report Codes.

Note 1: An 'unspecified error' report code (00) may indicate that an error code is contained in the data field.

Note 2: The 'instruction rejected' report code (01) indicates that the instruction has been rejected and was not executed.

Note 3: The 'data acquisition error' report code (10) indicates that data could not be acquired from the specified acquisition channel.

256-PE1-25-D PE-1 Response Word - Data Field, No Error

The response data field occupies bits 4 to 19, inclusive, of the response word. If a data acquisition or pulse instruction is executed without errors, this 16-bit field shall be used to carry up to 16 bits of acquired data in the case of a data acquisition instruction, or a pulse instruction acknowledgement code (all response data field bits set to zero) in the case of a pulse instruction.

256-PE1-26-O PE-1 Response Word - Optional Data Field Error Codes

Optionally if errors are detected during the execution of a data acquisition or pulse instruction and the response preamble field error indicator has been set to 1, the 16-bit response data field (bits 4 to 19 inclusive) shall be used to carry a user defined error code.

256-PE1-27-D PE-1 Response Word - Parity Bit

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

Note: The parity mode is set independently for data acquisition instructions and for pulse instructions. The parity bit value in a given response therefore depends on the type of instruction (data acquisition or pulse instruction) provoking the response.

256-PE1-28-D Response Parity Bit Value - Response Parity Mode = Even

When the response parity mode is ‘Even’, the parity bit shall be set so that the sum of the parity bit plus all ‘one’ bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is ‘None’, the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

7. Protocol Extension 2 - Register Load Instruction Protocol.

Protocol Extension 2 defines the protocol used to load register locations on a Remote Terminal.

Up to seven register load locations can be addressed directly at each Remote Terminal address using Protocol Extension 2.

> Note: Methods which can be used to extend the number of addressable registers are described in Appendix 2.

7.1. Register Load Instructions - General Requirements.

This section defines a set of general requirements which must be met by Protocol Extension 2 entities.

256-PE2-1-D Register Load Instructions - Use

A register load instruction shall be issued by the Bus Controller in order to load a 16-bit data value into a register located at the addressed Remote Terminal.

256-PE2-2-D Register Load Instructions - Rejection Criteria

A Remote Terminal that receives a register load instruction but does not implement a register at the register load address specified in that instruction shall reject that register load instruction.

256-PE2-3-D Register Load Instructions - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives a register load instruction and does not reject it shall transfer the 16-bit data field of that instruction into the register identified by the register load address and shall generate a 4-255 Data Bus response.

256-PE2-4-D Register Load Instructions - Action on Receipt, Instruction Rejected

A Remote Terminal that receives a register load instruction and rejects it shall not transfer the 16-bit data field of that instruction into the register identified by the register load address but shall generate a 4-255 Data Bus response.

256-PE2-5-D Register Load Instruction - Response

The response to a register load instruction shall contain an acknowledgement code and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the register load instruction.

256-PE2-6-O Register Load Instruction – Optional Response Parity Mode

Optionally, the PE-2 response parity mode at a given Remote Terminal shall be 'Even'.

Note 1: A response parity mode of 'Even' means that the terminal is required to generate an even parity value covering the response data field of PE-2 responses. An even parity value is used by the PE-2 entity receiving the response to verify the quality of the response data.

Note 2: By default, the PE-1 response parity mode at a Remote Terminal is 'None'.

7.2. Register Load Instructions - Interrogation Word Format.

In Protocol Extension 2, the terminal data field of the 4-255 Data Bus interrogation word is divided into the following fields:

- A 3-bit register load address field
- A 16-bit register load data field

This format, in relation to the complete 4-255 Data Bus interrogation word, is shown in Figure 16.

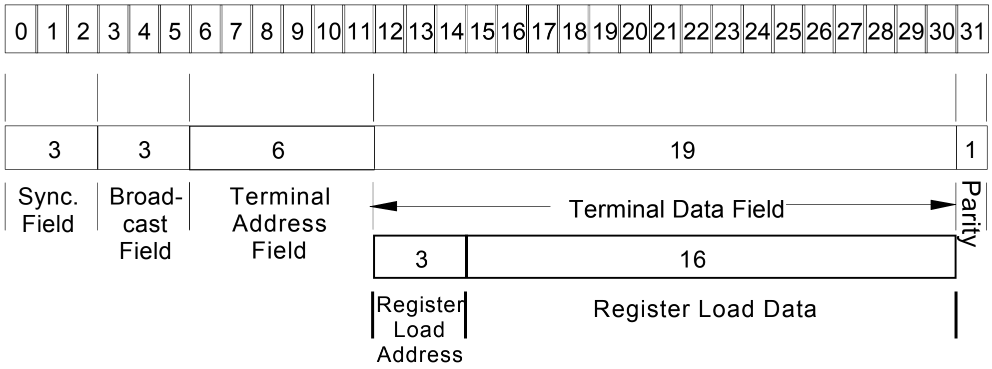

Figure 16 - Register Load Instruction Interrogation Word Format.

256-PE2-7-D <u>Register Load Instruction - Register Load Address Field</u>

The register load address field occupies bits 12, 13 and 14 of the interrogation word. This field shall contain a 3-bit address identifying the register into which the data contained in the interrogation is to be loaded. The register load address value zero (000) is reserved and shall not be used. All other 3-bit values in this field are used by this protocol extension to give a total of seven directly addressable register load locations at each terminal address.

256-PE2-8-D <u>Register Load Instruction - Register Load Data Field</u>

The register load data field occupies bits 15 to 30, inclusive, of the interrogation word. This field shall contain the 16 bits of data which are to be written into the specified location of the addressed Remote Terminal.

7.3. Register Load Instructions - Response Word Format.

The response word format used by Protocol Extension 2 complies with the 4-255 Data Bus 21-bit response word format shown in Figure 17.

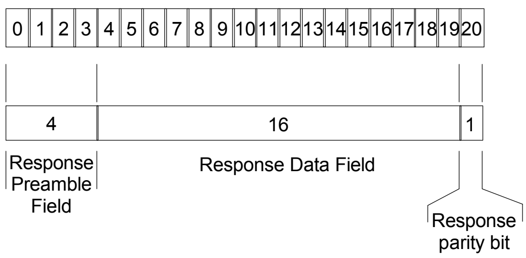

Figure 17 - Register Load Instruction Response Word Format.

256-PE2-9-D Register Load Instruction - Response Preamble Field

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. These four bits shall be used to indicate data quality, to report errors, and to request Bus Controller attention.

The format of the 4-255 Data Bus response preamble field for Protocol Extension 2 is shown in Figure 18.

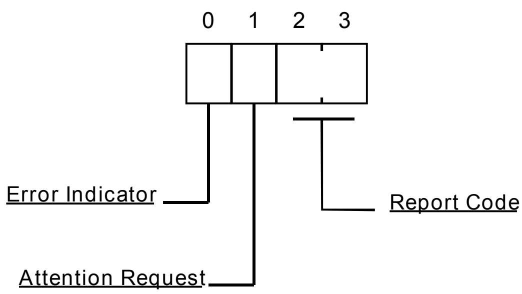

Figure 18 - Register Load Instruction - Response Preamble Field.

256-PE2-10-D PE-2 Response Preamble Field - Error Indicator

The response preamble field error indicator bit (bit 0) shall be set to zero (0) if a register load instruction is not rejected and the requested operation is completed successfully. This bit shall be set to one (1) if the register load instruction is rejected or if an error is detected during the execution of the requested operation.

256-PE2-11-D PE-2 Response Preamble Field - Attention Request

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE2-12-D PE-2 Response Preamble Field - Report Code, No Error

If no error occurs, the response preamble field report code field (bits 2 and 3) is reserved and by default shall be set to zero (00).

256-PE2-13-O PE-2 Response Preamble Field - Report Code, Error

Optionally, if an error occurs, the Remote Terminal shall set the response preamble field report code to indicate the type of error as per Table 5. If an error occurs and the Remote Terminal does not indicate the type of error, the preamble field report code shall be set to zero (00) to indicate an unspecified error.

Response Preamble Field				
Error Indicator	Attention Request	Report Code		Meaning
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	No register at specified load address
1	X	1	1	Reserved

Table 5 - Register Load Instructions - Response Preamble Field Report Codes.

Note 1: An 'unspecified error' report code (00) may indicate that an error code is contained in the data field.

Note 2: The 'instruction rejected' report code (01) indicates that the instruction has been rejected and was not executed.

Note 3: The 'no register at specified load address' report code (10) indicates that there is not a register at the register load address specified in the instruction.

256-PE2-14-D PE-2 Response Word - Data Field, No Error

The response data field occupies bits 4 to 19, inclusive, of the response word. If a register load instruction is executed without errors, this 16-bit field shall be set to the acknowledgement code value of all zeroes .

256-PE2-15-O PE-2 Response Word - Optional Data Field Error Codes

Optionally if errors are detected during the execution of a register load instruction and the response preamble field error indicator has been set to 1, the 16-bit response data field shall be used to carry a user defined error code.

256-PE2-16-D PE-2 Response Word - Parity Bit

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

256-PE2-17-D Response Parity Bit Value - Response Parity Mode = Even

When the PE-2 response parity mode is 'Even', the parity bit shall be set so that the sum of the parity bit plus all 'one' bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is 'None', the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

8. Protocol Extension 3 - Basic Control Instruction Protocol.

Protocol Extension 3 defines a set of control instructions which are used for basic operation of Remote Terminals regardless of their type or function.

It is recommended that designers of Remote Terminals should critically examine their basic terminal control requirements in the light of this protocol extension and should, wherever possible, make use of the instructions provided here.

The basic control instructions are:

- Remote Terminal Reset, RT_RESET,
- Remote Terminal Halt, RT_HALT,
- Remote Terminal Run, RT_RUN,
- Remote Terminal Status, RT_STATUS,
- Remote Terminal Perform Self Test, RT_TEST,
- Remote Terminal Report Self Test Results, RT_REPORT,
- Remote Terminal Broadcast Reception Enable, RT_BCST_ENABLE,
- Remote Terminal Broadcast Reception Disable, RT_BCST_DISABLE,
- Remote Terminal Clear Attention Request, RT_ATTN_CLEAR.

8.1. Basic Control Instruction Protocol - General Requirements.

This section defines a set of general requirements which must be met by Protocol Extension 3 entities.

256-PE3-1-O <u>Basic Control Instruction Protocol – Optional Response Parity Mod</u>e

Optionally, the basic control instruction protocol response parity mode at a given Remote Terminal shall be ‘Even’.

> Note 1: A response parity mode of ‘Even’ means that the PE-3 entity is required to generate an even parity value covering the response data field of basic control instruction responses. An even parity value is used by the PE-3 entity receiving the response to verify the quality of the data.
>
> Note 2: By default, the PE-3 response parity mode at a Remote Terminal is ‘None’.

256-PE3-2-D Basic Control Instruction Protocol – Parameter Value

Where the action of a basic control instruction is qualified by the value contained in the basic instruction parameter field so that application specific actions are performed, a parameter value of zero shall be used to indicate that normal, non-application specific basic control instruction actions are to be performed. A value other than zero shall indicate that special, application specific basic control instruction actions are to be performed.

Note: As an example, an RT_RESET instruction with a parameter value of zero should cause a general reset of the Remote Terminal while an RT_RESET instruction with a non-zero parameter value might cause only certain parts of a Remote Terminal to be reset.

8.2. Basic Control Instruction Set.

This section includes the requirements specific to each basic control instruction.

8.2.1. Remote Terminal Reset, RT_RESET.

256-PE3-3-D RT_RESET - Use

The Bus Controller shall issue a Remote Terminal Reset, RT_RESET, instruction to cause the addressed Remote Terminal to perform a reset operation.

256-PE3-4-D RT_RESET - Rejection Criteria

There are no rejection criteria associated with the RT_RESET instruction. The RT_RESET instruction shall not be rejected.

256-PE3-5-D RT_RESET - Action on Receipt

A Remote Terminal that receives an RT_RESET instruction shall perform a reset operation. The precise activity which occurs during a reset operation at a Remote Terminal is specific to the terminal in question and no particular behaviour is prescribed in this standard except that the Remote Terminals attention request flag shall be deasserted. The Remote Terminal shall generate a 4-255 Data Bus response to an RT_RESET instruction.

Note 1: In most cases the reset operation will be equivalent to a full power cycle or cold start reset and the Remote Terminal will enter a stable waiting state following execution of the reset operation.

Note 2: A terminal reset may result in the protocol handlers for other protocol extensions being restarted. This will cause the local instance of those protocol extensions to be reset and may require action by the Bus Controller to restore them to a desired state.

256-PE3-6-D <u>RT_RESET - Response</u>

The response to an RT_RESET instruction shall be a basic status response and shall indicate whether errors were detected during execution of the instruction.

Depending on the time required by the Remote Terminal to perform a reset, the contents of the status flags in the response may not reflect the post reset status. Furthermore, it is not unlikely that more than one word will be required to report the status of some Remote Terminals. In such cases the response word can only contain a subset of the post reset status and the full status must be acquired by an alternative means.

8.2.2. Remote Terminal Halt, RT_HALT.

256-PE3-7-D <u>RT_HALT - Use</u>

The Bus Controller shall issue a Remote Terminal Halt, RT_HALT, instruction to cause the addressed Remote Terminal to suspend its activity.

> Note: If use is made of the parameter carried with an RT_HALT instruction, e.g. to identify specific processes which must be halted, a parameter value of zero should be used for the main RT_HALT instruction, i.e. the instruction which causes the terminal as a whole to suspend activity.

256-PE3-8-D <u>RT_HALT - Mandatory Rejection Criteria</u>

There are no mandatory rejection criteria associated with the RT_HALT instruction.

256-PE3-9-O <u>RT_HALT - User Defined Rejection Criteria</u>

Optionallythe user shall define additional rejection criteria for the RT_HALT instruction.

256-PE3-10-D <u>RT_HALT - Action on Receipt, Instruction not Rejected</u>

A Remote Terminal that receives an RT_HALT instruction and does not reject it shall enter a stable holding state. It shall not exit this state until an RT_RUN instruction is received. The Remote Terminal shall generate a 4-255 Data Bus response to an RT_HALT instruction.

> Note 1: The precise action on receipt of an RT_HALT, e.g. the processes which are actually halted, is terminal specific. However, halted terminals must maintain an operational protocol stack capable of servicing at least PE-3 instructions. Instructions

for other protocol extensions received while a Remote Terminal is halted may be undeliverable and cause an undeliverable instruction response to be generated.

Note 2: When a Remote Terminal is halted, the RUNNING bit should be set to zero and the HALTED bit should be set to one in all basic status responses.

256-PE3-11-D RT_HALT - Action on Receipt, Instruction Rejected

A Remote Terminal that receives an RT_HALT instruction and rejects it shall generate a 4-255 Data Bus response indicating that the instruction has been rejected.

256-PE3-12-D RT_HALT - Response

The response to an RT_HALT instruction shall be a basic status response and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_HALT instruction.

8.2.3. Remote Terminal Run, RT_RUN.

256-PE3-13-D RT_RUN - Use

The Bus Controller shall issue a Remote Terminal Run, RT_RUN, instruction to cause the addressed Remote Terminal to resume the activity of processes which have been halted by an RT_HALT instruction or to exit a stable waiting state following an RT_RESET instruction.

Note: If use is made of the parameter carried with an RT_RUN instruction, e.g. to identify specific processes which must be resumed, a parameter value of zero should be used for the main RT_RUN instruction, i.e. the instruction which causes all terminal activity to be resumed.

256-PE3-14-D RT_RUN - Mandatory Rejection Criteria

A Remote Terminal that receives an RT_RUN instruction but is not in a halted state or a stable waiting state, shall reject that instruction.

256-PE3-15-O RT_RUN - User Defined Rejection Criteria

Optionallythe user shall define additional rejection criteria for the RT_RUN instruction.

256-PE3-16-D RT_RUN - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives an RT_RUN instruction and does not reject it, shall exit the halted or stable waiting state and shall generate a 4-255 Data Bus response.

Note: While a Remote Terminal is running, i.e. not in a halted or stable waiting state, the RUNNING bit should be set to one and the HALTED bit should be set to zero in all basic status responses.

256-PE3-17-D RT_RUN - Action on Receipt, Instruction Rejected

A Remote Terminal that receives an RT_RUN instruction and rejects it shall generate a 4-255 Data Bus response indicating that the instruction has been rejected.

256-PE3-18-D RT_RUN - Remote Terminal Response

The response to an RT_RUN instruction shall be a basic status response and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_RUN instruction.

8.2.4. Remote Terminal Status, RT_STATUS.

256-PE3-19-D RT_STATUS - Use

The Bus Controller shall issue a Remote Terminal Status, RT_STATUS, instruction to cause the addressed Remote Terminal to update its status information.

Note: This instruction does not acquire the status data at the Bus Controller since this may require more than one interrogation.

256-PE3-20-D RT_STATUS - Mandatory Rejection Criteria

There are no mandatory rejection criteria associated with the RT_STATUS instruction.

256-PE3-21-O RT_STATUS - User Defined Rejection Criteria

Optionallythe user shall define additional rejection criteria for the RT_STATUS instruction.

256-PE3-22-D RT_STATUS - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives an RT_STATUS instruction and does not reject it shall update its status information and assert its status data ready flag to indicate that fresh status information is available. That status information may subsequently be acquired by the Bus Controller. The Remote Terminal shall generate a 4-255 Data Bus response to the RT_STATUS instruction.

Note: The basic status response includes a STATUS_DATA_READY bit which is used to convey the state of the Remote Terminals status data ready flag. However, depending on the time required for the Remote Terminal to update its status information, this flag may not be asserted before the response to the RT_STATUS instruction is generated.

256-PE3-23-D RT_STATUS - Action on Receipt, Instruction Rejected

A Remote Terminal that receives an RT_STATUS instruction and rejects it shall generate a 4-255 Data Bus response indicating that the instruction has been rejected.

256-PE3-24-D RT_STATUS - Response

The response to an RT_STATUS instruction shall be a basic status response and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_STATUS instruction.

8.2.5. Remote Terminal Perform Self Test, RT_TEST.

256-PE3-25-D RT_TEST - Use

The Bus Controller shall issue a Remote Terminal Test, RT_TEST, instruction to cause the addressed Remote Terminal to perform its self test routine.

256-PE3-26-D RT_TEST - Mandatory Rejection Criteria

There are no mandatory rejection criteria associated with the RT_TEST instruction.

256-PE3-27-O RT_TEST - User Defined Rejection Criteria

Optionally the user shall define terminal specific rejection criteria for the RT_TEST instruction.

256-PE3-28-D RT_TEST - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives an RT_TEST instruction and does not reject it shall initiate its self test routine and shall generate a 4-255 Data Bus response.

Note 1: The precise set of tests which should be performed during a Remote Terminal self test operation, and exactly when the tests are performed following the receipt of the RT_TEST instruction, is Remote Terminal specific.

Note 2: Once the tests have been performed, the presence of self test data can be indicated to the Bus Controller by setting SELF_TEST_READY bit in basic status responses to one.

256-PE3-29-D <u>RT_TEST - Action on Receipt, Instruction Rejected</u>

A Remote Terminal that receives an RT_TEST instruction and rejects it shall not initiate its self test routine but shall generate a 4-255 Data Bus response indicating that the instruction was rejected.

256-PE3-30-D <u>RT_TEST - Remote Terminal Response</u>

The response to an RT_TEST instruction shall be a basic status response and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_TEST instruction.

8.2.6. Remote Terminal Report Self Test Results, RT_REPORT.

256-PE3-31-D <u>RT_REPORT - Use</u>

The Bus Controller shall issue a Remote Terminal Report, RT_REPORT, instruction to acquire the self test result summary from the addressed Remote Terminal. The Bus Controller sets a parameter in the instruction to identify the report required.

> Note: The complete self test results may comprise more than 16-bits of data. If this is the case, the method of acquiring that data is application specific.

256-PE3-32-D <u>RT_REPORT - Mandatory Rejection Criteria</u>

There are no mandatory rejection criteria associated with the RT_REPORT instruction.

256-PE3-33-O <u>RT_REPORT - User Defined Rejection Criteria</u>

Optionally the user shall define terminal specific rejection criteria for the RT_REPORT instruction.

256-PE3-34-D <u>RT_REPORT - Action on Receipt, Instruction not Rejected</u>

A Remote Terminal that receives an RT_REPORT instruction and does not reject it shall provide a self test result summary word identified by the parameter field value in the instruction and shall generate a 4-255 Data Bus response. If the parameter field value is zero, the response shall be a basic status response. If the parameter field value is non-zero, the response shall be user defined.

256-PE3-35-D <u>RT_REPORT - Action on Receipt, Instruction Rejected</u>

A Remote Terminal that receives an RT_REPORT instruction and rejects it shall generate a 4-255 Data Bus response indicating that the instruction was rejected.

256-PE3-36-D RT_REPORT – Response

The response to an RT_REPORT instruction shall be a basic status response or a user defined status response, and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_REPORT instruction.

8.2.7. Remote Terminal Broadcast Reception Enable, RT_BCST_ENABLE.

256-PE3-37-D RT_BCST_ENABLE - Use

The Bus Controller shall issue a Remote Terminal Broadcast Reception Enable, RT_BCST_ENABLE, instruction to enable broadcast interrogation reception at the Remote Terminal address to which the instruction is sent.

> Note: Remote Terminals which can selectively enable or disable broadcast interrogation reception should disable broadcast interrogation reception on start-up or after a reset. Broadcast interrogation reception should subsequently be explicitly enabled at such Remote Terminals.

256-PE3-38-D RT_BCST_ENABLE - Rejection Criteria

A Remote Terminal that receives an RT_BCST_ENABLE instruction but cannot selectively enable and disable broadcast interrogation reception shall reject that RT_BCST_ENABLE instruction.

256-PE3-39-D RT_BCST_ENABLE - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives an RT_BCST_ENABLE instruction and does not reject it shall accept subsequently received broadcast interrogations and shall generate a 4-255 Data Bus response.

> Note: When broadcast interrogation reception is enabled at a Remote Terminal, the BROADCASTS_ENABLED bit in basic status responses should be set to one.

256-PE3-40-D RT_BCST_ENABLE - Action on Receipt, Instruction Rejected

A Remote Terminal that receives an RT_BCST_ENABLE instruction and rejects it shall generate a 4-255 Data Bus response.

256-PE3-41-D RT_BCST_ENABLE - Response

The response to an RT_BCST_ENABLE instruction shall be a basic status response containing the status after the execution of the RT_BCST_ENABLE instruction, and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_BCST_ENABLE instruction.

8.2.8. Remote Terminal Broadcast Reception Disable, RT_BCST_DISABLE.

256-PE3-42-D RT_BCST_DISABLE - Use

The Bus Controller shall issue a Remote Terminal Broadcast Reception Disable, RT_BCST_DISABLE, instruction to disable broadcast interrogation reception at the Remote Terminal address to which the instruction is sent.

256-PE3-43-D RT_BCST_DISABLE - Rejection Criteria

A Remote Terminal that receives an RT_BCST_DISABLE instruction but cannot selectively enable and disable broadcast interrogation reception shall reject that RT_BCST_DISABLE instruction.

256-PE3-44-D RT_BCST_DISABLE - Action on Receipt, Instruction not Rejected

A Remote Terminal that receives an RT_BCST_DISABLE instruction and does not reject it shall generate a 4-255 Data Bus response and shall ignore subsequently received broadcast interrogations until broadcast reception is re-enabled (i.e. by an RT_BCST_ENABLE).

> Note: When broadcast interrogation reception is disabled at a Remote Terminal, the BROADCASTS_ENABLED bit in basic status responses should be set to zero.

256-PE3-45-D RT_BCST_DISABLE - Action on Receipt, Instruction Rejected

A Remote Terminal that receives an RT_BCST_DISABLE instruction and rejects it shall generate a 4-255 Data Bus response.

256-PE3-46-D RT_BCST_DISABLE - Response

The response to an RT_BCST_DISABLE instruction shall be a basic status response containing the status after the execution of the RT_BCST_DISABLE instruction, and shall indicate whether the instruction was rejected and whether errors were detected during execution of the RT_BCST_DISABLE instruction.

8.2.9. Remote Terminal Clear Attention Request, RT_ATTN_CLEAR.

256-PE3-47-D RT_ATTN_CLEAR - Use

The Bus Controller shall issue a Remote Terminal Attention Clear Request, RT_ATTN_CLEAR, instruction to cause the addressed Remote Terminal to clear its attention request indicator.

256-PE3-48-D RT_ATTN_CLEAR - Rejection Criteria

There are no rejection criteria associated with the RT_ATTN_CLEAR instruction. The RT_ATTN_CLEAR instruction shall not be rejected.

256-PE3-49-D RT_ATTN_CLEAR - Action on Receipt

A Remote Terminal that receives an RT_ATTN_CLEAR instruction shall clear its attention request indicator and shall generate a 4-255 Data Bus response.

256-PE3-50-D RT_ATTN_CLEAR - Response

The response to an RT_ATTN_CLEAR instruction shall be a basic status response containing the status after the execution of the RT_ATTN_CLEAR instruction, and shall indicate whether errors were detected during execution of the RT_ATTN_CLEAR instruction.

8.3. Basic Control Instruction Encoding.

8.3.1. Basic Control Instructions - Interrogation Word Format.

In Protocol Extension 3, the terminal data field of the 4-255 Data Bus interrogation word is divided into the following four fields:

- A 3-bit mode identifier field,
- A 4-bit instruction group identifier field,
- An 8-bit instruction code field,
- A 4-bit parameter field.

This format, in relation to the complete 4-255 Data Bus interrogation word, is shown in Figure 19.

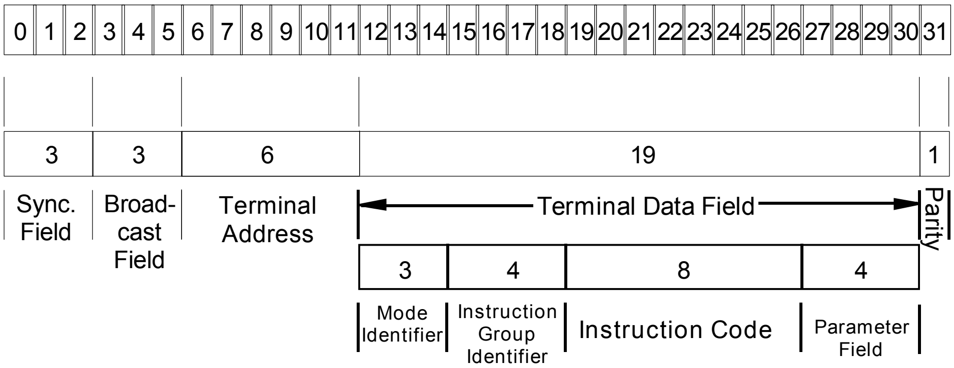

Figure 19 - Basic Control Interrogation Word Format.

256-PE3-51-D Basic Control Instructions - Mode Identifier Field

The mode identifier field occupies bits 12, 13, and 14 of the interrogation word and shall be set to 000 binary.

256-PE3-52-D Basic Control Instructions - Instruction Group Identifier Field

The instruction group identifier field occupies interrogation word bits 15 to 18 inclusive. This field shall be set to 0001 binary.

256-PE3-53-D Basic Control Instructions - Instruction Code Field

The 8-bit instruction code field occupies interrogation word bits 19 to 26 inclusive. This field shall be used to identify the operation to be performed on receipt of a basic control instruction.

256-PE3-54-D Basic Control Instructions - Instruction Code Assignments

The basic control instructions shall be encoded as specified in Table 6.

Note: Where this protocol extension does not provide all of the necessary Remote Terminal control instructions for a particular application, designers should consider using the unassigned codes within this basic control instruction group to ensure compatibility with other protocol extensions.

Mode Identifier = 000 Instruction Group Identifier = 1111									
Interrogation word bit number									Code assignment
19	20	21	22	23	24	25	26	27 28 29 30	
Instruction Code								Parameter Field	
0	0	0	0	0	0	0	0	x	Reserved
0	0	0	0	0	0	0	1	x	RT_HALT
0	0	0	0	0	0	1	0	x	RT_RUN
0	0	0	0	0	0	1	1	x	RT_STATUS
0	0	0	0	0	1	0	0	x	RT_TEST
0	0	0	0	0	1	0	1	x	RT_RESET
0	0	0	0	0	1	1	0	0	RT_REPORT - basic status response
0	0	0	0	0	1	1	0	≠0	RT_REPORT - user defined response
0	0	0	0	0	1	1	1	x	RT_BCST_ENABLE
0	0	0	0	1	0	0	0	x	RT_BCST_DISABLE
0	0	0	0	1	0	0	1	x	RT_ATTN_CLEAR
0	0	0	0	1	1	x	x	x	Reserved
0	0	0	1	x	x	x	x	x	Reserved
Any other code								x	Unassigned

Table 6 - Basic Control Instruction Encoding.

256-PE3-55-D <u>Basic Control Instructions - Parameter Field</u>

The parameter field occupies bits 27 to 30 inclusive and shall be used to convey a parameter with the basic control instruction. If a parameter value is not defined for a given basic control instruction, this field shall be set to zero (0000).

Note: The parameter value can be used to qualify the actions which are performed by a Remote Terminal on receipt of a given basic control instruction.

8.3.2. Basic Control Instructions - Response Word Format.

The response word format used by Protocol Extension 3 complies with the 4-255 Data Bus 21-bit response word format shown in Figure 20.

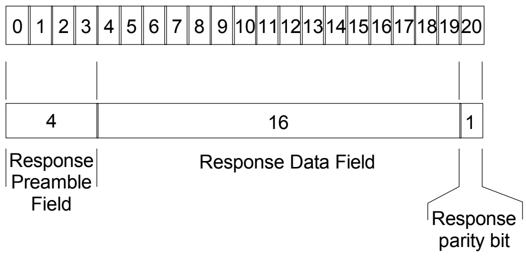

Figure 20 - Basic Control Instruction Response Word Format.

256-PE3-56-D <u>Basic Control Instruction - Response Preamble Field</u>

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. These four bits shall be used to indicate data quality, to report errors, and to request Bus Controller attention.

The response preamble field for Protocol Extension 3 is shown in Figure 21.

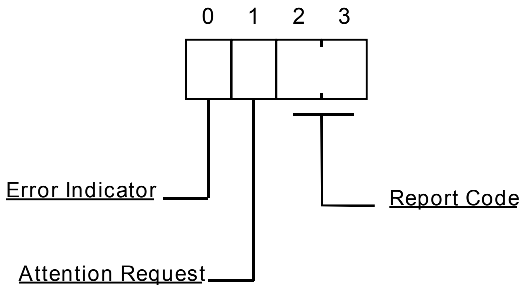

Figure 21 - Basic Control Instruction - Response Preamble Field.

256-PE3-57-D <u>PE-3 Response Preamble Field - Error Indicator</u>

The response preamble field error indicator bit (bit 0) shall be set to zero (0) if a basic control instruction is not rejected and the requested operation is completed successfully. This bit shall be set to one (1) if the basic control instruction is rejected or if an error is detected during the execution of the instruction.

256-PE3-58-D <u>PE-3 Response Preamble Field - Attention Request</u>

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE3-59-D <u>PE-3 Response Preamble Field - Report Code</u>

If no error occurs, the response preamble field report code field (bits 2 and 3) is reserved and by default shall be set to zero (00). If an error occurs, the Remote Terminal shall set the response preamble field report code to indicate the type of error as per Table 5.

Response Preamble Field				
Error Indicator	Attention Request	Report Code		Meaning
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	Unrecognised Instruction
1	X	1	1	Reserved

Table 7 - Register Load Instructions - Response Preamble Field Report Codes.

Note 1: If an error occurs a report code value of zero can be used to indicate that an error code is contained in the data field.

Note 2: The ‘Unrecognised Instruction’ report code is used to indicate that the instruction is not implemented at the terminal.

256-PE3-60-D <u>PE-3 Response Word - Data Field, No Error</u>

The response data field occupies bits 4 to 19, inclusive, of the response word. If a basic control instruction is executed without errors, this 16-bit field shall be used to convey the 16-bits of data which are required in response to that basic control instruction.

Note: Response data field formats for Protocol Extension 3 are defined in section 0

256-PE3-61-O <u>PE-3 Response Word - Optional Data Field Error Codes</u>

Optionally if errors are detected during the execution of a basic control instruction and the response preamble field error indicator has been set to 1, the 16-bit response data field (bits 4 to 19) shall be used to carry a user defined error code.

256-PE3-62-D <u>PE-3 Response Word - Parity Bit</u>

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

256-PE3-63-D Response Parity Bit Value - Response Parity Mode = Even

When the response parity mode is 'Even', the parity bit shall be set so that the sum of the parity bit plus all 'one' bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is 'None', the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

8.3.2.1. Basic Status Response Format.

The basic status response format is used for all of the Basic Control Instructions except the RT_REPORT instruction where a user defined response may be used if the instruction parameter value is non-zero.

256-PE3-64-D Basic Status Response - Data Field Format

The basic status response data field format shall be as shown in Figure 22.

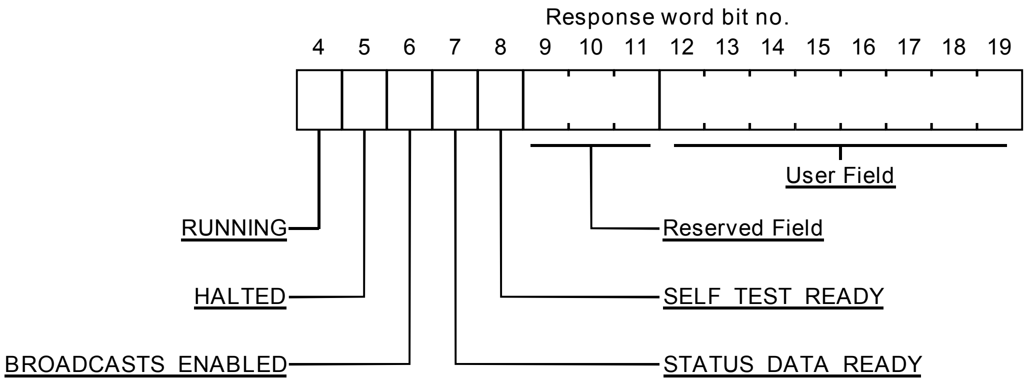

Figure 22 - Basic Status Response Data Field Format.

256-PE3-65-D Basic Status Response - RUNNING Bit

The basic status response RUNNING bit (bit 4) shall be set to one (1) to indicate that the processor or sequencer at the Remote Terminal is in running state when the basic status response is generated. Otherwise this bit shall be set to zero (0).

Note: The RUNNING bit refers to the status of the Remote Terminal as a whole, not to specific processes or applications.

256-PE3-66-D Basic Status Response - HALTED Bit

The basic status response HALTED bit (bit 5) shall be set to one (1) to indicate that the processor or sequencer at the Remote Terminal is in halted state when the basic status response is generated. Otherwise this bit shall be set to zero (0).

> Note: The HALTED bit refers to the status of the Remote Terminal as a whole, not to specific processes or applications.

256-PE3-67-D Basic Status Response - BROADCASTS_ENABLED Bit

The basic status response BROADCASTS_ENABLED bit (bit 6) shall be set to one (1) to indicate that broadcast interrogation reception at the Remote Terminal is enabled when the basic status response is generated. Otherwise this bit shall be set to zero (0).

256-PE3-68-D Basic Status Response - STATUS_DATA_READY Bit

The basic status response STATUS_DATA_READY bit (bit 7) shall be set to one (1) to indicate that fresh status data is available at the Remote Terminal. Otherwise this bit shall be set to zero (0).

> Note: The event or conditions which cause the STATUS_DATA_READY bit to be de-asserted are terminal specific.

256-PE3-69-D Basic Status Response - SELF_TEST_READY Bit

The basic status response SELF_TEST_READY bit (bit 8) shall be set to one (1) to indicate that self test results are available at the Remote Terminal when the basic status response is generated. Otherwise this bit shall be set to zero (0).

> Note: The event or conditions which cause the SELF_TEST_READY bit to be de-asserted are terminal specific.

256-PE3-70-D Basic Status Response - Reserved Field

Bits 9 to 11 inclusive of the basic status response are reserved and shall be set to zero (0).

256-PE3-71-D Basic Status Response - User Field

The basic status response User Field (bits 12 to 19 inclusive) shall be used to provide terminal specific status information. The format of the data within this field shall be user defined. By default, unused bits of the user field shall be set to zero (0).

> Note: As well as being terminal specific, the user field format may be instruction specific. I.e. at a given terminal, the information contained in the user field may be different for each of the PE-3 instructions.

9. Protocol Extension 4 - The Remote Bus Interface Protocol.

9.1. Introduction to the RBI Protocol.

Protocol Extension 4, is designed primarily to provide a block transfer capability over a 4-255 Data Bus and is referred to as the Remote Bus Interface, RBI, protocol.

The RBI protocol specification covers the names and definitions of RBI instructions, the encoding rules for RBI instructions over the 4-255 Data Bus, and definitions of procedures for transferring data using RBI instructions.

9.2. RBI Terminology.

The following terms are used in the specification of the RBI protocol:

Address Offset: An address which refers to a single memory location within a memory page.

Page Selector: Part of an address which identifies a memory page. Each page may consist of many memory locations.

RBI Output Data: Data transferred from a Remote Terminals memory onto the 4-255 Data Bus as a result of RBI instructions. RBI output may go to the 4-255 response bus or to the 4-255 block transfer bus.

RBI Entity: A collection of functions that together implement the RBI protocol in a terminal.

RBI Response: A standard 4-255 Data Bus response produced as a result of an RBI instruction.

Physical Address: A memory address produced by combining a page selector and an offset which refers to a physical location in the local memory.

Start Address: The address offset of the memory location containing the first word of a block of data.

Start Address Pointer: A pointer to a memory address offset which is initially loaded with the start address of a block of data but which is maintained by an RBI entity so that it points to the next word of a block of data.

Terminal: Terminal refers to a 4-255 Data Bus terminal. Unless explicitly stated otherwise, a terminal may be a Bus Controller or a Remote Terminal.

9.3. Principles of the RBI Protocol.

9.3.1. RBI Memory Access and Addresses.

The RBI protocol relies on the ability to access memory in a 4-255 Data Bus Remote Terminal independently of any processors or controllers operating in that terminal. This gives an effective Direct Memory Access, DMA, to the terminals memory under the control of the Bus Controller.

In order to do this, the RBI protocol must be able to generate memory addresses covering the physical memory space available in the Remote Terminal. For complex Remote Terminals this may be an extensive range of addresses requiring more than 16-bits. RBI protocol addresses are therefore made up of two components, a page selector and an address offset. These are combined to produce a physical address which identifies a location in the terminal memory. The precise way in which these components are combined (and even whether the page selector is used at all) is implementation dependent.

RBI memory accesses are analogous to conventional DMA. The RBI entity maintains a block length indicator and manages the address offset locally (using a start address pointer) so that blocks of memory can be read and written automatically. An RBI entity is therefore required to provide an address offset pointer, a page selector, and a block length indicator.

The RBI protocol provides instructions to initialise and read all of the required pointers and indicators. However, during a block transfer the local RBI entity must maintain these values so that they reflect the current status of the transfer, i.e. the start address pointer must be incremented and the length indicator decremented after each word of a block is transferred.

During a block transfer, the offset pointer can be incremented past its maximum value, i.e. rollover. This could result in data being inadvertently written to the bottom of a memory page. Therefore, the RBI protocol stipulates that instructions to access the Remote Terminal memory following the offset pointer rollover are prevented and an error is flagged. The page selector cannot be automatically incremented, it can only be explicitly loaded and read by the Bus Controller.

9.3.2. RBI Block Transfers.

The RBI protocol supports the transfer of blocks of data between terminals on the 4-255 Data Bus. Block transfers are handled in up to three stages:

- An initialisation stage where the terminals involved in the data transfer are set up,
- A transfer stage where the data is actually moved between terminals,
- An optional confirmation stage during which post transfer checks are performed.

The initialisation stage includes:

- Selecting the bus to be used for the transfer (interrogation, response, or block transfer),
- Setting up the direction of the transfer,
- Setting the page selector and start address for the data block,
- Setting the length of the data block.

Initialisation must be performed, using the 4-255 Data Bus interrogation and response buses, for each terminal participating in the transfer.

During the transfer stage, the number of RBI instructions depends on the bus being used to carry the data. If the 4-255 response bus is being used, each word of data must be strobed across the bus by an RBI instruction. If the 4-255 Block Transfer Bus, BTB, is being used, only a single RBI instruction is required to start the transfer which then continues autonomously.

The optional confirmation stage determines whether the correct number of words have been transferred and whether any low level errors (e.g. Litton errors, length errors, etc.) were detected. However, the RBI confirmation stage cannot verify the integrity of the data within the block. Data integrity can only be verified by higher layer communicating entities, i.e. those which are using the RBI protocol to implement a communication service. One obvious method of performing such checks is to embed checksum information in the block but many other methods could be used. If higher level data integrity checks are performed, the RBI confirmation stage may be unnecessary.

The following sections describe different transfer methods which are supported by the RBI protocol.

9.3.2.1. Transfers Between BC and RT via the Interrogation and Response Buses.

The most fundamental method is to transfer data word-by-word between the Bus Controller, BC, and a Remote Terminal, RT, using the 4-255 interrogation and response buses. To perform this type of transfer, the BC first configures the RBI in the RT to receive or transmit the block (i.e. sets the start address and page selector, length, direction, etc.) and then sends the appropriate number of proceed instructions (one for each word of the block). If the transfer is from the BC to the RT, each proceed instruction carries one data word. If the transfer is from the RT to the BC, the RBI must place each word of the block in the response to the proceed instruction. This method is only suitable for transferring data between the BC and one RT at a time.

This method can also be used to transfer data between RTs, using the BC to store and forward the data. The Bus Controller must configure one RT to send the block, and the other RT to receive it. The BC transfers the block to itself one word at a time by sending proceed instructions to the data sending RT, and forwards each word to each data receiving RT by sending them proceed instructions containing the data.

9.3.2.2. Direct RT to RT Transfers via the Response Bus.

Another method of transfer over the 4-255 Data Bus allows direct RT to RT transfers via the response bus using the programmed listening mode. The communicating RTs are each configured with the length of the block to be transferred and with the appropriate start address and page selector. The data sending RT is then configured to place RBI output data on the response bus, i.e. as a response bus talker. The other data receiving RTs (there may be several) are programmed as response bus listeners and are provided with a Programmed Listening Address, PLA, which matches the terminal address of the data sending RT. The appropriate number of proceed instructions are then sent to the data sending RT which provides one data word of the block in response to each proceed instruction. The data receiving RTs decode each proceed instruction sent to the data sending RT (using the PLA) and acquire the corresponding response word which contains the data.

This transfer method is known as a programmed listening transfer and, for RT to RT transfers, is more efficient than performing an RT to BC transfer followed by one or more BC to RT transfers since the data only appears on the bus once.

> Note: If an Enhanced 4-255 Data Bus Modem is used to implement the Physical and Access Control Sub-layer services, response bus listening RTs must set the Interrogation Word Receive Service address decoding level to 1 or 0 to ensure that they acquire the proceed instructions sent to the data sending RT.

9.3.2.3. Transfers via the Block Transfer Bus.

The 4-255 BTB provides a very efficient means of transferring data between terminals. As for the other transfer methods, each participating terminal (RT or BC) must be configured with the start address, page selector, and the length of the block to be transferred. The data sending terminal must be configured to place RBI output data on the 4-255 BTB, i.e. as a 4-255 BTB talker and all data receiving terminals must be configured as block transfer bus listeners.

To start the transfer, each data receiving terminal (there may be many) is sent a proceed instruction which tells it to acquire the next block transmitted on the 4-255 BTB. The data sending terminal is then sent a proceed instruction which causes it to transfer the entire block serially onto the 4-255 BTB with no further involvement of the Bus Controller.

4-255 BTB transfers are highly efficient since the only traffic on the interrogation and response buses is the RBI set-up instructions and the individual proceed instructions needed to start the transfer. All of the data traffic is carried on the 4-255 BTB leaving the 4-255 interrogation and response buses free for other operations. Furthermore, for the same signalling rate, the data transfer rate over the 4-255 BTB is twice that over the 4-255 interrogation and response buses since two 16-bit words are transferred per bus slot.

9.3.2.4. Relative Efficiency of Different Transfer Methods.

Table 8 shows the number of 4-255 Data Bus interrogations which are required to transfer a single block of n-words using the different transfer methods described above. This indicates the relative efficiencies of the methods in terms of 4-255 Data Bus bandwidth utilisation. Another consideration is the level of involvement of the Bus Controller. For the methods which use the 4-255 interrogation and response buses but do not use programmed listening, the BC must handle the data being transferred during the transfer stage. During a programmed listening transfer, the BC has to generate the appropriate number of proceed instructions but does not need to handle the data and could be free to perform other tasks. In the case of a 4-255 BTB transfer the transfer stage is essentially autonomous and the BC would therefore be free to perform other tasks.

	No. of 4-255 Data Bus interrogations required for an n-word block data transfer					
	(1 sender/1 receiver)			(1 sender/m receivers)		
	Initialisation Stage	Transfer Stage	Confirmation Stage (Optional)	Initialisation Stage	Transfer Stage	Confirmation Stage (Optional)
RT to BC transfer via I and R buses	4	n	1	Not Possible		
BC to RT transfer via I bus	4	n	1	4 x m	n x m	m
RT/RT transfer using BC to store and forward data	8	2 x n	2	4+(4 x m)	n+(n x m)	m+1
RT/RT using programmed listening mode	8	n	2	4+(4 x m)	n	m+1
Terminal/terminal transfer via BTB	8	2	2	4+(4 x m)	m+1	m+1

Table 8 - Relative Efficiency of Different RBI Transfer Methods.

9.3.3. RBI Block Transfer State Transition Model.

An RBI block transfer can be considered in terms of a state transition model with the state transition diagram shown in Figure 23.

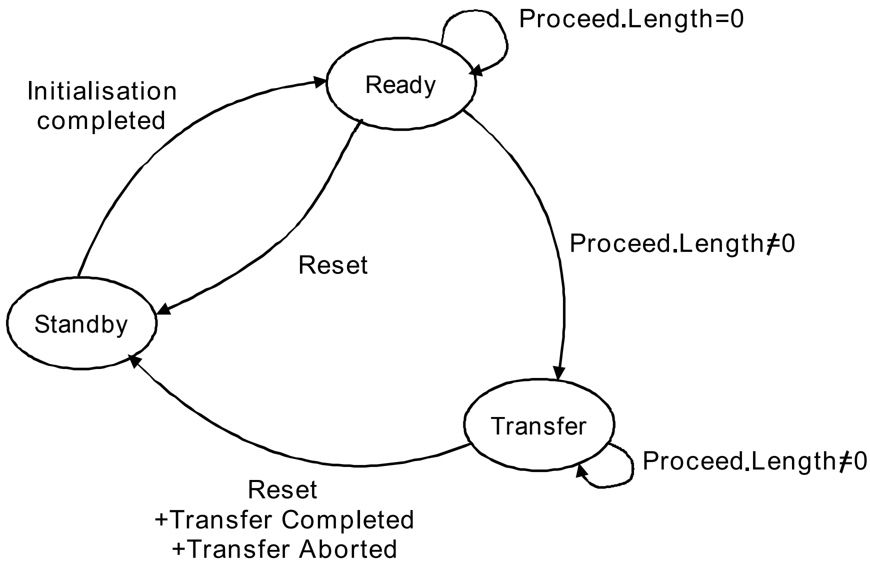

Figure 23 – RBI Block Transfer State Diagram.

The Standby state is the default state which an RBI entity adopts on power-up, following a reset, or following the completion of a block transfer. In this state an RBI entity will accept configuration instructions but will not accept instructions to perform data transfers to the Remote Terminal memory.

The Ready state is entered once RBI entities involved in a block transfer have been configured, i.e. the block length, start address, and bus to be used for the transfer have been set using the initialisation sequence. While in ready state RBI entities may be re-configured e.g by changing the start address or length values. If an RBI entity in ready state receives a proceed instruction when its length indicator is not zero, it enters the transfer state.

Transfer state is the state during which data is actually transferred to or from the Remote Terminals memory. Data can be transferred between terminals whose RBI entities are in transfer state. Once a block transfer has been completed normally, i.e. the length indicator has decremented to zero, or has been aborted, the RBI entity enters the standby state.

9.4. RBI Protocol - General Requirements.

This section defines a set of general requirements which must be met by RBI entities.

256-PE4-1-D RBI Protocol - Start Address Pointer

An RBI entity shall maintain a start address pointer which shall be used to provide the address offset for block transfer memory accesses.

256-PE4-2-D RBI Protocol - Block Transfer Page Selector

An RBI entity shall maintain a block transfer page selector which shall be used in combination with an address offset to generate a physical address.

256-PE4-3-D RBI Protocol - Block Length Indicator

An RBI entity shall maintain a block length indicator which shall be used to indicate the number of 16-bit words which remain to be transferred during a block transfer.

256-PE4-4-D RBI Protocol - Programmed Listening Address

An RBI entity shall maintain a programmed listening address which shall be used to identify a data sending terminal for response bus block transfers.

256-PE4-5-D RBI Protocol - Physical Address Generation

An RBI entity shall generate a physical address by combining the page selector and address offset.

256-PE4-6-O RBI Protocol – Optional Response Parity Mode

Optionally, the RBI response parity mode at a given Remote Terminal shall be 'Even'.

Note 1: A response parity mode of 'Even' means that the RBI entity is required to generate an even parity value covering the response data field of RBI responses. The even parity value is used by receiving RBI entities (i.e. in the Bus Controller and Remote Terminals operating as response bus listeners) to verify the quality of the data contained in the response.

Note 2: By default, the RBI response parity mode at a Remote Terminal is 'None'.

9.5. RBI Protocol Instruction Set

The RBI protocol instruction set includes the instructions which are needed to perform block transfer operations over the 4-255 Data Bus. These instructions support the initialisation stage, the transfer stage, and the optional confirmation stage of a block transfer.

256-PE4-7-D RBI Protocol - Instruction Set

The RBI protocol shall comprise the following instructions:

- Load Start Address, LOSA,
- Load Block Transfer Page Selector, LOPS_B,
- Load Block Length Indicator, LOLE,
- Read Start Address, RESA,
- Read Block Transfer Page Selector, REPS_B,
- Read Block Length Indicator, RELE,
- Read RBI Status, RBISTAT,
- Read User Status, USRSTAT,
- Set RBI Mode, RBIMODE,
- Reset RBI, RBIRES,
- Output User Control, RBIUSR,
- Proceed, PROC.
- Block Transfer Bus Abort, BTB_ABORT

Note: In a particular implementation, this instruction set may be extended to support device specific features. The encoding scheme for the RBI instructions allows considerable scope for the definition of additional instructions.

9.5.1. Load Start Address, LOSA.

256-PE4-8-D RBI LOSA Instruction - Use

The Bus Controller shall issue a Load Start Address, LOSA, instruction in order to set a 16-bit address offset value in the start address pointer of an RBI entity at the addressed Remote Terminal. The start address is the address offset of the first data word to be transferred during the next block transfer to or from that Remote Terminal.

256-PE4-9-D RBI LOSA Instruction - Rejection Criteria

There are no rejection criteria associated with the LOSA instruction. The LOSA instruction shall not be rejected.

256-PE4-10-D RBI LOSA Instruction - Action on Receipt

An RBI entity that receives a LOSA instruction shall load the 16-bit address offset contained in that instruction into its start address pointer and shall generate a 4-255 Data Bus response.

256-PE4-11-D RBI LOSA Instruction - Address Value

Any 16-bit address offset value shall be permitted in the LOSA instruction data field.

Note: Care should be taken that the address offset value provided, when combined with the page selector, corresponds to a physical memory location on the Remote Terminal. If a value which corresponds to a non-existent memory location is provided, an error may be flagged.

256-PE4-12-D RBI LOSA Instruction - Response

The response to a LOSA instruction shall contain the value of the start address pointer after executing the LOSA instruction, and shall indicate whether errors were detected during the execution of the LOSA instruction.

9.5.2. Load Block Transfer Page Selector, LOPS_B.

256-PE4-13-D RBI LOPS_B Instruction - Use

The Bus Controller shall issue a Load Block Transfer Page Selector, LOPS_B, instruction to set the block transfer page selector value in the RBI entity at the addressed Remote Terminal. The page selector value specifies the page containing the memory storage location to be used during the next block transfer to or from memory in that Remote Terminal.

256-PE4-14-D RBI LOPS_B Instruction - Rejection Criteria

There are no rejection criteria associated with the LOPS_B instruction. The LOPS_B instruction shall not be rejected.

256-PE4-15-D RBI LOPS_B Instruction - Action on Receipt

An RBI entity that receives a LOPS_B instruction shall use the data contained in the instruction as a memory page selector for block data transfers and shall generate a 4-255 Data Bus response.

256-PE4-16-D RBI LOPS_B Instruction - Page Selector Value

Any page selector value up to 15-bits in length shall be permitted in the LOPS_B instruction data field.

> Note: Care should be taken that the page selector value provided in a LOPS_B instruction corresponds to an actual physical memory area in the Remote Terminal. An error may be flagged if it is detected that the page selector value references a non-existent memory area.

256-PE4-17-D RBI LOPS_B Instruction - Response

The response to a LOPS_B instruction shall contain the block transfer page selector value after executing the LOPS_B instruction, and shall indicate whether any errors were detected during the execution of the LOPS_B instruction.

9.5.3. Load Block Length Indicator, LOLE.

256-PE4-18-D RBI LOLE Instruction - Use

The Bus Controller shall issue a Load Block Length Indicator, LOLE, instruction to set the length indicator maintained by the RBI entity at the addressed Remote Terminal. The length value is the number of 16-bit data words, to be transferred during the next block transfer to or from that Remote Terminal.

256-PE4-19-D RBI LOLE Instruction - Rejection Criteria

There are no rejection criteria associated with the LOLE instruction. The LOLE instruction shall not be rejected.

256-PE4-20-D RBI LOLE Instruction - Action on Receipt

An RBI entity that receives a LOLE instruction shall load the length value specified in that instruction into its block length indicator and shall generate a 4-255 Data Bus response.

256-PE4-21-D RBI LOLE Instruction - Length Value

The length indicator value contained in the LOLE instruction is a 14-bit binary encoded unsigned integer. This value is interpreted as the number of 16-bit data words to be transferred.

256-PE4-22-D RBI LOLE Instruction - Response

The response to a LOLE instruction shall contain the value of the block length indicator after executing the LOLE instruction, and shall indicate whether errors were detected during the execution of the LOLE instruction.

9.5.4. Read Start Address, RESA.

256-PE4-23-D RBI RESA Instruction - Use

The Bus Controller shall issue a Read Start Address, RESA, instruction to read the current value of the start address pointer maintained by the RBI entity at the addressed Remote Terminal.

256-PE4-24-D RBI RESA Instruction - Rejection Criteria

There are no rejection criteria associated with the RESA instruction. The RESA instruction shall not be rejected.

256-PE4-25-D RBI RESA Instruction - Action on Receipt

An RBI entity that receives a RESA instruction shall generate a 4-255 Data Bus response containing the current value of its start address pointer. The value of the start address pointer shall not be changed by this operation.

Note: If the RBI entity has been initialised but no data transfers have yet taken place, the value reported after a RESA instruction will match that which was set using the LOSA instruction.

256-PE4-26-D RBI RESA Instruction - Response

The response to a RESA instruction shall contain the current value of the start address pointer maintained by the RBI entity, and shall indicate whether errors were detected during the execution of the RESA instruction.

9.5.5. Read Block Transfer Page Selector, REPS_B.

256-PE4-27-D RBI REPS_B Instruction - Use

The Bus Controller shall issue a Read Block Transfer Page Selector, REPS_B, instruction in order to read the page selector maintained by the RBI entity at the addressed Remote Terminal.

256-PE4-28-D RBI REPS_B Instruction - Rejection Criteria

There are no rejection criteria associated with the REPS_B instruction. The REPS_B instruction shall not be rejected.

256-PE4-29-D RBI REPS_B Instruction - Action on Receipt

An RBI entity that receives a REPS_B instruction shall generate a 4-255 Data Bus response containing the current value of its block transfer page selector. The value of the block transfer page selector shall not be changed by this operation.

256-PE4-30-D RBI REPS_B Instruction - Response

The response to a REPS_B instruction shall contain the current value of the page selector maintained by the RBI entity, and shall indicate whether errors were detected during the execution of the REPS_B instruction.

9.5.6. Read Block Length Indicator, RELE.

256-PE4-31-D RBI RELE Instruction - Use

The Bus Controller shall issue a Read Block Length Indicator, RELE, instruction in order to read the block length indicator maintained by the RBI entity at the addressed Remote Terminal. The block length indicator value is the number of words which remain to be transferred during the current block transfer to or from that Remote Terminal.

256-PE4-32-D RBI RELE Instruction - Rejection Criteria

There are no rejection criteria associated with the RELE instruction. The RELE instruction shall not be rejected.

256-PE4-33-D RBI RELE Instruction - Action on Receipt

An RBI entity that receives a RELE instruction shall generate a 4-255 Data Bus response containing the current value of its block length indicator. The value of the block length indicator shall not be changed by this operation.

> Note: If the RBI entity has been initialised but no data transfers have yet taken place, the block length indicator value reported after a RELE instruction will match that which was set using the LOLE instruction.

256-PE4-34-D RBI RELE Instruction - Response

The response to a RELE instruction shall contain the current value of the block length indicator maintained by the RBI entity, and shall indicate whether errors were detected during the execution of the RELE instruction.

9.5.7. Read RBI Status, RBISTAT.

256-PE4-35-D RBI RBISTAT Instruction - Use

The Bus Controller shall issue a Read RBI Status, RBISTAT, instruction to read the status of the RBI entity at the addressed Remote Terminal. This instruction shall contain a reset flag bit indicating whether the RBI status flags should be reset after being read.

> Note: The status information acquired using the RBISTAT instruction is relevant to the operation of the RBI entity, and is independent of the Remote Terminal status. However, some implementations may use spare bits in this response to convey other status information which is not specific to the RBI entity.

256-PE4-36-D RBI RBISTAT Instruction - Rejection Criteria

There are no rejection criteria associated with the RBISTAT instruction. The RBISTAT instruction shall not be rejected.

256-PE4-37-D RBI RBISTAT Instruction - Action on Receipt

An RBI entity that receives an RBISTAT instruction shall generate a 4-255 Data Bus response. If the reset flag bit in the instruction data field is zero, the Data Transfer Overrun and Latched DMA Error status indicators (DTO_FLAG and LDMA_ERR respectively) shall be reset after the status read. If the reset flag bit in the instruction data field is one, all of the RBI status indicators, including the DTO_FLAG and LDMA_ERR status indicators, shall remain unchanged after the status read.

256-PE4-38-D RBI RBISTAT Instruction - Response

The response to an RBISTAT instruction shall be an RBI status response and shall indicate whether errors were detected during the execution of the RBISTAT instruction.

9.5.8. Read User Status, USRSTAT.

256-PE4-39-D RBI USRSTAT Instruction - Use

The Bus Controller shall issue a Read User Status, USRSTAT, instruction to acquire user status data from the RBI entity at the addressed Remote Terminal. This status data is user defined and may reflect the current status of the Remote Terminal as a whole.

256-PE4-40-D RBI USRSTAT Instruction - Rejection Criteria

There are no rejection criteria associated with the USRSTAT instruction. The USRSTAT instruction shall not be rejected.

256-PE4-41-D RBI USRSTAT Instruction - Action on Receipt

An RBI entity that receives a USRSTAT instruction shall acquire user status data and shall generate a 4-255 Data Bus response.

256-PE4-42-D RBI USRSTAT Instruction - Response

The response to a USRSTAT instruction shall contain 16-bits of user status data and shall indicate whether errors were detected during the execution of the USRSTAT instruction.

9.5.9. Set RBI Mode, RBIMODE.

256-PE4-43-D RBI RBIMODE Instruction - Use

The Bus Controller shall issue a Set RBI Mode, RBIMODE, instruction to configure the operational mode of the RBI entity at the addressed Remote Terminal. The RBIMODE instruction shall specify the following:

- Whether the RBI entity will send or receive data,
- The bus to be used for block data transfers,
- The programmed listening address.

256-PE4-44-D RBI RBIMODE Instruction - Rejection Criteria

An RBI entity shall reject an RBIMODE instruction if it does not support the requested mode.

Note: An RBIMODE instruction containing the reserved code 11 in the transfer bus selection field will be rejected as an unsupported mode.

256-PE4-45-D RBI RBIMODE Instruction - Action on Receipt, Instruction not Rejected

An RBI entity that receives an RBIMODE instruction and does not reject it shall adjust its configuration to match that specified by the instruction and shall generate a 4-255 Data Bus response.

256-PE4-46-D RBI RBIMODE Instruction - Action on Receipt, Instruction Rejected

An RBI entity that receives an RBIMODE instruction and rejects it shall not change its configuration but shall generate a 4-255 Data Bus response.

256-PE4-47-D RBI RBIMODE Instruction - Response

The response to an RBIMODE instruction shall be an RBI mode response indicating the mode after execution of the instruction, and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the RBIMODE instruction.

9.5.10. RBI Reset, RBIRES.

256-PE4-48-D RBI RBIRES Instruction - Use

The Bus Controller shall issue an RBI Reset, RBIRES, instruction to set the RBI entity at the addressed Remote Terminal to a known state.

Note: An RBIRES instruction should not cause a reset of the Remote Terminal. Only the RBI entity at that Remote Terminal should be affected.

256-PE4-49-D RBI RBIRES Instruction - Rejection Criteria

There are no rejection criteria associated with the RBIRES instruction. The RBIRES instruction shall not be rejected.

256-PE4-50-D RBI RBIRES Instruction - Action on Receipt

An RBI entity that receives an RBIRES instruction shall:

- Abandon any ongoing block transfers,
- Set the RBI status parameters to reflect the post reset condition of the RBI entity,
- Clear the following status indicators:
    - DTO_FLAG,
    - DMA_ERR,
    - LDMA_ERR,
    - SAR_FLAG,
    - BTB_BUSY,
    - BTB_ABORTED,
    - BTB_ERROR,
- Set the start address pointer value to zero,
- Set the block length indicator to zero,
- Set the page selector to zero,
- Set the RBI mode to interrogation bus receiver with programmed listening address zero,
- Generate a 4-255 Data Bus response.

256-PE4-51-D RBI RBIRES Instruction - Response

The response to an RBIRES instruction shall be an RBI status response indicating the status after execution of the RBIRES instruction, and shall indicate whether errors were detected during the execution of the RBIRES instruction.

9.5.11. Output User Control, RBIUSR.

256-PE4-52-D RBI RBIUSR Instruction - Use

The Bus Controller shall issue an Output User Control, RBIUSR, instruction to provide up to 14-bits of user defined data or control signals to the addressed Remote Terminal via an RBI entity. The signals provided in this way may be used to configure or control the addressed Remote Terminal.

Note: The actual number of user control signals which can be provided to the user is implementation specific.

256-PE4-53-D RBI RBIUSR Instruction - Rejection Criteria

There are no rejection criteria associated with the RBIUSR instruction. The RBIUSR instruction shall not be rejected.

256-PE4-54-D RBI RBIUSR Instruction - Action on Receipt

An RBI entity that receives an RBIUSR instruction shall make the data contained in the data field of that instruction available to users at the Remote Terminal and shall generate a 4-255 Data Bus response.

Note: This is typically achieved by writing the data to a register located in the Remote Terminal.

256-PE4-55-D RBI RBIUSR Instruction - Response

The response to an RBIUSR instruction shall be the same as that for a USRSTAT instruction.

9.5.12. Proceed, PROC.

256-PE4-56-D RBI PROC Instruction - Use

The Bus Controller shall issue a Proceed, PROC, instruction to cause RBI entities which have been configured to perform a block transfer to exchange data.

256-PE4-57-D RBI PROC Instruction - Rejection Criteria

An RBI entity that receives a PROC instruction shall reject that instruction if any of the following conditions apply:

- The block length indicator value is zero,
- No RBIMODE instruction has been received since the last RBIRES or power-on reset,
- The RBI entity is configured to transfer data on the Block Transfer Bus and a BTB_ABORT instruction has been received since the last RBIMODE instruction,
- A start address pointer rollover has occurred since the last power-on reset and a LOSA or RBIRES instruction has not been received following that rollover.

Note: Start address pointer rollover occurs when the pointer is incremented past all ones.

256-PE4-58-D <u>RBI PROC Instruction - Data Field</u>

If the PROC instruction is sent to an RBI entity configured to receive a block transfer on the interrogation bus, the instruction data field shall contain the 16-bit data word to be transferred to the addressed Remote Terminal. Otherwise, the instruction data field shall be set to all zeroes.

256-PE4-59-D <u>RBI PROC Instruction - Action on Receipt</u>

The precise action resulting from a PROC instruction shall be dependent on the selected block transfer mode and the current state of the participating RBI entities, and is defined in the following requirements.

256-PE4-60-D <u>RBI PROC - Interrogation Bus Listener</u>

An RBI entity that receives a PROC instruction and which:

A) has been configured as a 4-255 interrogation bus listener and,
B) does not reject the instruction,

shall perform the following sequence of actions in the order given:

1) Transfer the data contained in the instruction data field to the physical address currently referenced by the start address pointer combined with the block transfer page selector.

2) Increment the start address pointer to reference the next memory location.

3) Decrement the block length indicator value.

4) Generate a 4-255 Data Bus response which shall contain the new value of the block length indicator and shall indicate whether errors were detected during the execution of the PROC instruction.

256-PE4-61-D <u>RBI PROC - Response Bus Talker</u>

An RBI entity that receives a PROC instruction and which:

A) has been configured as a 4-255 response bus talker and,
B) does not reject the instruction,

shall perform the following sequence of actions in the order given:

1) Read the data from the physical address currently referenced by the start address pointer combined with the block transfer page selector for inclusion in the response to the interrogation.

2) Increment the start address pointer to reference the next memory location.

3) Decrement the block length indicator value.

4) Generate a 4-255 Data Bus response which shall contain the 16-bit data read from the memory and shall indicate whether errors were detected during the execution of the PROC instruction.

256-PE4-62-D RBI PROC - Response Bus Listener

An RBI entity which:

A) has been configured as a 4-255 response bus listener and,
B) has a block length indicator value not equal to zero,

and which identifies a PROC instruction addressed to the Remote Terminal whose address matches its currently configured programmed listening address shall perform the following sequence of actions in the order given:

1) Acquire the response to that PROC instruction from the 4-255 response bus.

2) Examine the response preamble field bits. If the response preamble field error indicator bit is set to one, the response shall be ignored. Otherwise the following steps shall be performed.

3) Transfer the data word contained in that response into the physical address which is being referenced by the start address pointer combined with the block transfer page selector.

4) Increment the start address pointer to reference the next memory location.

5) Decrement the block length indicator value.

The 4-255 response bus listener Remote Terminal shall not generate a response to this interrogation.

> Note: During step 2 of this sequence some responses may be ignored which could result in some data words of a block not being received even though they were transmitted. This condition can be detected either by the optional confirmation stage of the RBI protocol or by higher level block data integrity checks.

256-PE4-63-D RBI PROC - Block Transfer Bus Talker

An RBI entity that receives a PROC instruction and which:

A) has been configured as a 4-255 block transfer bus talker and,

B) does not reject the instruction,

shall perform the following sequence of actions in the order given:

1) Generate a 4-255 Data Bus response which shall contain the block length indicator value and shall indicate whether errors were detected during the initial execution of the PROC instruction. If no errors are detected, the following steps shall be performed.

2) Transfer the data word referenced by the block transfer page selector combined with the start address pointer onto the 4-255 block transfer bus.

3) Increment the start address pointer to reference the next memory location.

4) Decrement the block length indicator value.

5) Repeat actions 2, 3, and 4 until the block length indicator value is zero.

The resulting output stream on the 4-255 block transfer bus shall be a continuous stream of data bits starting with the first bit of the first data word and finishing with the last bit of the last data word of the block.

256-PE4-64-D RBI PROC - Block Transfer Bus Listener

An RBI entity that receives a PROC instruction and which:

A) has been configured as a 4-255 block transfer bus listener and,
B) does not reject the instruction,

shall perform the following sequence of actions in the order given:

1) Generate a 4-255 Data Bus response which shall contain the block length indicator value and shall indicate whether errors were detected during the initial execution of the PROC instruction. If no errors are detected, the following steps are performed.

   At this point, the RBI entity at the Remote Terminal listens on the block transfer bus. The Bus Controller must issue a proceed instruction to the block transfer bus talker in order to start the block transfer. On detecting activity on the block transfer bus, the following steps are performed.

2) Acquire 16-bits of data from the 4-255 block transfer bus.

3) Write the 16-bits of acquired data to the physical address referenced by the start address pointer and the block transfer page selector.

4) Increment the start address pointer to reference the next memory location.

5) Decrement the block length indicator value.

6) Repeat actions 2, 3, 4, and 5 until the block length indicator value equals zero.

Note 1: It is the detection of activity on the block transfer bus which causes the listening terminal to acquire data, <u>not</u> the detection of the proceed instruction addressed to the block transfer bus talker.

Note 2: If the Block transmission is not continuous, i.e. if activity on the block transfer bus stops before all words of the data block have been received, the BTB_ERROR indicator should be asserted. The BTB_ERROR indicator should remain asserted even if block transfer bus activity subsequently resumes.

256-PE4-65-D <u>RBI PROC – Action on Receipt, Instruction Rejected</u>

An RBI entity that rejects a PROC instruction shall not change the values of its start address pointer or block length indicator, and shall not transfer data to or from memory but shall generate a 4-255 Data Bus response.

256-PE4-66-D <u>RBI PROC - Response</u>

The response to an RBI PROC instruction shall be an RBI proceed response and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the PROC instruction.

9.5.13. Block Transfer Abort, BTB_ABORT.

256-PE4-67-D <u>RBI BTB_ABORT Instruction - Use</u>

The Bus Controller shall issue a Block Transfer Abort, BTB_ABORT, instruction to cause the RBI entity at the addressed Remote Terminal to stop any on-going block transfers on the block transfer bus.

Note: A BTB_ABORT instruction can be sent to individual block transfer bus talkers and listeners, or may be broadcast. Sending a BTB_ABORT instruction to the talker will stop data being transmitted on the block transfer bus which will effectively abort the transfer in any listening terminals. Sending a BTB_ABORT to a listener causes that listener to stop receiving data on the block transfer bus but the talker will continue to transmit and any other listeners will continue to receive the data block. Broadcasting a BTB_ABORT instruction ensures that the talker and all listeners receive it simultaneously thereby stopping the block transfer in all participating terminals.

256-PE4-68-D RBI BTB_ABORT Instruction - Rejection Criteria

There are no rejection criteria associated with the BTB_ABORT instruction. The BTB_ABORT instruction shall not be rejected.

256-PE4-69-D RBI BTB_ABORT Instruction - Action on Receipt

An RBI entity that receives a BTB_ABORT instruction and is configured to transmit or receive data on the block transfer bus shall:

- Abandon any on-going block transfers on the block transfer bus,
- De-assert the BTB_BUSY status indicator,
- Assert the BTB_ABORTED status indicator,
- Reject all further proceed instructions until a new RBIMODE instruction is received,
- Not transmit or receive data on the block transfer bus until a new BTB transfer is initialised,
- Generate a 4-255 Data Bus response.

  Note: The RBI start address pointer, block transfer page selector, and block length indicator shall not be changed by a BTB_ABORT instruction. This allows the Bus Controller to determine how far a block transfer had progressed before the BTB_ABORT instruction was issued.

256-PE4-70-D RBI BTB_ABORT Instruction - Response

The response to a BTB_ABORT instruction shall be an RBI status response indicating the status after executing the BTB_ABORT instruction, and shall indicate whether errors were detected during the execution of the BTB_ABORT instruction.

9.6. RBI Status Flags.

This section describes a set of status indicators which must be maintained by RBI entities.

256-PE4-71-D RBI Status Indicators - Data Transfer Overrun Flag

An RBI entity shall maintain a data transfer overrun flag called DTO_FLAG. This flag shall be asserted if a PROC instruction is received when the length indicator value is zero. If asserted, it shall be de-asserted on any of the following events:

- Receiving an RBIRES instruction,
- A power-on reset,
- Receiving a load length register, LOLE, instruction,
- Receiving an RBISTAT instruction with its reset flag bit set to zero.

256-PE4-72-D RBI Status Indicators - DMA Error Flag

An RBI entity shall maintain a DMA error flag called DMA_ERR. This flag shall be asserted if the RBI entity is unable to access the Remote Terminal memory during a data transfer. DMA_ERR shall be de-asserted on any of the following events:

- Immediately prior to performing any data transfer to or from the Remote Terminal memory,
- Receiving an RBIRES instruction,
- A power-on reset.

256-PE4-73-D RBI Status Indicators - Latched DMA Error Flag

An RBI entity shall maintain a Latched DMA error flag called LDMA_ERR. This flag shall be asserted if the RBI is unable to access the Remote Terminal memory during a data transfer. LDMA_ERR shall be de-asserted on any of the following events:

- Receiving an RBIRES instruction,
- Receiving an RBIMODE instruction,
- Receiving an RBISTAT instruction with its reset flag bit set to zero,
- A power-on reset.

Note: The Latched DMA error flag differs from the normal DMA error flag by remaining asserted until it is explicitly cleared by the Bus Controller. This allows the Bus Controller to detect DMA errors which have occurred during a block transfer which cannot be detected immediately.

256-PE4-74-D RBI Status Indicators - Start Address Rollover Flag

An RBI entity shall maintain a start address rollover flag called SAR_FLAG. This flag shall be asserted if the start address pointer is incremented past all ones during a block transfer. The SAR_FLAG shall be de-asserted on any of the following events:

- Receiving an RBIRES instruction,
- Receiving a LOSA instruction,
- Receiving an RBISTAT instruction with its reset flag bit set to zero,
- A power-on reset.

256-PE4-75-D RBI Status Indicators - Length Zero Flag

An RBI entity shall maintain a status flag called LEN_ZERO, which shall be asserted whenever the length indicator value is zero.

256-PE4-76-D <u>RBI Status Indicators - Start Address Pointer Full Flag</u>

An RBI entity shall maintain a status flag called SAR_FULL, which shall be asserted whenever the start address pointer is at its maximum value, i.e. all ones.

256-PE4-77-D <u>RBI Status Indicators - BTB Busy Flag</u>

An RBI entity shall maintain a status flag called BTB_BUSY, which shall be asserted whenever it is performing a block transfer over the block transfer bus, either as a listener or a talker.

> Note: A terminal is considered to be performing a block transfer from the receipt of the first proceed instruction following the block transfer initialisation, until the block length indicator is decremented to zero or until a BTB_ABORT or RBI_RES instruction is received. The BTB_BUSY flag will therefore be asserted during that time.

256-PE4-78-D <u>RBI Status Indicators - BTB Aborted Flag</u>

An RBI entity shall maintain a status flag called BTB_ABORTED, which shall be asserted following the receipt of a BTB_ABORT instruction. BTB_ABORTED shall be de-asserted on any of the following events:

- Receiving an RBIRES instruction,
- A power-on reset,
- Receiving an RBIMODE instruction.

256-PE4-79-D <u>RBI Status Indicators - BTB Error Flag</u>

An RBI entity shall maintain a status flag called BTB_ERROR, which shall be asserted by a block transfer bus listener if activity on the block transfer bus stops before all words of a block have been received during a BTB block transfer, i.e. if a block transmission is not continuous. BTB_ERROR shall be de-asserted on any of the following events:

- Receiving an RBIRES instruction,
- A power-on reset,
- Receiving an RBIMODE instruction.

9.7. RBI Protocol Procedures.

This section defines a number of procedures to be used as part of the RBI protocol. These procedures are intended to ensure operational consistency at the Bus Controller. Remote Terminals must be capable of handling instructions received in any sequence.

Figure 23 shows a state transition diagram for an RBI block transfer which indicates how the following procedures can be used to complete a block transfer.

256-PE4-80-D RBI Procedures - Instruction Sequencing

RBI entities shall execute RBI instructions issued at any time and in any sequence.

256-PE4-81-D RBI Procedures - Minimum Slot Separation

There shall be no minimum bus slot separation between RBI instructions, i.e. RBI instructions to the same RBI entity may be issued in consecutive bus slots.

256-PE4-82-D RBI Procedures – Remote Terminal Block Transfer Initialisation Procedure

The Bus Controller shall initialise RBI entities at Remote Terminals which are about to participate in a block transfer using the following sequence:

1) Set the block transfer page selector value by sending a LOPS_B instruction. (Not necessary if the block transfer page selector is not used or remains unchanged between block transfers).

2) Set the memory start address for the data block by sending a LOSA instruction.

3) Configure the transfer mode (including the programmed listening address) by sending an RBIMODE instruction.

4) Set the block length by sending a LOLE instruction.

The Bus Controller shall not start a block transfer to or from a Remote Terminal if an error occurs during any of the preceding steps, e.g. if an instruction is rejected or the returned value does not match the value sent in the instruction.

256-PE4-83-D RBI Procedures - RT to BC Transfer via the Response Bus

To perform a block transfer between a Remote Terminal and the Bus Controller via the 4-255 response bus, the following sequence shall be used:

1) Initialise the RBI entity at the data sending Remote Terminal as a 4-255 response bus talker using the Remote Terminal block transfer initialisation procedure.

2) Issue PROC instructions to the RBI entity at the sending Remote Terminal until all words of the block have been transferred. (One PROC instruction is needed for each word of the block).

Note: PROC instructions do not need to be issued in consecutive 4-255 Data Bus slots.

256-PE4-84-D RBI Procedures - BC to RT Transfer via the Interrogation Bus

To perform a block transfer between the Bus Controller and one or more Remote Terminals via the 4-255 interrogation bus, the following sequence shall be used:

1) Initialise the RBI entities at all participating Remote Terminals as 4-255 interrogation bus listeners using the Remote Terminal block transfer initialisation procedure.

2) Issue PROC instructions to the RBI entity at a receiving Remote Terminal with each word of data until all words of the block have been transferred. (One PROC instruction is needed for each word of the block).

3) Repeat action 2 for each receiving Remote Terminal

Note: PROC instructions do not need to be issued in consecutive 4-255 Data Bus slots.

256-PE4-85-D RBI Procedures - RT to RT Transfer using the Bus Controller

To perform a block transfer between two or more Remote Terminals via the 4-255 interrogation and response buses using the BC to store and forward data, the following sequence shall be used:

1) Initialise the RBI entity at the data sending Remote Terminal as a 4-255 response bus talker using the Remote Terminal block transfer initialisation procedure.

2) Initialise the RBI entity at each data receiving Remote Terminal as a 4-255 interrogation bus listener using the Remote Terminal block transfer initialisation procedure.

3) Issue a PROC instruction to the RBI entity at the data sending Remote Terminal to acquire the first word of data at the BC.

4) Issue a PROC instruction to the data sending Remote Terminal to acquire the second word of data at the BC.

5) Issue a PROC instruction to the RBI entity at the data sending Remote Terminal to acquire the next word of data at the BC.

6) Issue a PROC instruction to the RBI entity at each data receiving Remote Terminal, with the data field of the PROC instruction containing the data word acquired from the sender.

7) Repeat actions 5 and 6 until the last data word has been acquired from the sending terminal.

8) Issue a PROC instruction to the RBI entity at each data receiving Remote Terminal, with the data field of the instruction containing the second to last data word of the block.

9) Issue a PROC instruction to the RBI entity at each data receiving Remote Terminal, with the data field of the instruction containing the last data word of the block.

256-PE4-86-D <u>RBI Procedures - Direct RT to RT Transfer via the Response Bus</u>

To perform a direct block transfer between two or more Remote Terminals via the 4-255 response bus, the following sequence shall be used:

1) Initialise the RBI entity at the data sending Remote Terminal as a 4-255 response bus talker using the Remote Terminal block transfer initialisation procedure.

2) Initialise the RBI entity at each data receiving Remote Terminal as a 4-255 response bus listener with the programmed listening address set to the terminal address of the data sending terminal using the Remote Terminal block transfer initialisation procedure.

3) Issue a PROC instruction to RBI entity at the data sending Remote Terminal to transfer one word of the data block.

4) Repeat action 3 until all words of the block have been transferred

256-PE4-87-D <u>RBI Procedures - Transfer between Terminals using the Block Transfer Bus</u>

To perform a block transfer between two or more terminals via the 4-255 block transfer bus, the following sequence shall be used:

1) Initialise the RBI entity at the data sending terminal as a 4-255 block transfer bus talker using the Remote Terminal block transfer initialisation procedure.

2) Initialise the RBI entity at each data receiving terminal as a 4-255 block transfer bus listener using the Remote Terminal block transfer initialisation procedure.

3) Issue a PROC instruction to the RBI entity at each receiving terminal (this causes each receiving RBI entity to assert its BTB_BUSY status indicator and to start listening on the block transfer bus).

4) Issue a PROC instruction to the RBI entity at the data sending terminal (this causes the data sending RBI entity to assert its BTB_BUSY status indicator and to transmit the block on the block transfer bus).

Note: No further transfers can be performed on the block transfer bus until the current transfer is completed or aborted.

256-PE4-88-O RBI Procedures - Block Length Confirmation Procedure

Optionally, to confirm that the length of the data block transferred is correct at a Remote Terminal, the Bus Controller shall issue an RBISTAT instruction to the RBI entity at that terminal. The block length is deemed to be correct if the following flag conditions apply:

- DTO_FLAG - not asserted, i.e. no overrun has occurred,
- LEN_ZERO - asserted, i.e. block length indicator = 0,
- LDMA_ERR - not asserted, i.e. no DMA errors detected,
- BTB_BUSY - not asserted, i.e. BTB block transfer has completed,
- BTB_ABORTED - not asserted, i.e. BTB block transfer completed normally,
- BTB_ERROR – not asserted, i.e. BTB block transfer was continuous.

Any of the flags not being in the stated condition indicates that a block transfer error has occurred.

Note: This check does not verify that the content of the transferred block is correct. The data content can only be properly verified by performing higher level checks.

9.8. RBI Protocol Encoding.

This section specifies the instruction encoding for RBI instructions in 4-255 Data Bus interrogation words and the encoding of the corresponding response words. Protocol Extension 4 uses a standard 4-255 Data Bus interrogation word as shown in Figure 24.

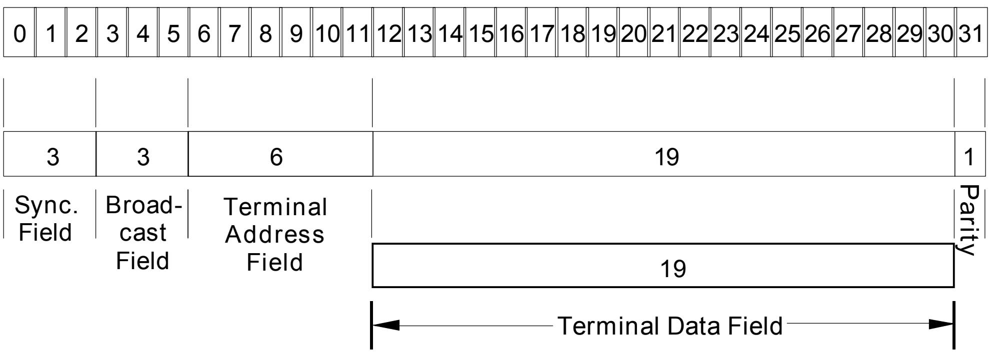

Figure 24 - RBI Protocol Instruction Word Format

9.8.1. Rationale.

The encoding scheme defined here is in line with the 4-255 Data Bus principles in encoding all instructions into a single interrogation word. Furthermore, orthogonal encoding has been used wherever possible. However, given that there are only 19 bits of user data available in an interrogation word and that many instructions to an RBI entity require the transfer of 16 bits of data, the instruction encoding space is very limited. Despite this, the RBI protocol encoding scheme ensures a unique encoding for each instruction and does not require instructions to be transmitted in any given order.

9.8.2. RBI Protocol - Interrogation Word Format.

The RBI Protocol uses a standard 4-255 Data Bus interrogation word as shown in Figure 25.

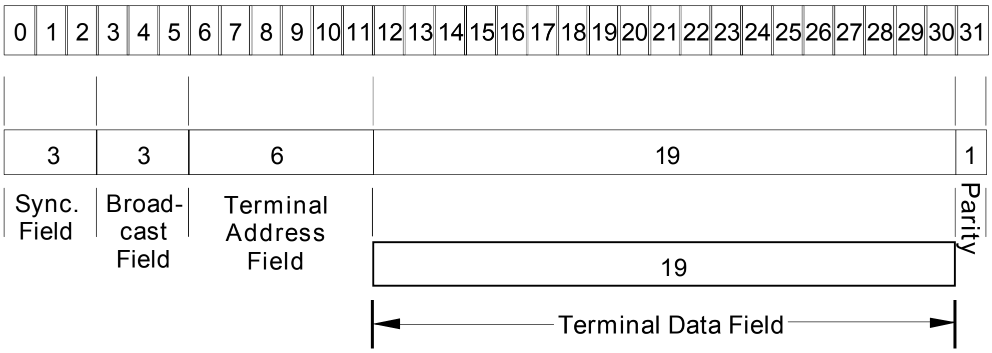

Figure 25 – RBI Protocol Instruction Word Format

256-PE4-89-D <u>RBI Instruction Encoding</u>

For RBI instructions, the terminal data field of the 4-255 interrogation word shall be encoded as shown in Table 9.

Interrogation Word Bit Number																			Instruction
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
0	0	1	0	0	<Length>														Load Block Length Indicator, LOLE
0	0	1	0	1	<User Instruction>														Output User Control, RBIUSR
0	0	1	1	0	0	0	0	0	<RBI Mode Word>										Set RBI Mode, RBIMODE
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	Read Start Address, RESA
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	Read Block Length Indicator, RELE
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	Read Block Transfer Page Selector, REPS_B
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	Block Transfer Abort, BTB_ABORT
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	Read RBI Status, RBISTAT (Reset flags)
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	Read RBI Status, RBISTAT (No flag reset)
0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	Read User Status, USRSTAT
0	0	1	1	0	0	0	0	1	<Any value more than 0110>										Reserved
0	0	1	1	0	0	0	1	0	<Any value less than all										Reserved
0	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	Reset RBI, RBIRES
0	0	1	1	1	0	0	1	1	x	x	x	x	x	x	x	x	x	x	Reserved
0	0	1	1	1	0	1	x	x	x	x	x	x	x	x	x	x	x	x	RBI Feature Group 1
0	0	1	1	1	1	0	x	x	x	x	x	x	x	x	x	x	x	x	RBI Feature Group 2
0	0	1	1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
1	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
1	0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
1	0	1	1	<Block Transfer Page selector>															Load Block Transfer Page Selector, LOPS_B
1	1	0	<Address>																Load Start Address, LOSA
1	1	1	<Data>																Proceed, PROC

Table 9 - RBI Protocol Instruction Set Encoding.

Note: The RBI feature groups are available for RBI device manufacturers to use for device specific instructions.

9.8.2.1. RBI Instruction Mode Word Encoding.

The encoding of the RBI mode word is shown in Figure 26.

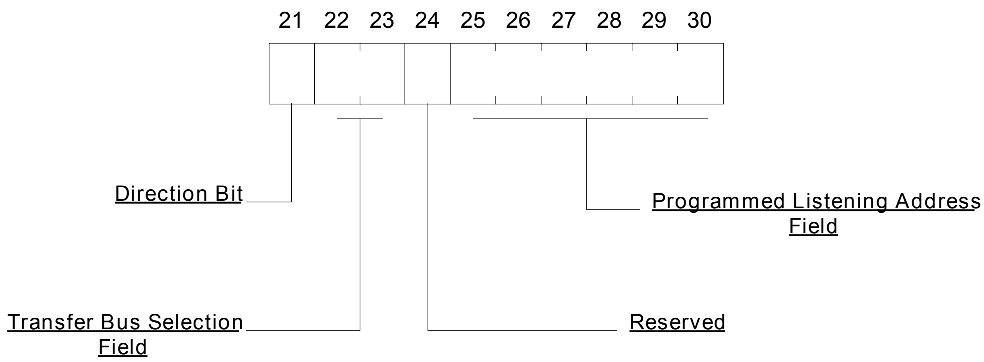

Figure 26 - RBI Mode Word Encoding.

256-PE4-90-D RBI Instruction Encoding - RBI Mode Word

The 10-bit RBI mode word occupies bits 21 to 30 inclusive of the RBIMODE instruction and shall be divided into four components as follows:

- A single direction bit,
- A 2-bit transfer bus selection field,
- A single reserved bit,
- A 6-bit programmed listening address field.

256-PE4-91-D RBI Mode Word - Direction Bit

The RBI mode word direction bit (interrogation word bit 21) shall be set to one (1) to configure the Remote Terminal as a talker. The direction bit shall be set to zero (0) to configure the Remote Terminal as a listener.

256-PE4-92-D RBI Mode Word - Transfer Bus Selection Field

The RBI mode word transfer bus selection field (interrogation word bits 22 and 23) shall be used to select the bus on which a block transfer will be performed. The encoding of this field shall be as shown in Table 10.

Transfer Bus Selection Field Code		Bus to Use for Block Transfer
0	0	Interrogation Bus
0	1	Response Bus
1	0	Block Transfer Bus
1	1	Reserved

Table 10 - RBI Mode Word - Transfer Bus Selection Field Codes.

256-PE4-93-D <u>RBI Mode Word - Reserved Bit</u>

In the RBI mode word, interrogation word bit 24 is reserved and by default shall be set to zero (0).

256-PE4-94-D <u>RBI Mode Word - Programmed Listening Address Field</u>

The RBI mode word programmed listening address field (interrogation word bits 25 to 30 inclusive) shall be set to the terminal address of the data sending terminal when the Remote Terminal is being configured for a block transfer using the programmed listening mode. If the Remote Terminal is not being configured for a programmed listening mode transfer, this field shall be set to zero (000000).

9.8.3. RBI Protocol - Response Word Format.

The response word format used by Protocol Extension 4 complies with the 4-255 Data Bus 21-bit response word format shown in Figure 27.

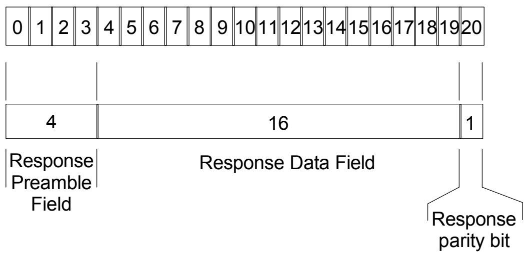

Figure 27 - RBI Protocol Response Word Format.

256-PE4-95-D <u>RBI Response - Preamble Field</u>

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. These four bits shall be used to indicate data quality, to report errors, and to request Bus Controller attention.

The response preamble field for Protocol Extension 4 is shown Figure 28.

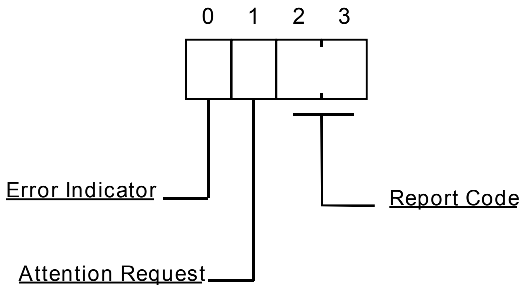

Figure 28 - RBI Instruction - Response Preamble Field.

256-PE4-96-D RBI Response Preamble Field - Error Indicator

The response preamble field error indicator bit (bit 0) shall be set to one (1) to indicate that an error occurred during the execution of the RBI instruction or that the instruction was rejected. This bit shall be set to zero (0) to indicate that the RBI instruction was not rejected and that no errors were detected during its execution.

256-PE4-97-D RBI Response Preamble Field - Attention Request

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE4-98-D RBI Response Preamble Field - Report Code

If no error occurs, the response preamble field report code field (bits 2 and 3) is reserved and by default shall be set to zero (00). If an error occurs, the RBI entity may optionally set the response preamble field report code to indicate the type of error as per Table 11. If an error occurs and the RBI entity does not indicate the type of error, the preamble field report code shall be set to zero (00) to indicate an unspecified error.

Note: If an error occurs a report code value of zero can be used to indicate that an error code is contained in the data field.

Response Preamble Field				Meaning
Error Indicator	Attention Request	Report Code		
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	Reserved
1	X	1	1	Reserved

Table 11 - RBI Protocol - Response Preamble Field Report Codes.

256-PE4-99-D <u>RBI Response Word - Data Field</u>

The response data field occupies bits 4 to 19, inclusive, of the 21-bit response word. This field shall be used to convey the 16-bits of data which are required in response to the RBI instructions.

Note: The data required in response to an RBI instruction includes data produced during normal execution of the instruction as well as optional error codes that may be generated when errors are detected during instruction execution.

256-PE4-100-D <u>RBI Response Word - Parity Bit</u>

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

256-PE4-101-D <u>Response Parity Bit Value - Response Parity Mode = Even</u>

When the RBI response parity mode is ‘Even’, the parity bit shall be set so that the sum of the parity bit plus all ‘one’ bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is ‘None’, the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

9.8.3.1. RBI Status Response Encoding.

256-PE4-102-D <u>RBI Status Response - Response Encoding</u>

The response data field of an RBI status response shall be encoded as shown in Figure 29.

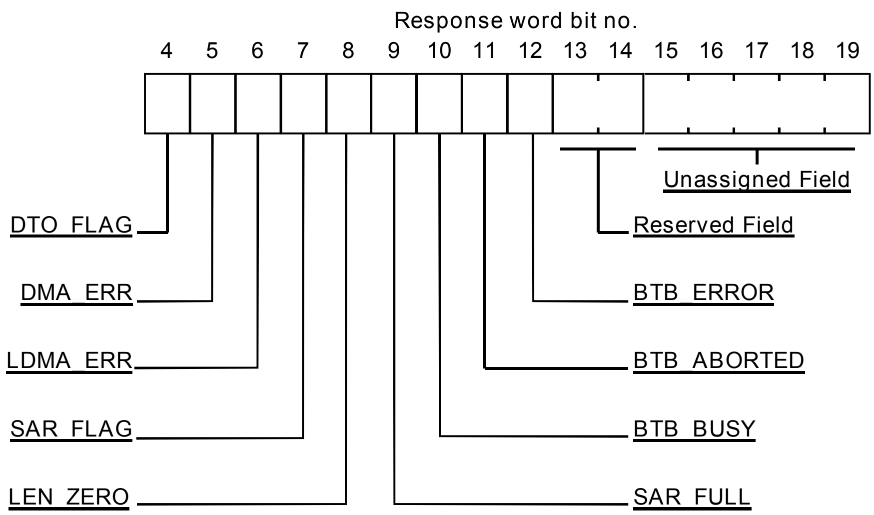

Figure 29 - RBI Status Response Encoding.

256-PE4-103-D RBI Status Response - Response Fields

The RBI status response shall be divided into ten components as follows:

- A DTO_FLAG bit,
- A DMA_ERR bit,
- A LDMA_ERR bit,
- A SAR_FLAG bit,
- A LEN_ZERO bit,
- A SAR_FULL bit,
- A BTB_BUSY bit,
- A BTB_ABORTED bit,
- A BTB_ERROR bit,
- A 2-bit reserved field,
- A 5-bit unassigned field.

256-PE4-104-D RBI Status Response - DTO_FLAG Bit

The RBI status response DTO_FLAG bit (bit 4) shall be set to one (1) if the DTO_FLAG status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-105-D RBI Status Response - DMA_ERR Bit

The RBI status response DMA_ERR bit (bit 5) shall be set to one (1) if the DMA_ERR status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-106-D RBI Status Response - LDMA_ERR Bit

The RBI status response LDMA_ERR bit (bit 6) shall be set to one (1) if the LDMA_ERR status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-107-D RBI Status Response - SAR_FLAG Bit

The RBI status response SAR_FLAG bit (bit 7) shall be set to one (1) if the SAR_FLAG status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-108-D RBI Status Response - LEN_ZERO Bit

The RBI status response LEN_ZERO bit (bit 8) shall be set to one (1) if the LEN_ZERO status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-109-D RBI Status Response - SAR_FULL Bit

The RBI status response SAR_FULL bit (bit 9) shall be set to one (1) if the SAR_FULL status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-110-D RBI Status Response - BTB_BUSY Bit

The RBI status response BTB_BUSY bit (bit 10) shall be set to one (1) if the BTB_BUSY status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-111-D RBI Status Response - BTB_ABORTED Bit

The RBI status response BTB_ABORTED bit (bit 11) shall be set to one (1) if the BTB_ABORTED status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-112-D RBI Status Response - BTB_ERROR Bit

The RBI status response BTB_ERROR bit (bit 12) shall be set to one (1) if the BTB_ERROR status indicator is asserted. Otherwise this bit shall be set to zero (0).

256-PE4-113-D RBI Status Response – Reserved Field

Bits 13 and 14 of the RBI status response are reserved and shall be set to zero (0).

256-PE4-114-D RBI Status Response - Unassigned Field

Bits 15 to 19 inclusive of the RBI status response are unassigned and may be assigned for user specific purposes. Any unused bits shall be set to zero (0).

9.8.3.2. RBI Proceed Response Encoding.

256-PE4-115-D RBI Proceed Response - Response Encoding

The response data field of an RBI proceed response shall be encoded as shown in Figure 30.

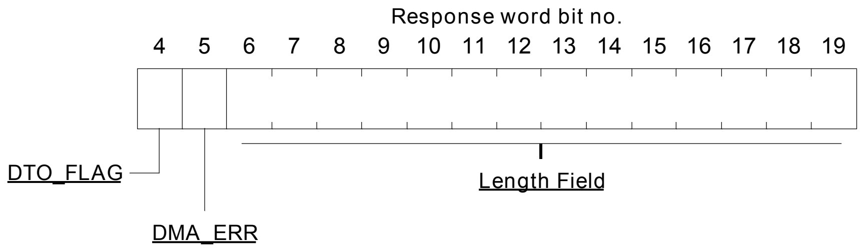

Figure 30 - RBI Proceed Response Encoding.

256-PE4-116-D RBI Proceed Response - Response Fields

The RBI proceed response shall be divided into three components as follows:

- A DTO_FLAG bit,
- A DMA_ERR bit,
- A 14-bit length field.

256-PE4-117-D RBI Proceed Response - DTO_FLAG Bit

The RBI status response DTO_FLAG bit (bit 4) shall be set to one (1) if the DTO_FLAG status indicator is asserted. Otherwise this bit shall be set to zero (0).

Note: Where possible, the value of the DTO_FLAG bit should reflect the status after the execution of the proceed instruction. However this will not be possible for all RBI configurations (e.g. for a BTB transfer). Users should be aware of this when interpreting this flag.

256-PE4-118-D RBI Proceed Response - DMA_ERR Bit

The RBI status response DMA_ERR bit (bit 5) shall be set to one (1) if the DMA_ERR status indicator is asserted. Otherwise this bit shall be set to zero (0).

Note: Where possible, the value of the DMA_ERR bit should reflect the status after the execution of the proceed instruction. However this will not be possible for all RBI configurations (e.g. for a BTB transfer). Users should be aware of this when interpreting this flag.

256-PE4-119-D RBI Proceed Response - Length Field

The RBI proceed response length field (bits 6 to 19 inclusive) shall contain the value of the RBI block length indicator prior to executing the proceed instruction.

9.8.3.3. RBI Mode Response Encoding.

256-PE4-120-D RBI Mode Response - Response Encoding

The response data field of an RBI mode response shall be encoded as shown in Figure 31.

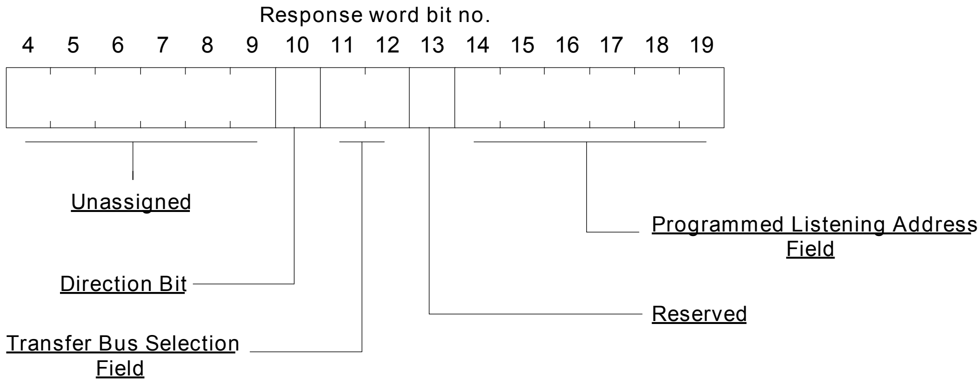

Figure 31 - RBI Mode Response Format.

256-PE4-121-D RBI Mode Response - Response Fields

The RBI mode response shall be divided into five components as follows:

- A 6-bit unassigned field,
- A single direction bit,
- A 2-bit transfer bus selection field,
- A single reserved bit,
- A 6-bit programmed listening address field.

256-PE4-122-D RBI Mode Response - Unassigned Bits

Bits 4 to 9 inclusive of the RBI mode response are unassigned and may be assigned for user specific purposes. Any unused bits shall be set to zero (0).

256-PE4-123-D RBI Mode Response - Direction Bit

The RBI mode response direction bit (bit 10) shall be set to one (1) to indicate that the Remote Terminal is configured as a talker. The direction bit shall be set to zero (0) to indicate that the Remote Terminal is configured as a listener.

256-PE4-124-D RBI Mode Response - Transfer Bus Selection Field

The RBI mode response transfer bus selection field (bits 11 and 12) shall be set to indicate which bus the RBI entity is currently configured to use for block transfers. The encoding of this field shall be as shown in Table 12.

Transfer Bus Selection Field Code		Bus to Use for Block Transfer
0	0	Interrogation Bus
0	1	Response Bus
1	0	Block Transfer Bus
1	1	Reserved

Table 12 - RBI Mode Response - Transfer Bus Selection Field Codes.

Note: The transfer bus selection field code 11 does not correspond to a valid transfer bus selection and therefore should not occur in an RBI mode response.

256-PE4-125-D RBI Mode Response - Reserved Bit

The RBI mode response bit 13 is reserved and by default shall be set to zero (0).

256-PE4-126-D RBI Mode Response - Programmed Listening Address Field

The RBI mode response programmed listening address field (bits 14 to 19 inclusive) shall be set to the Remote Terminals currently configured programmed listening address.

10. Protocol Extension 5 - Serial Time Distribution Protocol.

Protocol Extension 5, the Serial Time Distribution Protocol provides a mechanism for the distribution of time and synchronisation data serially across the 4-255 Data Bus from a Central Time Management System, CTMS, located in the Central Data Management Unit, to all Remote Terminals. The Remote Terminals can use this time and synchronisation data to maintain their Local Time Management System, LTMS, which ensures that a consistent time reference can be accessed locally in each Remote Terminal.

The time distribution mechanism relies on a coherent clock being maintained between the 4-255 Data Bus and both the CTMS and LTMS.

Other parameters can also be transferred in the time distribution message. These could be used to distribute system data (including additional time reference parameters) from the Central Data Management Unit to all Remote Terminals connected to the 4-255 Data Bus.

10.1. Serial Time Distribution Protocol - Terminology.

The following terms are used in the specification of the serial time distribution protocol:

Coarse Time: An integer count of the number of seconds which have elapsed since a mission defined epoch.

CTMS: Central Time Management System, the on-board system responsible for maintaining the spacecraft time reference.

CTMS Message: A message transmitted by the CTMS which contains time and other data which may be used by other on-board systems to maintain a local time reference which is synchronous with the spacecraft time reference.

CTMS Message Bit Interval: A time interval during which one bit of a CTMS message is transmitted.

Fine Time: An integer count of sub-second time elements which can be added to the coarse time to obtain a precision elapsed time since the mission defined epoch.

Initialisation Message: A CTMS message which contains time initialisation information which the LTMS must use to set its local time reference.

LTMS: Local Time Management System, a system which maintains a local time reference which can be accessed by entities in the same terminal.

Synchronisation Instant: The instant at which the time data received in a CTMS message becomes valid.

Update Message: A CTMS message which contains time update information which the LTMS may use to verify its local time reference. The LTMS is not obliged to set its local time reference according to the update message (see also: initialisation message).

The following terms defined in the 4-255 Data Bus specification are also used:

Slot: A time period which corresponds to the transmission of a single interrogation word.

Slot Start Time: The time corresponding to the detection point of the first high level of the synchronisation marker at the local connection point of the terminal.

10.2. Serial Time Distribution Protocol - Operating Principles.

The Serial Time Distribution Protocol uses a single broadcast field bit of the 4-255 Data Bus interrogation word to convey a synchronisation marker and a CTMS message to all Remote Terminals. The message contains the onboard time reference maintained by the CTMS and is transmitted, together with the synchronisation marker once every second.

> Note: The 4-255 Data Bus signalling rate should be referenced to the CTMS master oscillator.

A continuous modulated signal can be recovered from the broadcast bit by latching its value at the slot start time of each slot as shown in Figure 32. The recovered signal is a Manchester encoded data stream. Therefore each bit of data in the recovered signal occupies two 4-255 Data Bus slots. The nominal signalling rate, i.e. assuming the 4-255 Data Bus signalling rate of 2^{19} bps, is therefore 8192 bits per second and the bit time for each recovered data bit is approximately 121µs.

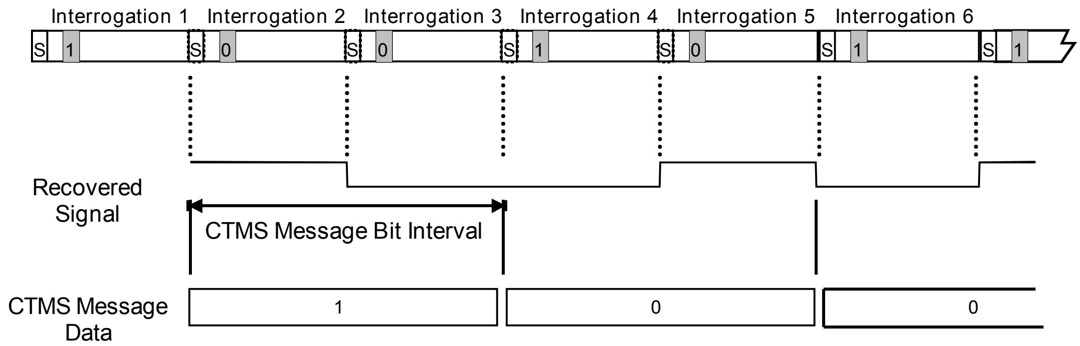

Figure 32 - CTMS Message Recovery from 4-255 Data Bus Broadcast Bit.

The synchronisation marker consists of a modulation pattern which is distinct from any possible data modulation pattern. This marker indicates both the instant that the time data in the preceding CTMS message becomes valid, and that the first bit of a new CTMS message is about to be transmitted.

The CTMS message comprises:

- CTMS status data,
- CTMS coarse time data,
- Optional CTMS pulse data,
- Optional CTMS waveform data,
- Optional parameters.

The coarse time data is formatted as a CCSDS Calendar Unsegmented Time Code (CUC) corresponding to the coarse time valid at the next synchronisation instant. Only coarse time data need be transmitted because the synchronisation marker is transmitted so that it is received by the Remote Terminal at the instant the CTMS fine time counter becomes zero. By transmitting the CTMS message at one second intervals the LTMS time can be kept coherent with the CTMS time. Furthermore, the periodicity of CTMS messages allows the LTMS to verify the time conveyed in the messages.

The vast majority of CTMS messages which are transmitted are update messages. The LTMS uses these messages to check its locally maintained elapsed time counter. If there is a difference between the time provided in the update message and that maintained by the LTMS, the LTMS may choose how it deals with this difference, e.g. it may ignore or it might immediately update its elapsed time reference to match.

However, on occasions the CTMS may send an initialisation message. On receipt of an initialisation message, the LTMS must set its elapsed time to match that contained in the initialisation message. This may result in a time discontinuity being observed at the LTMS.

The transmission of initialisation messages is expected to be a very rare occurrence during normal spacecraft operations.

10.3. Serial Time Distribution Protocol - Signalling and Modulation Requirements.

256-PE5-1-D <u>Serial Time Distribution Protocol - Signal Transmission</u>

CTMS messages shall be transmitted in the interrogation word broadcast field bit 0 (4-255 Data Bus interrogation word bit 3).

256-PE5-2-D <u>Serial Time Distribution Protocol - Signal Modulation</u>

The CTMS message shall be Manchester encoded. Each CTMS message bit interval shall be divided into two equal length segments. Each segment may be a high or a low level.

256-PE5-3-D <u>Serial Time Distribution Protocol - Manchester Segment Length</u>

Each segment of the Manchester encoded signal shall be exactly one 4-255 Data Bus slot in length. Thus the CTMS message bit interval shall be exactly two 4-255 Data Bus slots.

Figure 33 shows the encoding of logical values and the synchronisation marker.

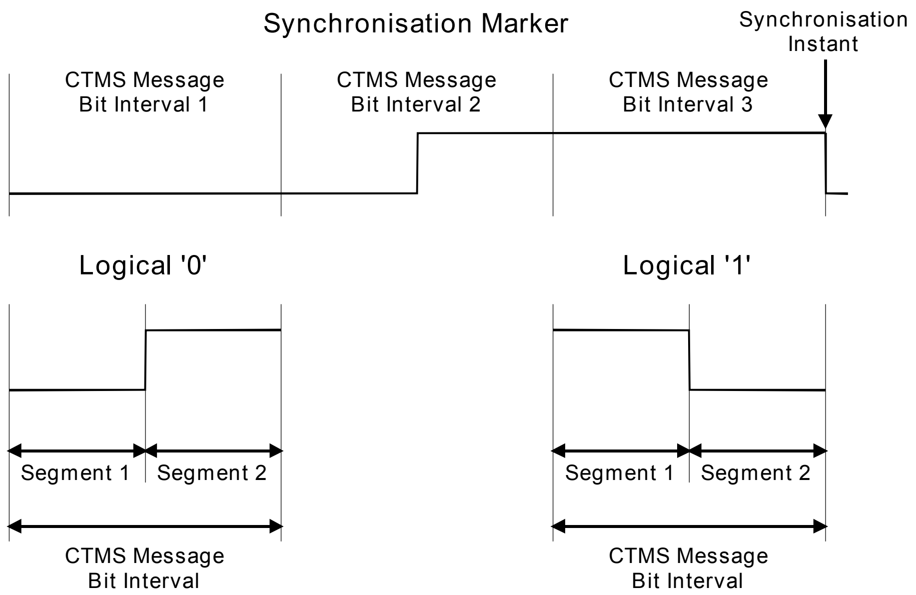

Figure 33 - Encoding of Logical '1', '0', and Synchronisation Marker.

256-PE5-4-D Serial Time Distribution Protocol - Logical '1' Encoding

A logical '1' value shall be encoded as a high level in the first segment of the CTMS message bit interval followed by a low level in the second segment.

256-PE5-5-D Serial Time Distribution Protocol - Logical '0' Encoding

A logical '0' value shall be encoded as a low level in the first segment of the CTMS message bit interval followed by a high level in the second segment.

256-PE5-6-D Serial Time Distribution Protocol - Synchronisation Marker Encoding

The Serial Time Distribution Protocol synchronisation marker shall be encoded in three CTMS message bit intervals. The first CTMS message bit interval of the synchronisation marker shall comprise two segments with a low level. The second CTMS message bit interval of the synchronisation marker shall comprise a logical '0' value. The third CTMS message bit interval of the synchronisation marker shall comprise two segments with a high level.

10.4. Serial Time Distribution Protocol - Synchronisation Marker Requirements.

256-PE5-7-D Serial Time Distribution Protocol - Synchronisation Instant

The synchronisation instant at the receiving terminal shall be the end of the second segment of the third CTMS message bit interval of the synchronisation marker.

Note 1: This reference is always marked by a falling edge because the first CTMS message bit following the synchronisation marker is constrained to be a logical '0'.

Note 2: The broadcast bits of the 4-255 Data Bus interrogation word are valid at the slot start time of the slot following that in which they are transmitted. The Bus Controller therefore writes the synchronisation marker one slot before the fine time value in the CTMS reaches zero.

256-PE5-8-D Serial Time Distribution Protocol - Synchronisation Marker Timing

The synchronisation marker timing reference shall be issued by the Bus Controller so that the synchronisation instant determined at the Remote Terminal corresponds with the instant at which the fine time value in the CTMS becomes zero.

Note: Since the CTMS fine time equals zero only once per second, this implies that the synchronisation marker cannot normally be issued more than once per second. However, in the very rare case of an initialisation message being transmitted it is possible that two synchronisation markers can be transmitted within one second.

256-PE5-9-D <u>Serial Time Distribution Protocol - Synchronisation Instant Significance</u>

The synchronisation instant at the Remote Terminal shall signify that the coarse time contained in the previous CTMS message matches that in the CTMS and that the CTMS fine time has become zero. The synchronisation instant shall also mark the start of a new CTMS message.

10.5. Serial Time Distribution Protocol - CTMS Message Requirements and Formats.

10.5.1. CTMS Message.

The Serial Time Distribution Protocol CTMS message format is shown in Figure 34. The following requirements define that format.

> Note: During serial transmission of the CTMS message on the 4-255 Data Bus, a parity bit is inserted after every eight data bits. The actual number of bits transmitted is therefore greater than the number of data bits in the CTMS message.

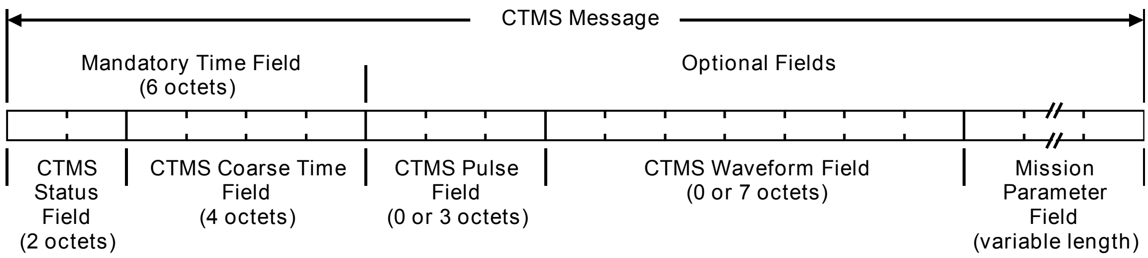

Figure 34 - Serial Time Distribution Protocol CTMS Message Format.

256-PE5-10-D <u>Serial Time Distribution Protocol - CTMS Message Format</u>

The serial CTMS message shall comprise a mandatory Time Field. This may be followed by any combination of three optional fields called the CTMS Pulse Field, the CTMS Waveform Field, and the Mission Parameter Field.

> Note: The presence of the CTMS pulse and CTMS waveform fields is indicated by flags in the mandatory time field of the CTMS message. The inclusion of the optional fields is determined by the CTMS on a message-by-message basis.

10.5.2. Time Field.

256-PE5-11-D CTMS Message - Time Field Format

The Time Field of a CTMS message is mandatory in every message and shall comprise a two octet CTMS Status Field followed by a four octet CTMS Coarse Time Field.

256-PE5-12-D Time Field - CTMS Status Field

The mandatory CTMS status field comprises two octets (16-bits) of CTMS status data. The format of the CTMS status field is shown in Figure 35.

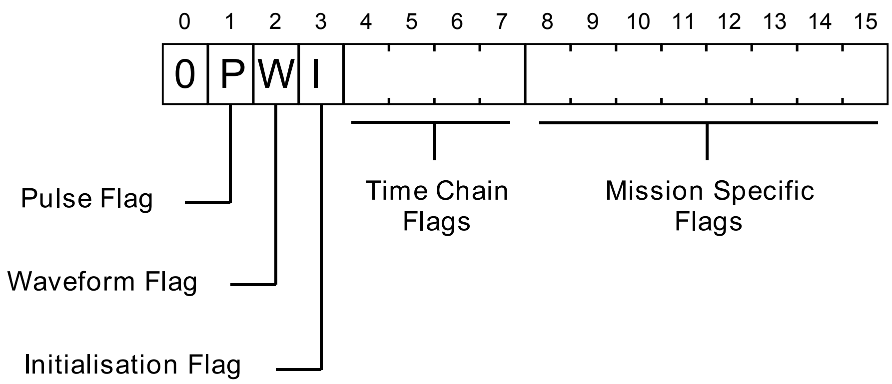

Figure 35 - CTMS Status Field.

256-PE5-13-D CTMS Status Field - Zero Bit

The first bit of the CTMS status bit, bit 0, shall be set to zero (0) to ensure reliable detection of the synchronisation marker.

256-PE5-14-D CTMS Status Field - Pulse Flag

Bit 1 of the CTMS status field is the pulse flag. It shall be set to one (1) to indicate that the optional CTMS pulse field is present in this CTMS message. Otherwise this bit shall be set to zero (0).

256-PE5-15-D CTMS Status Field - Waveform Flag

Bit 2 of the CTMS status field is the waveform flag. It shall be set to one (1) to indicate that the optional CTMS waveform field is present in this CTMS message. Otherwise this bit shall be set to zero (0).

256-PE5-16-D CTMS Status Field - Initialisation Flag

Bit 3 of the CTMS status data field is the initialisation flag. It shall be set to one (1) to indicate that this CTMS message is a time initialisation message. A value of zero in this bit shall indicate that this CTMS message is a time update message.

256-PE5-17-D CTMS Status Field - Time Chain Flags

Bits 4 to 7 inclusive of the CTMS status data field are reserved for use as time chain flags. By default these flags shall be set to zero (0).

256-PE5-18-D CTMS Status Field - Mission Specific Flags

Bits 8 to 15 inclusive of the CTMS status data field are unassigned and may be used to convey mission specific flags. By default any unused bits shall be set to zero (0).

256-PE5-19-D CTMS Coarse Time Field

The four octets of coarse time data contained in the CTMS coarse time field of the CTMS message shall comprise the coarse time code element of the T-field of a CCSDS Calendar Unsegmented Time Code format, i.e. a 32-bit binary coded unsigned integer representing the number of whole seconds elapsed since the time code epoch. This gives the number of elapsed seconds since the time code epoch with an ambiguity period of 2^{32} seconds (about 136 years).

256-PE5-20-D CTMS Coarse Time Epoch

The time code epoch for the CTMS message coarse time data shall be mission defined.

Note: The implicit CCSDS CUC time code P-field value is therefore 00101100 binary.

256-PE5-21-D CTMS Coarse Time Value

The coarse time value contained in the CTMS message shall correspond to the value of coarse time at the CTMS at the next synchronisation instant.

10.5.3. Optional CTMS Pulse Field.

The presence of the CTMS pulse field is indicated by the pulse flag in the CTMS status field. This section defines the format of the CTMS pulse field. The CTMS pulse field can be used to provide a pulse start time for Remote Terminals which have a pulse generator capability and enables pulses having a precise, synchronous relationship with the CTMS to be generated by a Remote Terminal.

256-PE5-22-D CTMS Pulse Field - Presence

The presence of the CTMS pulse field in the CTMS message shall be optional on a message-by-message basis. Its presence shall be indicated by the pulse flag in the CTMS status field.

256-PE5-23-D CTMS Pulse Field - Use

The CTMS pulse field shall be included in the CTMS message to provide a sub-second resolution fine time value which, when added to the coarse time specified in the time field of that CTMS message, results in a precision pulse start time which is synchronous with the CTMS.

> Note: A single pulse is generated locally following each CTMS message containing a pulse field. A pulse will not be generated if the CTMS pulse field is absent.

256-PE5-24-D CTMS Pulse Field - Format

The CTMS pulse field shall comprise three octets of fine time data encoded as per the fine time elements of a CCSDS CUC.

> Note: The fixed length of three octets gives a resolution down to 2^{-24} seconds (about 60ns) for the fine time value which can be communicated to the Remote Terminal.

10.5.4. Optional CTMS Waveform Field.

The presence of the CTMS waveform field is indicated by the waveform flag in the CTMS status field. The CTMS waveform field can be used to specify waveform characteristics to Remote Terminals which have waveform generation capabilities.

The waveform generation specifications in this field control the shape and polarity of the waveform generated but cannot specify a precise phase relationship with the CTMS. However, an external pulse is usually used to start the generation of a waveform and the timing of this pulse can be accurately specified using the CTMS message pulse field. Therefore, by using the pulse field and the waveform field together, it is possible to specify the waveshape, the polarity, and the phase relationship with the CTMS.

Figure36 shows typical waveshapes which can be specified in the CTMS waveform field.

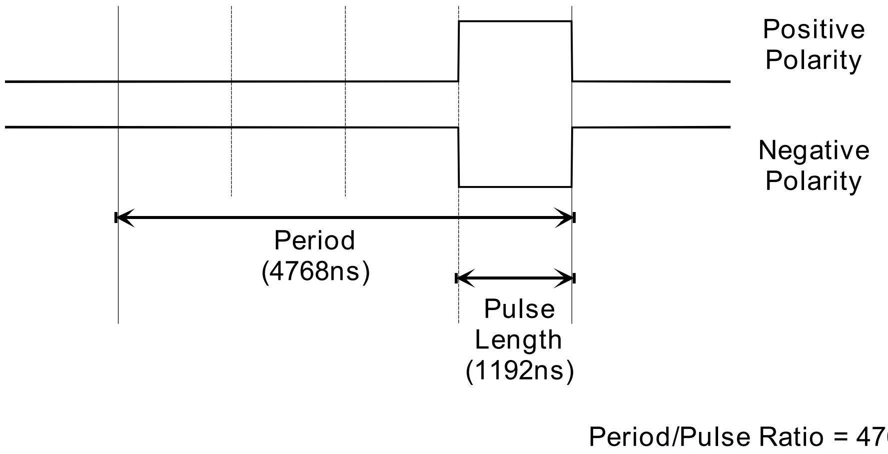

Figure 36 - Typical Waveshapes Specified in the CTMS Waveform Field.

256-PE5-25-D <u>CTMS Waveform Field - Presence</u>

The presence of the CTMS waveform field in the CTMS message shall be optional on a message-by-message basis. Its presence shall be indicated by the waveform flag in the CTMS status field.

256-PE5-26-D <u>CTMS Waveform Field - Use</u>

The CTMS waveform field shall be included in the CTMS message to provide waveform parameters to a Remote Terminal's local waveform generation function.

> Note: This optional field need not be transmitted in every CTMS message. Once a waveform generator has been configured by a CTMS message with a CTMS waveform field, an external signal must be provided to start the generation of a waveform.

256-PE5-27-D <u>CTMS Waveform Field - Format</u>

The CTMS waveform field shall consist of seven octets of data arranged in the following fields:

- A 2-octet coarse length data field,
- A 3-octet fine length data field,
- A 1-bit polarity indicator,
- A 15-bit duty cycle ratio field.

This format is shown in Figure 37.

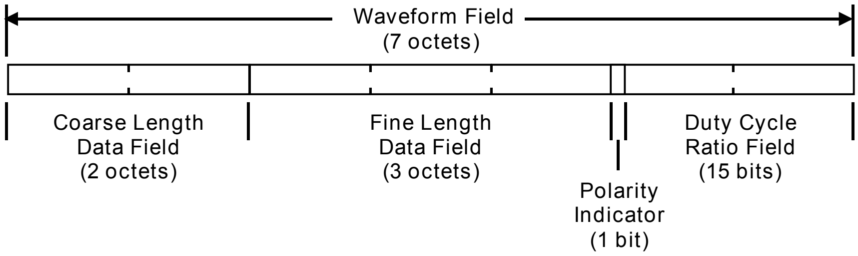

Figure 37 - CTMS Message Waveform Field Format.

256-PE5-28-D <u>CTMS Waveform Field - Coarse Length Data Field</u>

The coarse length data field shall consist of two octets containing a binary coded integer count value of the whole number of seconds of the pulse duration. The range of time values which can be provided in this field is 0 up to 2^{16}-1 seconds.

256-PE5-29-D <u>CTMS Waveform Field - Fine Length Data Field</u>

The fine length data field shall consist of three octets containing a binary coded integer count value indicating the sub-second part of the pulse duration.

Note: The fine length data which can be communicated to the Remote Terminal has a resolution of 2^{24} seconds (approximately 60ns).

256-PE5-30-D <u>CTMS Waveform Field - Polarity Indicator Bit</u>

The polarity indicator bit shall be set to zero (0) to indicate that the waveform consists of positive going pulses. This bit shall be set to one (1) to indicate that the waveform consists of negative going pulses.

256-PE5-31-D <u>CTMS Waveform Field - Duty Cycle Ratio Field</u>

The duty cycle ratio field shall consist of 15-bits containing a binary coded integer count value indicating the duty cycle ratio of the output waveform.

256-PE5-32-D <u>CTMS Waveform Field - Duty Cycle Ratio Value and Stop Command</u>

The duty cycle ratio value shall be zero or in the range 2 to 2^{15}-1. Values in the range 2 to 2^{15}-1 are valid duty cycle ratio values. The value zero shall be used to instruction the Remote Terminal waveform generator to stop outputting a waveform. The value 1 is reserved.

10.5.5. Optional Mission Parameter Field.

This section contains the requirements relating to the mission parameter field.

The mission parameter field may be included in the CTMS message and used to convey data to Remote Terminals.

There is no defined method of indicating the presence of the mission parameter field although users may choose to use one of the unassigned mission specific flag bits of the CTMS status field for this purpose. It is also possible to use the mission specific flag bits to provide selective addressing of the mission parameter field to certain terminals.

256-PE5-33-O Mission Parameter Field - Use

Optionally, a mission parameter field shall be appended to the CTMS message and shall be used to convey user defined/mission specific data to Remote Terminals.

> Note: The mission parameter field can be appended to every CTMS message or may only be appended occasionally. The means of indicating the presence of the mission parameter field in a given message is user defined.

256-PE5-34-D Mission Parameter Field - Maximum Length

The maximum length of the mission parameter field transmitted on a 4-255 data bus using the normal signalling rate of 2^{19} bps shall be 893 octets (eight hundred and ninety-three).

> Note: For other signalling rates different maximum length may apply. The maximum length can be calculated as the maximum number of whole octets which can be transmitted between the transmission of the last bit of the 7-octet CTMS waveform field and the first bit of the next synchronisation marker assuming that the 3-octet CTMS pulse field is also present in the message.

10.6. Serial Time Distribution Protocol - Message Transmission.

256-PE5-35-D Serial Time Distribution Protocol - CTMS Message Transmission

The Serial Time Distribution Protocol CTMS message shall be transmitted bit serially using consecutive 4-255 Data Bus interrogation words, i.e. without inter-bit gaps.

256-PE5-36-D CTMS Message Transmission - Parity Check Bits

After every eight bits of CTMS message data transmitted, a parity check bit shall be transmitted. This bit shall apply to the eight bits preceding it and shall be an even parity bit, i.e. the sum of 1's in the preceding eight bits plus the parity bit itself shall be even.

256-PE5-37-D CTMS Message Transmission - Message/Synchronisation Marker Relationship

The first bit of the Serial Time Distribution Protocol CTMS message shall be transmitted in the bit interval immediately following the synchronisation marker.

256-PE5-38-D CTMS Message Transmission - Inter-message Symbols

After the last bit of the CTMS message has been transmitted, a continuous stream of logical zeroes shall be transmitted until the next synchronisation marker is due.

> Note: The stream of Manchester encoded zero bits can be interpreted at the receiver as octets with the correct even parity. However, the number of octets will not be an integer because of the parity bits and because of the length of the synchronisation marker. The last octet before the synchronisation marker will be incomplete but this shall not be flagged as an error.

11. Protocol Extension 6 - Virtual Circuit Protocol

Protocol Extension 6, the Virtual Circuit Protocol, provides virtual circuit capabilities for communication between 4-255 Data Bus terminals. A virtual circuit connection is a communication link between two terminals which appears to each of those terminals like a dedicated link through which they can transfer data at any time.

PE-6 achieves this over the 4-255 Data Bus by allocating bus slots for one terminal to send data to another. These slots can be allocated periodically, i.e. every nth slot, or opportunistically, i.e. whenever bus bandwidth permits. The slots allocated to a particular sender/receiver pair are identified by a Virtual Circuit Identifier, VCI, combined with the Terminal Address, TA, of the sending terminal.

Terminals communicating via virtual circuits have the capability to transfer streams of data. These streams may be formatted, i.e. organised into blocks, packets or other structures, or may be entirely unformatted. PE-6 provides mechanisms for indicating format delimiters within the data stream to ensure that synchronisation can be maintained between the sender and the receiver.

PE-6 also enables special, user defined virtual circuit control codes to be carried in the data stream.

Virtual circuit communication is suitable for applications where one specific terminal has a frequent need to transmit data to another specific terminal. Data generation may be synchronous or asynchronous, PE-6 readily supports both. It should be noted however, that the sending terminal will need to provide FIFO style data buffering and the receiving terminal must be ready to receive data at all times. The size of the senders buffer will depend on several factors including the rate of data production, the allocation method for the virtual circuit slots (periodic or opportunistic), and the effective bandwidth of the virtual circuit.

PE-6 makes use of the response preamble field of the 4-255 Data Bus response word in order to signal data quality, format delimiting, and word type information between the communicating terminals and to provide a mechanism for requesting service from the Bus Controller.

PE-6 specifies:

- A set of instructions to set-up, maintain, and dismantle a virtual circuit,
- Procedures for performing virtual circuit data transfers,
- Instruction set encoding in 4-255 Data Bus words.

11.1. Virtual Circuit Protocol - Operating Principles.

The virtual circuit capability is provided over the 4-255 Data Bus by using the response bus listening capability of Remote Terminals. The data sender terminal responds to interrogations received from the Bus Controller but these responses can also be acquired by appropriately configured response bus listeners. By placing the data to be transferred in the responses, that data is effectively transferred between the sender and the listening terminals.

A virtual circuit can have only one sending Remote Terminal but may have one or many listening terminals which receive the data.

11.1.1. Virtual Circuit Identification.

Each virtual circuit on a 4-255 Data Bus is identified by a Virtual Circuit Identifier, VCI, and the Terminal Address, TA, of the data sender terminal. Figure 38 illustrates the use of the VCI and TA to identify virtual circuits. When transmitting data on a virtual circuit, only the VCI is used to identify the virtual circuit (because the terminal may not be aware of its own address at the Virtual Circuit Protocol level). Because of the range of VCI values, Remote Terminals can transmit data on a maximum of eight outgoing virtual circuits and each outgoing circuit must have a unique VCI. When receiving data from an incoming virtual circuit, both the VCI and the TA are used for circuit identification. A single Remote Terminal could therefore theoretically receive data on eight virtual circuits from each of the other Remote Terminals on the 4-255 Data Bus.

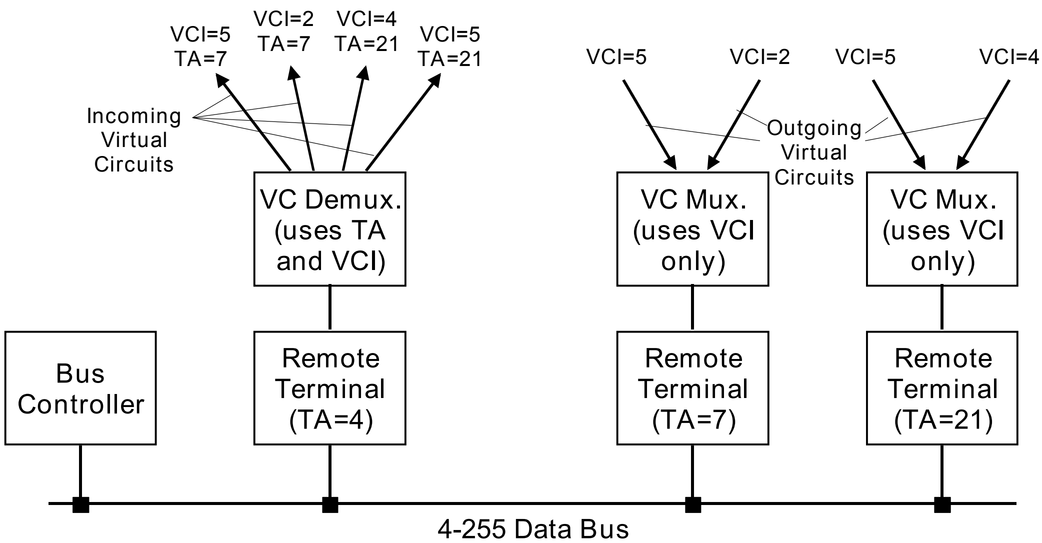

Figure 38 - Identification of Incoming and Outgoing Virtual Circuits.

11.1.2. Virtual Circuit Operation.

There are three operational phases during the lifetime of a virtual circuit:

- the connection phase, during which the terminals are configured and a virtual circuit established,
- the active phase, during which data is transferred on the virtual circuit,
- the disconnection phase, during which a virtual circuit is dismantled.

11.1.2.1. Virtual Circuit Connection Phase.

During the virtual circuit connection phase the Bus Controller configures the sender and receiver terminals by providing a VCI to all participating terminals and by informing the receiving terminals of the sending Remote Terminal address. This is done using a virtual circuit connect instruction, VC_CONNECT.

11.1.2.2. Virtual Circuit Transfer Phase.

During the transfer phase the Bus Controller issues virtual circuit proceed instructions, VC_PROCEED, to the data sender terminal. These instructions contain the VCI. There is no limit to the number of proceeds which can be sent for a given virtual circuit.

The data sender Remote Terminal decodes the VC_PROCEED and generates a response word containing the data to be transferred. The response preamble field bits must be set to indicate the quality of the data contained in the response, the type of word which is being transferred, and whether the Remote Terminal requires servicing by the Bus Controller.

Data receiving Remote Terminals have been programmed to monitor interrogations sent to the data sender terminal and must recognise the VC_PROCEED with the senders terminal address, and the VCI. When they do so, they acquire the corresponding response from the response bus and use the response preamble field bits to determine whether the data is valid or not, whether it is a data word or a control word, and whether it is a particular word of a format. Virtual circuit control words are words which control the flow of data within the data stream but are not part of the data themselves.

> Note: If an Enhanced 4-255 Data Bus Modem is used to implement the Physical and Access Control Sub-layer services, virtual circuit receiving RTs must set the Interrogation Word Receive Service address decoding level to 1 or 0 to ensure that they acquire the proceed instructions sent to virtual circuit sending RTs.

11.1.2.3. Virtual Circuit Disconnection Phase.

During the disconnection phase of a virtual circuit, the Bus Controller stops issuing VC_PROCEED instructions and issues virtual circuit disconnect, VC_DISCONNECT, instructions to all participating Remote Terminals. The VC_DISCONNECT instruction contains the terminal address of the sender and the VCI of the virtual circuit to be disconnected. After receiving a VC_DISCONNECT, terminals will ignore VC_PROCEED instructions with that VCI and TA unless they have been reconnected by repeating the virtual circuit connection phase.

> Note: If several Remote Terminals are receiving data on the same virtual circuit, each receiving terminal can be disconnected without disturbing the other receiving Remote Terminals.

11.1.3. Bi-directional Virtual Circuit Connections.

The PE-6 virtual circuit provides only uni-directional communication between a data sender Remote Terminal and one or more data receiving Remote Terminals on each virtual circuit. In order to achieve a bi-directional virtual circuit communication between two terminals, two virtual circuits must be configured between them, one for communication in each direction. The same VCI could be used for both virtual circuits but each would have a different sender TA.

11.2. Virtual Circuit Protocol - General Requirements.

This section defines a set of general requirements which must be met by Protocol Extension 6 entities.

256-PE6-1-O Virtual Circuit Protocol – Optional Response Parity Mode

Optionally, the virtual circuit protocol response parity mode at a given Remote Terminal shall be ‘Even’.

> Note 1: A response parity mode of ‘Even’ means that a virtual circuit sender is required to generate a parity value covering the response data field of virtual circuit protocol responses. A virtual circuit receiver is required to check the parity value in received virtual circuit protocol responses. To make use of parity checking across a virtual circuit, both the sender and receiver must be set to even parity mode. The even parity value may also be used by the Bus Controller to verify the quality of the data contained in the response.
>
> Note 2: The selection of response parity mode is per Remote Terminal and not per virtual circuit. Care must therefore be taken to select an appropriate response parity mode considering all of the virtual circuits which must be supported by a given Remote Terminal.
>
> Note 3: By default, the PE-6 data response parity mode at a Remote Terminal is ‘None’.

11.3. Virtual Circuit Protocol Instruction Set.

256-PE6-2-D VC Protocol - Instruction Set

The virtual circuit protocol shall comprise the following instructions:

- Virtual Circuit Connect, VC_CONNECT
- Virtual Circuit Proceed, VC_PROCEED
- Virtual Circuit Disconnect, VC_DISCONNECT

11.3.1. Virtual Circuit Connect, VC_CONNECT.

256-PE6-3-D VC_CONNECT Instruction - Use

The Bus Controller shall issue a Virtual Circuit Connect, VC_CONNECT, instruction to connect the addressed Remote Terminal to a virtual circuit. Two parameters shall be contained in this instruction:

- A 6-bit data value which indicates whether the Remote Terminal is to send or receive data on the virtual circuit, and if it is to receive data, contains the Terminal Address, TA, of the corresponding data sender Remote Terminal.
- A 3-bit Virtual Circuit Identifier, VCI, which, in combination with the data sender TA, uniquely identifies a virtual circuit.

  Note: The 3-bit VCI means that each data sender Remote Terminal can output data to a maximum of eight virtual circuits. However, data receiving Remote Terminals use both the VCI and the data sender Remote Terminal TA to identify a virtual circuit and can therefore receive data from up to eight virtual circuits per Terminal Address.

256-PE6-4-D VC_CONNECT Instruction – Data Sender TA Value

Data sender terminal address values in the range 1 to 63 decimal (000001 to 111111 binary) shall be permitted in the VC_CONNECT instruction.

256-PE6-5-D VC_CONNECT Instruction - VCI Value

Virtual circuit identifier values in the range 0 to 7 decimal (000 to 111 binary) shall be permitted in the VC_CONNECT instruction.

256-PE6-6-D VC_CONNECT Instruction - Rejection Criteria

There are no rejection criteria associated with the VC_CONNECT instruction. The VC_CONNECT instruction shall not be rejected.

256-PE6-7-D VC_CONNECT Instruction - Action on Receipt

The action of a PE-6 entity that receives a VC_CONNECT instruction depends on the value of the 6-bit data sender terminal address contained in the instruction as follows:

- If the 6-bit data sender terminal address is 63 (111111), the PE-6 entity shall configure as a virtual circuit data sender using the VCI provided in the VC_CONNECT instruction and shall generate a 4-255 Data Bus response.
- If the 6-bit data sender terminal address is any value other than 0 (000000) or 63 (111111), the PE-6 entity shall configure as a data receiver, shall acquire the response data from all subsequently detected VC_PROCEED interrogations with the specified VCI sent to the Remote Terminal with that TA, and shall generate a 4-255 Data Bus response.

256-PE6-8-D VC_CONNECT Instruction - Response

The response to a VC_CONNECT instruction shall be a VC control response and shall indicate whether errors were detected during the execution of the VC_CONNECT instruction.

11.3.2. Virtual Circuit Proceed, VC_PROCEED.

256-PE6-9-D VC_PROCEED Instruction - Use

The Bus Controller shall issue a Virtual Circuit Proceed, VC_PROCEED, instruction to cause data to be transferred across a virtual circuit. The VC_PROCEED instruction shall be sent to the Remote Terminal configured as a virtual circuit data sender and shall contain a VCI value to indicate which virtual circuit the VC_PROCEED instruction applies to.

256-PE6-10-D VC_PROCEED Instruction - Rejection Criteria

A PE-6 entity shall reject a VC_PROCEED instruction if the virtual circuit identified by the VCI in that instruction is not currently connected at that terminal.

256-PE6-11-D VC_PROCEED Instruction - Action on Receipt, VC Sender, Instruction not Rejected

A PE-6 entity that receives a VC_PROCEED instruction and does not reject it shall generate a 4-255 Data Bus response containing a 16-bit word of data for the specified virtual circuit in the response to the instruction. If there is no data ready for that virtual circuit, any convenient value (e.g. last value, zero, user defined error code, etc.) shall be sent and the response preamble field shall be set to indicate that no data was ready.

256-PE6-12-D VC_PROCEED Instruction - Action on Receipt, VC Sender, Instruction Rejected

A PE-6 entity that receives a VC_PROCEED instruction and rejects it shall generate a 4-255 Data Bus response but shall not transfer any virtual circuit data in that response.

256-PE6-13-D VC_PROCEED Instruction - Action on Detection, VC Receiver

A PE-6 entity which is configured as a data receiver for a given virtual circuit which identifies a VC_PROCEED instruction containing the VCI for that virtual circuit and addressed to the data sender for that virtual circuit shall acquire the corresponding response from the response bus, but shall not generate a 4-255 Data Bus response to the VC_PROCEED instruction.

256-PE6-14-D VC_PROCEED Instruction - Response

The response to a VC_PROCEED instruction shall contain the data or control word to be transmitted on the specified virtual circuit and shall indicate the type of word, whether the instruction was rejected, and whether errors occurred during the execution of the VC_PROCEED instruction.

> Note: The VC receiver should ignore VC_PROCEED responses which indicate an error. The value in the response data field can therefore be replaced by a user defined error code by a VC sender which detects an error during the execution of a VC_PROCEED instruction.

11.3.3. Virtual Circuit Disconnect, VC_DISCONNECT.

256-PE6-15-D VC_DISCONNECT Instruction - Use

The Bus Controller shall issue a Virtual Circuit Disconnect, VC_DISCONNECT, instruction to disable a virtual circuit connection with the specified VCI and data sender TA. To disable a virtual circuit completely, VC_DISCONNECT instructions shall be sent to each terminal participating in that connection.

256-PE6-16-D VC_DISCONNECT Instruction – Data Sender TA Value

Data sender terminal address values in the range 1 to 63 decimal (000001 to 111111 binary) shall be permitted in the VC_DISCONNECT instruction.

256-PE6-17-D VC_DISCONNECT Instruction - VCI Value

Virtual circuit identifier values in the range 0 to 7 decimal (000 to 111 binary) shall be permitted in the VC_DISCONNECT instruction.

256-PE6-18-D VC_DISCONNECT Instruction - Rejection Criteria

A VC_DISCONNECT instruction with a VCI value of 0 (000000) shall be rejected.

256-PE6-19-D VC_DISCONNECT Instruction - Action on Receipt, Instruction not Rejected

The action of a PE-6 entity that receives a VC_DISCONNECT instruction and does not reject it depends on the value of the 6-bit data sender terminal address contained in the instruction as follows:

- If the 6-bit data sender terminal address is 63 (111111), the PE-6 entity shall cease to send data on the virtual circuit with the specified VCI and shall generate a 4-255 Data Bus response,
- If the 6-bit data sender terminal address is any value other than 0 (000000) or 63 (111111), the PE-6 entity shall cease to receive data on the virtual circuit with the specified VCI and sender terminal address and shall generate a 4-255 Data Bus response.

256-PE6-20-D VC_DISCONNECT Instruction - Action on Receipt, Instruction Rejected

A PE-6 entity that receives a VC_DISCONNECT instruction and rejects it shall not disconnect any currently established virtual circuits but shall generate a 4-255 Data Bus response.

256-PE6-21-D VC_DISCONNECT Instruction - Response

The response to a VC_DISCONNECT instruction shall be a VC control response and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the VC_DISCONNECT instruction.

11.4. Virtual Circuit Protocol - Operating Procedures.

This section defines the procedures to be used to operate virtual circuits using PE-6 instructions over the 4-255 Data Bus.

11.4.1. Virtual Circuit Connection Phase.

256-PE6-22-D Virtual Circuit Procedures - VC Connection Establishment

The virtual circuit connection phase shall be performed using the following sequence:

1) Issue a VC_CONNECT instruction with the appropriate data sender terminal address and VCI value to each data receiving terminal of the virtual circuit.

2) Issue a VC_CONNECT instruction with the data sender terminal address 63 (111111) and the appropriate VCI value to the data sender Remote Terminal of the virtual circuit.

11.4.2. Virtual Circuit Disconnection Phase.

256-PE6-23-D Virtual Circuit Procedures - VC Disconnection

To perform a virtual circuit disconnection, the following sequence shall be used:

1) Issue a VC_DISCONNECT instruction with the data sender terminal address 63 (111111) and the appropriate VCI value to the data sender Remote Terminal of the virtual circuit.

2) Issue a VC_DISCONNECT instruction with the data sender terminal address and the appropriate VCI value to each participating data receiving Remote Terminal of the virtual circuit.

11.5. Virtual Circuit Protocol Encoding.

This section describes the encoding of the virtual circuit protocol instructions and response in 4-255 Data Bus interrogation and response words.

11.5.1. Virtual Circuit Protocol - Interrogation Word Format.

In Protocol Extension 6, the terminal data field of the 4-255 Data Bus interrogation word is divided into the following three fields:

- A 3-bit mode identifier field,
- A 4-bit instruction group identifier field,
- A 3-bit instruction code field,
- A 6-bit instruction data field,
- A 3-bit VCI field,

This format is shown in Figure 39.

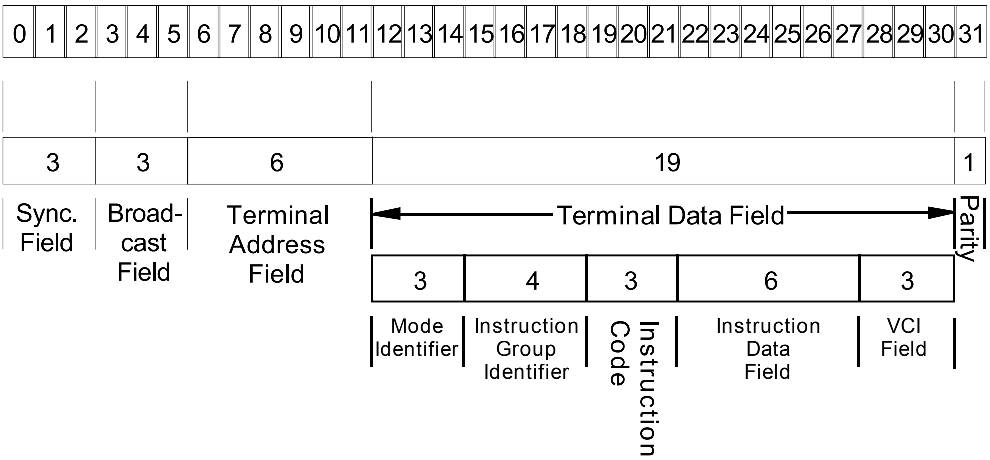

Figure 39 - Virtual Circuit Protocol Interrogation Word Format.

256-PE6-24-D <u>Virtual Circuit Protocol - Mode Identifier Field</u>

The mode identifier field occupies bits 12, 13, and 14 of the interrogation word and is used as a mode identifier. This field shall be set to 000 binary.

256-PE6-25-D <u>Virtual Circuit Protocol - Instruction Group Identifier Field</u>

The instruction group identifier field occupies interrogation word bits 15 to 18 inclusive. This field shall be set to 1111 binary.

256-PE6-26-D <u>Virtual Circuit Protocol - Instruction Code Field</u>

The 3-bit instruction code field occupies interrogation word bits 19 to 21 inclusive. This field shall be used to identify the operation to be performed on receipt of a virtual circuit protocol instruction.

256-PE6-27-D <u>Virtual Circuit Protocol - Instruction Data Field</u>

The 6-bit instruction data field occupies interrogation word bits 22 to 27 inclusive. This field is used to convey data associated with the virtual circuit protocol instructions.

256-PE6-28-D <u>Virtual Circuit Protocol - VCI Field</u>

The 3-bit VCI field occupies interrogation word bits 28 to 30 inclusive. This field shall be set to the VCI value of the selected virtual circuit.

256-PE6-29-D <u>Virtual Circuit Protocol - Instruction Code Assignments</u>

The encoding of the virtual circuit protocol instructions shall be as shown in Table 13.

Mode Identifier = 000 Group Identifier = 1111												
Interrogation word bit number												Code allocation
19	20	21	22	23	24	25	26	27	28	29	30	
Instruc-tion Code			Instruction Data Field						VCI Field			
1	0	1	0	0	0	0	0	0	VCI			VC_PROCEED
1	0	1	Data Sender TA or 111111						VCI			VC_CONNECT
1	1	1	Data Sender TA or 111111						VCI			VC_DISCONNECT

Table 13 - Virtual Circuit Protocol Code Allocations.

11.5.2. Virtual Circuit Protocol - Response Word Format.

The response word format used by the virtual circuit protocol complies with the 4-255 Data Bus 21-bit response word format shown in Figure 40.

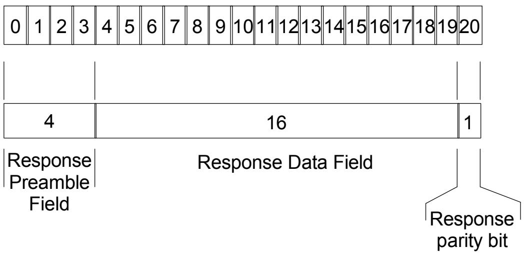

Figure 40 - Virtual Circuit Protocol Response Word Format.

256-PE6-30-D <u>Virtual Circuit Protocol - Response Preamble Field</u>

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. This field shall be used to indicate errors, to indicate virtual channel word types, and to obtain Bus Controller attention.

The response preamble field for Protocol Extension 6 is shown in Figure 41.

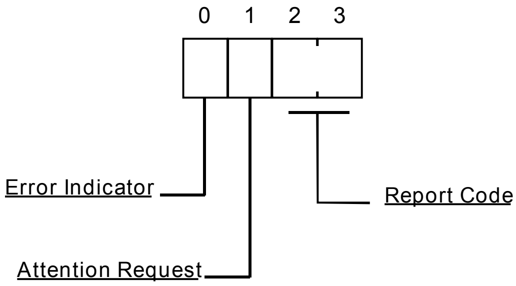

Figure 41 - Virtual Circuit Protocol - Response Preamble Field.

256-PE6-31-D PE-6 Response Preamble Field - Error Indicator

The response preamble field error indicator bit (bit 0) shall be set to zero (0) if a virtual circuit protocol instruction is not rejected and the requested operation is completed successfully. This bit shall be set to one (1) if a virtual circuit protocol instruction is rejected or if an error is detected during the execution of the instruction.

Note: The error indicator must be set to one in the case that a data sender terminal receives a VC_PROCEED instruction but does not have data available for transmission.

256-PE6-32-D PE-6 Response Preamble Field - Attention Request

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE6-33-D PE-6 Response Preamble Field - Report Code (VC_CONNECT and VC_DISCONNECT)

For responses to the VC_CONNECT, and VC_DISCONNECT instructions the response preamble field report code (bits 2 and 3) shall be used as follows:

- If no error is detected, the report code is reserved and by default shall be set to zero (00).
- If an error is detected, the report code shall be used to indicate the type of error as per Table 14.

Response Preamble Field Error Indicator	Attention Request	Report Code		Meaning
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	Reserved
1	X	1	1	Reserved

Table 14 -Virtual Circuit Protocol - Report Codes for VC_CONNECT and VC_DISCONNECT Instructions.

Note: The 'unspecified error' report code can be used to indicate that an error code is contained in the data field.

256-PE6-34-D <u>PE-6 Response Preamble Field - Report Code (VC_PROCEED)</u>

For responses to the VC_PROCEED instruction the response preamble field report code (bits 2 and 3) shall be used as defined in Table 15.

Response Preamble Field Error Indicator	Attention Request	Report Code		Meaning
0	X	0	0	Continuation Data Word
0	X	0	1	Last Word of Format
0	X	1	0	First Word of Format
0	X	1	1	Virtual Circuit Control Word
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	No Data Ready
1	X	1	1	Reserved

Table 15 -Virtual Circuit Protocol - Report Codes for VC_PROCEED Instruction.

Note 1: When no error is detected, the word type is used by the receiver to determine how to handle the data. Continuation data words contain normal data transmitted in the virtual circuit data stream. Unstructured data streams might use only continuation data words. Last word and first word codes can be used to delimit formatted data units (e.g. frames or packets) within the stream. Virtual circuit control words are user defined and can be used to convey protocol control information across the virtual circuit. By convention, if the length of data units in the stream is one, the report code shall be set to indicate 'last word', i.e. 01.

Note 2: The 'unspecified error' report code can be used to indicate that an error code is contained in the data field.

256-PE6-35-D <u>PE-6 Response Word - Data Field, No Error</u>

The response word data field occupies bits 4 to 19, inclusive, of the response word. If a virtual circuit protocol instruction is executed without error, this 16-bit field shall be used to convey data in response to virtual circuit protocol instructions.

256-PE6-36-O <u>PE-6 Response Word - Optional Data Field Error Codes</u>

Optionally, if errors are detected during the execution of a virtual circuit protocol instruction and the response preamble field error indicator has been set to one, the 16-bit response data field shall be used to convey a user defined error code.

256-PE6-37-D <u>PE-6 Response Word - Parity Bit</u>

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

256-PE6-38-D <u>Response Parity Bit Value - Response Parity Mode = Even</u>

When the response parity mode is ‘Even’, the parity bit shall be set so that the sum of the parity bit plus all ‘one’ bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is ‘None’, the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

11.5.2.1. VC Control Response Format.

The VC control response format is used for the VC_CONNECT and VC_DISCONNECT instructions.

256-PE6-39-D <u>VC Control Response - Data Field Format</u>

The VC control response data field format shall be as shown Figure 42.

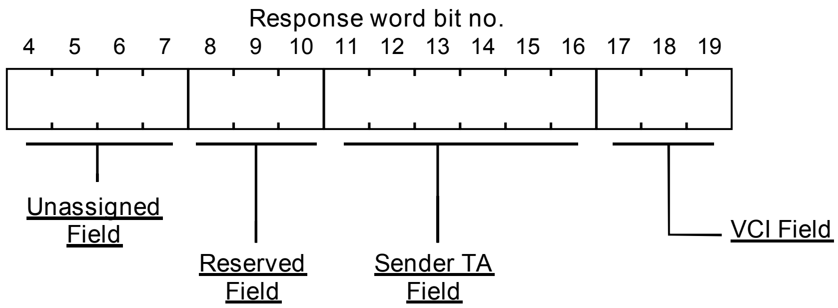

Figure 42 - VC Control Response Data Field Format.

256-PE6-40-D VC Control Response - Response Fields

The VC control response data field shall be divided into three components as follows:

- A 4-bit unassigned field,
- A 3-bit reserved field,
- A 6-bit sender TA field,
- A 3-bit VCI field.

256-PE6-41-D VC Control Response - Unassigned Field

Bits 4 to 7 inclusive of the VC control response are unassigned and may be assigned for user specific purposes. Any unused bits shall be set to zero (0).

256-PE6-42-D VC Control Response - Reserved Field

Bits 8 to 10 inclusive of the VC control response are reserved and shall be set to zero (0).

256-PE6-43-D VC Control Response – Data Sender TA Field

The data sender TA field (bits 11 to 16 inclusive) shall be set to the data sender terminal address specified in the virtual circuit control instruction (VC_CONNECT or VC_DISCONNECT) which provoked the response.

256-PE6-44-D VC Control Response - VCI Field

The VCI field (bits 17 to 19 inclusive) shall be set to the VCI value specified in the virtual circuit control instruction (VC_CONNECT or VC_DISCONNECT) which provoked the response.

12. Protocol Extension 7 - Memory Access Protocol

Protocol Extension 7, the Memory Access Protocol, MAP, provides a mechanism to read and write data directly to memory in a Remote Terminal, one word at a time.

12.1. Memory Access Protocol - Terminology.

The following terms are used in the specification of the Memory Access Protocol:

Address Offset: An address which refers to a single memory location within a memory page.

Memory Mage Selector: Part of an address which identifies a memory page. Each page may consist of many memory locations.

Physical Address: A memory address produced by combining a memory page selector and an offset which refers to a physical location in the local memory.

12.2. Memory Access Protocol - Operating Principles.

The Memory Access Protocol provides a means for a Bus Controller to access the memory of a Remote Terminal for reading and writing data. Two special instructions are also provided for semaphore manipulation, which operate on a read-modify-write principle.

12.2.1. Memory Addresses.

The Memory Access Protocol provides word-by-word access to memory in a 4-255 Data Bus Remote Terminal independently of any processors or controllers operating in that terminal. For complex Remote Terminals an extensive range of addresses requiring more than 16-bits may be needed. MAP addresses are therefore made up of two components, a memory page selector and an address offset. These are combined to produce a physical address which identifies a location in the terminal memory. The precise way in which these components are combined (and even whether the memory page selector is used at all) is implementation dependent.

A complete address (memory page selector and address offset value) and the data needed for memory write operations cannot be transferred in a single 4-255 Data Bus interrogation. Therefore, the MAP entity in a Remote Terminal must maintain a local memory page selector and an address memory offset pointer. The Memory Access Protocol provides instructions to initialise and read these.

12.2.2. Memory Read Operation.

To read a single Remote Terminal memory location, the memory page selector maintained by the MAP entity at that Remote Terminal is first loaded with the appropriate value using a load memory page selector instruction, and then a memory read instruction containing the address offset is sent. The Remote Terminal MAP entity combines its local memory page selector with the address offset provided in the instruction to generate a physical address. The data value at this address is then read and placed in the 4-255 Data Bus response to the memory read instruction.

The Bus Controller can read any memory location on a given page of a Remote Terminals memory once the memory page selector has been loaded simply by issuing memory read instructions with the appropriate address offset. I.e. there is no need to re-load the memory page selector between memory reads to the same page.

12.2.3. Memory Write Operation.

To write a single Remote Terminal memory location, the memory page selector and memory offset pointer maintained by the MAP entity at that Remote Terminal are first loaded with appropriate memory page selector and address offset values using load memory page selector and load memory offset pointer instructions respectively. A memory write instruction is then sent containing the data to be written. On receiving this instruction, the Remote Terminal MAP entity generates a physical address by combining its local memory page selector and memory offset pointer values and writes the data provided in the memory write instruction to that physical address.

The behaviour of the memory offset pointer after performing a memory write is configurable through a memory offset pointer mode instruction. It can be configured to increment automatically, decrement automatically, or to remain unchanged.

To perform a series of memory write operations to contiguous locations on the same page of a Remote Terminal memory, the memory page selector and address pointer are loaded with appropriate values and the memory offset pointer is configured either to auto-increment or to auto-decrement. A series of memory write instructions is then sent, each containing a data word. Following each memory write instruction the Remote Terminal MAP entity updates its memory offset pointer so that each data word is written into an adjacent physical memory location.

To prevent unexpected behaviour due to memory offset pointer rollover, memory write instructions are rejected if the memory offset pointer has been incremented beyond all ones in auto-increment mode, or decremented below all zeroes in auto-decrement mode. Memory write instructions will continue to be rejected until the memory offset pointer is re-loaded.

12.2.4. Semaphore Control.

Semaphores are typically used for inter-process signalling and synchronisation. For example, a data producing process may indicate when data is available for collection by a data consuming process. For reliable signalling, operations on semaphores need to be protected so that one process cannot modify a semaphore while it is being used by another process. Typically, this protection is achieved by using atomic read-modify-write accesses to semaphores.

Two special instructions for semaphore control are provided in the Memory Access Protocol. These are the semaphore-up and semaphore-down instructions and both operate as read-modify-write instructions.

On receipt of a semaphore-up instruction, the MAP entity reads the data from the memory location referenced by the memory offset pointer combined with the memory page selector, increments the data by one, and then writes it back to the same memory location. The semaphore down operates in the same way except that the data is decremented by one. In both cases, the memory should not be released to the Remote Terminal processor in between the read and write operation to ensure that it cannot modify the semaphore during that period.

12.3. Memory Access Protocol - General Requirements.

This section defines a set of general requirements which must be met by Remote Terminals which implement the Memory Access Protocol.

256-PE7-1-D MAP - Memory Page Selector

A MAP entity shall maintain a memory page selector which shall be used in combination with an address offset to produce a physical address for memory accesses.

256-PE7-2-D MAP – Memory Offset Pointer

A MAP entity shall maintain a memory offset pointer which shall be used to provide the address offset for memory write operations.

256-PE7-3-D MAP – Memory Offset Pointer Operating Modes

The memory offset pointer shall be configurable to operate in three modes:

- Auto-incrementing mode, the memory offset pointer shall increment to point to the next memory location after each memory write operation,
- Auto-decrementing mode, the memory offset pointer shall decrement to point to the next memory location after each memory write operation,
- Fixed address mode, the memory offset pointer value shall remain unchanged after each write operation.

256-PE7-4-D MAP Memory Offset Pointer – Mode after Power-up/Reset

Following power-up or after a reset, the memory offset pointer shall be in fixed mode, i.e. the memory offset pointer value will remain unchanged after a memory write instruction.

256-PE7-5-D MAP Memory Offset Pointer – Value after Power-up/Reset.

Following power-up or after a reset, the value of the MAP memory offset pointer shall be zero.

256-PE7-6-D MAP Memory Mage Selector – Value after Power-up/Reset.

Following power-up or after a reset, the value of the MAP memory page selector shall be zero.

256-PE7-7-O MAP – Optional Response Parity Mode

Optionally, the MAP response parity mode at a given Remote Terminal shall be 'Even'.

> Note 1: A response parity mode of 'Even' means that the MAP entity is required to generate a parity value covering the response data field of MAP responses. The even parity value is used by the MAP entity receiving the response to verify the quality of the response data.
>
> Note 2: By default, the MAP response parity mode at a Remote Terminal is 'None'.

12.4. Memory Access Protocol Instruction Set.

The Memory Access Protocol instruction set includes the instructions which are needed to write to and read from a memory in a Remote Terminal via the 4-255 Data Bus.

256-PE7-8-D MAP - Instruction Set

The Memory Access Protocol shall comprise the following instructions:

- Load Memory Page Selector, LOAD_MPS,
- Load Memory Offset Pointer, LOAD_MOP,

- Read Memory Page Selector, READ_MPS,
- Read Memory Offset Pointer, READ_MOP,
- Memory Read, MEMRD,
- Memory Write, MEMWRI,
- Memory Offset Pointer Mode, MOP_MODE,
- Semaphore-up, SEM_UP,
- Semaphore-down, SEM_DOWN,

12.4.1. Load Memory Page Selector, LOAD_MPS.

256-PE7-9-D LOAD_MPS Instruction - Use

The Bus Controller shall issue a Load Memory Page Selector, LOAD_MPS, instruction to set the memory page selector value in the MAP entity at the addressed Remote Terminal. The page selector value identifies the page containing the memory storage location to be used during MAP read and write operations at that Remote Terminal.

256-PE7-10-D LOAD_MPS Instruction - Rejection Criteria

There are no rejection criteria associated with the LOAD_MPS instruction. The LOAD_MPS instruction shall not be rejected.

256-PE7-11-D LOAD_MPS Instruction - Action on Receipt

A MAP entity that receives a LOAD_MPS instruction shall load the value contained in the instruction into the memory page selector and shall generate a 4-255 Data Bus response.

256-PE7-12-D LOAD_MPS Instruction - Memory Page Selector Value

Any memory page selector value up to 15-bits in length shall be permitted in the LOAD_MPS instruction data field.

> Note: Care should be taken that the memory page selector value provided in a LOAD_MPS instruction corresponds to an actual physical memory area in the Remote Terminal.

256-PE7-13-D LOAD_MPS Instruction - Response

The response to a LOAD_MPS instruction shall contain the memory page selector value after the execution of the instruction, and shall indicate whether any errors were detected during the execution of the LOAD_MPS instruction.

12.4.2. Load Memory Offset Pointer, LOAD_MOP.

256-PE7-14-D <u>LOAD_MOP Instruction - Use</u>

The Bus Controller shall issue a Load Memory Offset Pointer, LOAD_MOP, instruction in order to set the memory offset pointer in the MAP entity at the addressed Remote Terminal.

256-PE7-15-D <u>LOAD_MOP Instruction - Rejection Criteria</u>

There are no rejection criteria associated with the LOAD_MOP instruction. The LOAD_MOP instruction shall not be rejected.

256-PE7-16-D <u>LOAD_MOP Instruction - Action on Receipt</u>

A MAP entity that receives a LOAD_MOP instruction shall load the value contained in the instruction into the memory offset pointer and shall generate a 4-255 Data Bus response.

256-PE7-17-D <u>LOAD_MOP Instruction - Address Value</u>

Any 16-bit address offset value shall be permitted in the LOAD_MOP instruction data field.

Note: Care should be taken that the address offset value provided, when combined with the locally maintained memory page selector, corresponds to a physical memory location on the Remote Terminal.

256-PE7-18-D <u>LOAD_MOP Instruction - Response</u>

The response to a LOAD_MOP instruction shall contain the value of the memory offset pointer after executing the instruction, and shall indicate whether errors were detected during the execution of the LOAD_MOP instruction.

12.4.3. Read Memory Page Selector, READ_MPS.

256-PE7-19-D <u>READ_MPS Instruction - Use</u>

The Bus Controller shall issue a Read Memory Page Selector, READ_MPS, instruction in order to read the current value of the memory page selector in the MAP entity at the addressed Remote Terminal.

256-PE7-20-D <u>READ_MPS Instruction - Rejection Criteria</u>

There are no rejection criteria associated with the READ_MPS instruction. The READ_MPS instruction shall not be rejected.

256-PE7-21-D READ_MPS Instruction - Action on Receipt

A MAP entity that receives a READ_MPS instruction shall generate a 4-255 Data Bus response containing the current value of its memory page selector. The value of the memory page selector shall not be changed by this operation.

256-PE7-22-D READ_MPS Instruction - Response

The response to a READ_MPS instruction shall contain the value of the memory page selector and shall indicate whether errors were detected during the execution of the READ_MPS instruction.

12.4.4. Read Memory Offset Pointer, READ_MOP.

256-PE7-23-D READ_MOP Instruction - Use

The Bus Controller shall issue a Read Memory Offset Pointer, READ_MOP, instruction to read the current value of the memory offset pointer in the MAP entity at the addressed Remote Terminal.

256-PE7-24-D READ_MOP Instruction - Rejection Criteria

There are no rejection criteria associated with the READ_MOP instruction. The READ_MOP instruction shall not be rejected.

256-PE7-25-D READ_MOP Instruction - Action on Receipt

A MAP entity that receives a READ_MOP instruction shall generate a 4-255 Data Bus response containing the current value of the memory offset pointer. The value of the memory offset pointer shall not be changed by this operation.

256-PE7-26-D READ_MOP Instruction - Response

The response to a READ_MOP instruction shall contain the value of the memory offset pointer and shall indicate whether errors were detected during the execution of the READ_MOP instruction.

12.4.5. Memory Read, MEMRD.

256-PE7-27-D MEMRD Instruction - Use

The Bus Controller shall issue a Memory Read, MEMRD, instruction to read a single 16-bit data word from the memory of the addressed Remote Terminal. The MEMRD instruction shall include the 16-bit memory address offset. The physical address from which data is to be read is determined by combining the memory address offset provided in the instruction with the memory page selector value which is maintained locally by the MAP entity.

256-PE7-28-D MEMRD Instruction - Rejection Criteria

There are no rejection criteria associated with the MEMRD instruction. The MEMRD instruction shall not be rejected.

256-PE7-29-D MEMRD Instruction - Action on Receipt

A MAP entity that receives a MEMRD instruction shall read the data value contained in the physical address referenced by the memory address offset provided in the instruction data field combined with the locally maintained memory page selector. The MAP entity shall then generate a 4-255 Data Bus response containing that data value.

256-PE7-30-D MEMRD Instruction - Address Value

Any 16-bit address offset value shall be permitted in the MEMRD instruction data field.

Note: Care should be taken that the address offset provided, when combined with the locally maintained memory page selector, corresponds to a physical memory location on the Remote Terminal.

256-PE7-31-D MEMRD Instruction - Response

The response to a MEMRD instruction shall contain 16-bits of data read from the Remote Terminal physical memory location produced by combining the memory page selector with the address offset provided in the MEMRD instruction, and shall indicate whether errors were detected during the execution of the MEMRD instruction.

12.4.6. Memory Write, MEMWRI.

256-PE7-32-D MEMWRI Instruction - Use

The Bus Controller shall issue a Memory Write, MEMWRI, instruction to write a 16-bit data value into the memory of the addressed Remote Terminal. The physical address to which the data is to be written is determined by combining the memory page selector and memory offset pointer values which are maintained locally by the MAP entity.

256-PE7-33-D MEMWRI Instruction - Rejection Criteria, Auto-increment Mode

When the memory offset pointer is in auto-increment mode, a MEMWRI instruction shall be rejected if a previous MEMWRI instruction caused the memory offset pointer to increment beyond all ones and a LOAD_MOP instruction has not subsequently been received.

256-PE7-34-D MEMWRI Instruction - Rejection Criteria, Auto-decrement Mode

When the memory offset pointer is in auto-decrement mode, a MEMWRI instruction shall be rejected if a previous MEMWRI instruction caused the memory offset pointer to decrement below all zeroes and a LOAD_MOP instruction has not subsequently been received.

256-PE7-35-D MEMWRI Instruction - Action on Receipt, Instruction not Rejected

A MAP entity that receives a MEMWRI instruction and does not reject it shall write the 16-bit data value provided in the instruction data field into the physical address referenced by the locally maintained memory offset pointer and memory page selector. If the memory offset pointer is in auto-increment mode it shall be incremented after writing the data. If the memory offset pointer is in auto-decrement mode it shall be decremented after writing the data. If the memory offset pointer is in fixed address mode it shall not be changed. The MAP entity shall then generate a 4-255 Data Bus response containing the new value of the memory offset pointer.

256-PE7-36-D MEMWRI Instruction - Action on Receipt, Instruction Rejected

A MAP entity that receives a MEMWRI instruction and rejects it shall not write data to memory and shall not update the memory offset pointer value. It shall generate a 4-255 Data Bus response.

256-PE7-37-D MEMWRI Instruction - Data Value

Any 16-bit data value shall be permitted in the MEMWRI instruction data field.

256-PE7-38-D MEMWRI Instruction - Response

The response to a MEMWRI instruction shall contain the value of the memory offset pointer after executing the instruction, and shall indicate whether the instruction was rejected and whether errors were detected during the execution of the MEMWRI instruction.

12.4.7. Memory Offset Pointer Mode, MOP_MODE.

256-PE7-39-D MOP_MODE Instruction - Use

The Bus Controller shall issue a Memory Offset Pointer Mode, MOP_MODE, instruction to configure or enquire the memory offset pointer mode in the MAP entity at the addressed Remote Terminal. The memory offset pointer mode may be one of the following values:

- Auto-increment,
- Auto-decrement,
- Fixed Address,
- Enquire.

Note: The enquire setting is used to obtain the current memory offset pointer mode in the MAP entity without changing the mode.

256-PE7-40-D MOP_MODE Instruction - Rejection Criteria

There are no rejection criteria associated with the MOP_MODE instruction. The MOP_MODE instruction shall not be rejected.

256-PE7-41-D MOP_MODE Instruction - Action on Receipt, Mode = Auto-increment, Auto-decrement, or Fixed Address

A MAP entity that receives a MOP_MODE instruction with the mode specified as auto-increment, auto-decrement, or fixed address shall set the memory offset pointer mode to that which is specified and shall report the new selected mode in a 4-255 Data Bus response.

256-PE7-42-D MOP_MODE Instruction - Action on Receipt, Mode = Enquire

A MAP entity that receives a MOP_MODE instruction with the mode specified as enquire shall not change the current memory offset pointer mode but shall generate a 4-255 Data Bus response containing the current memory offset pointer mode setting.

256-PE7-43-D MOP_MODE Instruction - Response

The response to a MOP_MODE instruction shall be a MOP_MODE response containing the memory offset pointer mode after executing the instruction, and shall indicate whether errors were detected during the execution of the MOP_MODE instruction.

12.4.8. Semaphore-up, SEM_UP.

256-PE7-44-D SEM_UP Instruction - Use

The Bus Controller shall issue a Semaphore-up, SEM_UP, instruction to increment a semaphore at the addressed Remote Terminal. The physical address of the semaphore is determined by combining the memory address offset and the memory page selector values maintained locally by the MAP entity.

256-PE7-45-D SEM_UP Instruction - Rejection Criteria

There are no rejection criteria associated with the SEM_UP instruction. The SEM_UP instruction shall not be rejected.

256-PE7-46-D SEM_UP Instruction - Action on Receipt

A MAP entity that receives a SEM_UP instruction shall:

- Lock the memory to prevent access by other entities at the Remote Terminal,
- Read the data value contained in the physical address referenced by the locally maintained memory address offset and memory page selector values,
- Increment the data value by one,
- Write the incremented data value back to the physical address referenced by the locally maintained memory address offset and memory page selector values,
- Release the memory.

The MAP entity shall then generate a 4-255 Data Bus response containing the incremented data value.

> Note: If the semaphore value is all ones prior to incrementing, it shall be incremented to zero.

256-PE7-47-D SEM_UP Instruction - Response

The response to a SEM_UP instruction shall contain the 16-bit semaphore value after executing the instruction, and shall indicate whether errors were detected during the execution of the SEM_UP instruction.

12.4.9. Semaphore-down, SEM_DOWN.

256-PE7-48-D SEM_DOWN Instruction - Use

The Bus Controller shall issue a Semaphore-down, SEM_DOWN, instruction to decrement a semaphore at the addressed Remote Terminal. The physical address of the semaphore is determined by combining the memory address offset and the memory page selector values maintained locally by the MAP entity.

256-PE7-49-D SEM_DOWN Instruction - Rejection Criteria

There are no rejection criteria associated with the SEM_DOWN instruction. The SEM_DOWN instruction shall not be rejected.

256-PE7-50-D SEM_DOWN Instruction - Action on Receipt

A MAP entity that receives a SEM_DOWN instruction shall:

- Lock the memory to prevent access by other entities at the Remote Terminal,
- Read the data value contained in the physical address referenced by the locally maintained memory address offset and memory page selector values,
- Decrement the data value by one,
- Write the decremented data value back to the physical address referenced by the locally maintained memory address offset and memory page selector values,
- Release the memory.

The MAP entity shall then generate a 4-255 Data Bus response containing the decremented data value.

> Note: If the semaphore value is zero prior to decrementing, it shall be incremented to all ones.

256-PE7-51-D SEM_DOWN Instruction - Response

The response to a SEM_DOWN instruction shall contain the 16-bit semaphore value after executing the instruction, and shall indicate whether errors were detected during the execution of the SEM_DOWN instruction.

12.5. Memory Access Protocol - Encoding.

This section specifies the instruction encoding for Memory Access Protocol instructions in 4-255 Data Bus interrogation words and the encoding of the corresponding response words.

The Memory Access Protocol uses a standard 4-255 Data Bus interrogation word as shown in Figure 43.

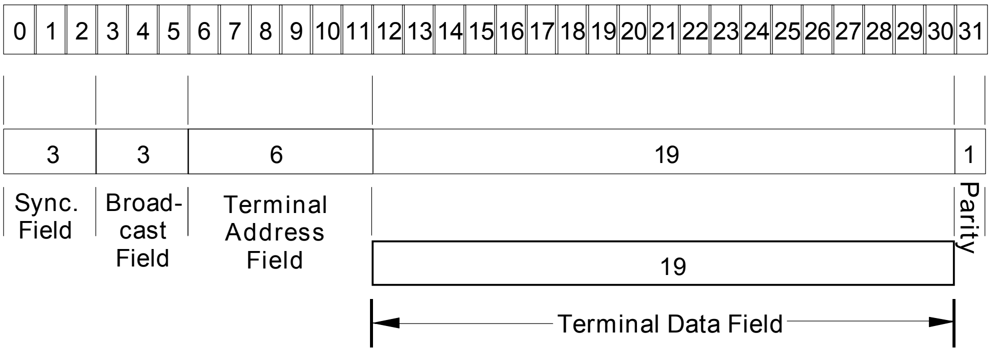

Figure 43 - Memory Access Protocol Instruction Word Format

12.5.1. Memory Access Protocol - Interrogation Word Format.

256-PE7-52-D <u>MAP Instruction Encoding</u>

For MAP instructions, the terminal data field of the 4-255 interrogation word shall be encoded as shown in Table 16.

Interrogation Word Bit Number																			Instruction
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Read Memory Page Selector, READ_MPS
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	Read Memory Offset Pointer, READ_MOP
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	Semaphore-up, SEM_UP
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	Semaphore-down, SEM_DOWN
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	Memory Offset Pointer Mode, MOP_MODE - (Fixed Address)
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	Memory Offset Pointer Mode, MOP_MODE - (Auto-increment)
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	Memory Offset Pointer Mode, MOP_MODE - (Auto-decrement)
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	Memory Offset Pointer Mode, MOP_MODE - (Enquire)
0	0	1	1	1	1	1	<Any value more than 111>												Reserved
0	1	0	<Address>																Load Memory Offset Pointer, LOAD_MOP
1	0	0	<Data>																Memory Write, MEMWRI
1	0	1	1	<Memory page selector value>															Load Memory Page Selector, LOAD_MPS
1	1	0	<Address>																Memory Read, MEMRD
1	1	1	x	x	X	x	x	x	x	x	x	x	x	x	x	x	x	x	Reserved

Table 16 - Memory Access Protocol Instruction Set Encoding.

12.5.2. Memory Access Protocol - Response Word Format.

The response word format used by the Memory Access Protocol complies with the 4-255 Data Bus 21-bit response word format shown in Figure 44.

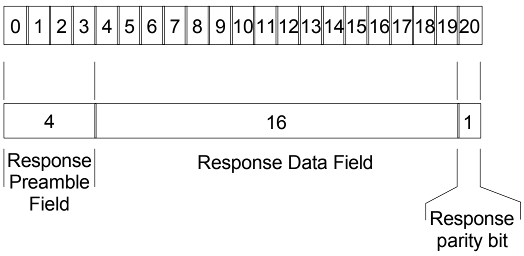

Figure 44 - Memory Access Protocol Response Word Format.

256-PE7-53-D MAP Response - Preamble Field

The response preamble field occupies bits 0, 1, 2 and 3 of the response word. These four bits shall be used to indicate data quality, to report errors, and to request Bus Controller attention.

The response preamble field for Protocol Extension 7 is shown in Figure 45.

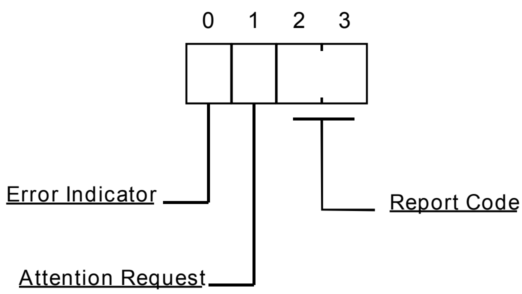

Figure 45 - Memory Access Protocol - Response Preamble Field.

256-PE7-54-D MAP Response Preamble Field - Error Indicator

The response preamble field error indicator bit (bit 0) shall be set to one (1) to indicate that an error occurred during the execution of the instruction or that the instruction was rejected. This bit shall be set to zero (0) to indicate that the instruction was not rejected and that no errors were detected during its execution.

256-PE7-55-D MAP Response Preamble Field - Attention Request

The response preamble field attention request bit (bit 1) is reserved and shall be set to zero (0).

Note: This bit is controlled within the Access Control Sub-layer.

256-PE7-56-D MAP Response Preamble Field - Report Code, No Errors

If no errors are detected during the execution of a MAP instruction, the response preamble field report code field (bits 2 and 3) is reserved and by default shall be set to zero (00).

256-PE7-57-O MAP Response Preamble Field - Report Code, Errors

Optionally, if an error is detected during the execution of a MAP instruction, the MAP entity shall set the response preamble field report code to indicate the type of error as defined in Table 17.

Note: If an error occurs a report code value of zero can be used to indicate that an error code is contained in the data field.

Response Preamble Field				
Error Indicator	Attention Request	Report Code		Meaning
0	X	X	X	Reserved
1	X	0	0	Unspecified Error
1	X	0	1	Instruction Rejected
1	X	1	0	Reserved
1	X	1	1	Reserved

Table 17 - Memory Access Protocol - Response Preamble Field Report Codes.

256-PE7-58-D MAP Response Word - Data Field, No Errors

If no errors are detected during the execution of a MAP instruction, the response data field (bits 4 to 19 inclusive) shall be used to convey the 16-bits of data which are required in response to that instruction.

256-PE7-59-O MAP Response Word - Data Field, Errors

Optionally if errors are detected during the execution of a MAP instruction and the response preamble field error indicator has been set to 1, the 16-bit response data field (bits 4 to 19 inclusive) shall be used to carry a user defined error code.

256-PE7-60-D MAP Response Word - Parity Bit

The parity bit of the response word (bit 20) shall contain either:

- A parity value covering the preceding 16-bits of the response word, i.e. the response data field,
- An additional (seventeenth) data bit.

256-PE7-61-D Response Parity Bit Value - Response Parity Mode = Even

When the response parity mode is ‘Even’, the parity bit shall be set so that the sum of the parity bit plus all ‘one’ bits in the preceding 16-bits of the response word is even.

Note: When the response parity mode is ‘None’, the parity bit may be set to one or zero. This allows it to be used as a seventeenth data bit.

12.5.2.1. Memory Access Protocol MOP_MODE Response Encoding.

256-PE7-62-D MAP - MOP_MODE Response Encoding

The response data field of a MAP MOP_MODE response shall be encoded as shown in Figure 46.

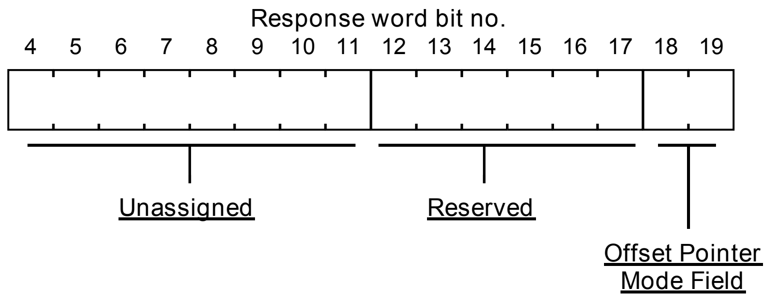

Figure 46 - Memory Access Protocol MOP_MODE Response Format.

256-PE7-63-D MOP_MODE Response - Response Fields

The MAP MOP_MODE response shall be divided into three components as follows:

- An 8-bit unassigned field,
- A 6-bit reserved field,
- A 2-bit memory offset pointer mode field.

256-PE7-64-D MOP_MODE Response - Unassigned Field

Bits 4 to 11 inclusive of the MAP MOP_MODE response are unassigned and may be assigned for user specific purposes. Any unused bits shall be set to zero (0).

256-PE7-65-D MOP_MODE Response – Reserved Field

Bits 12 to 17 inclusive of the MAP MOP_MODE response are reserved and shall all be set to zero (0).

256-PE7-66-D MOP_MODE Response - Pointer Address Mode Field

The MAP MOP_MODE response memory offset pointer mode field (bits 18 and 19) shall indicate the current memory offset pointer mode. The encoding of this field shall be as shown in Table 18.

Transfer Bus Selection Field Code		Pointer Address Mode
0	0	Fixed Address
0	1	Auto-increment
1	0	Auto-decrement
1	1	Reserved

Table 18 - MOP_MODE Response - Pointer Address Field Codes.

Appendix 1.Recommended Assignment of Protocol Extension 1 Instructions to Electrical User Interfaces.

This appendix contains recommendations for the assignment of Protocol Extension 1 instruction codes to the Electrical User Interfaces defined in PSS-04-253, Specification for the Electrical User Interfaces for use in 4-201 Data Handling Systems.

256-PE1-29-R Data Acquisition and Pulse Instructions - Channel Grouping

All of the channels accessed by a specific instruction code value shall use the same electrical user interface type.

256-PE1-30-R Data Acquisition and Pulse Instructions – Assignment to Electrical User Interfaces

The data acquisition and pulse instruction codes shall be assigned to the electrical user interfaces as shown in Table 19.

Digital Bi-Level Data Acquisition

Instruction Type Identifier	Instruction Code			Remote Terminal Operation
Bit 19	Bit 20	Bit 21	Bit 22	
0	0			
1				

Table 19 - Data Acquisition and Pulse Instruction Code Assignments to Electrical User Interfaces.

Appendix 1.Techniques for Extending the Addressing Range of Register Load Instructions.

This section describes three techniques which can be used to extend the addressing range of register load instructions beyond seven, namely:

- Using more than one terminal address at a given Remote Terminal,
- Using indirection techniques to access additional registers,
- Using two interrogations to carry a memory load instruction.

These are not the only techniques which can be used, but represent three distinct strategies. Other techniques are variations on the methods described here.

The address range extension of the register load instructions will typically provide access to tens of registers. Remote Terminals which must provide access to more registers or blocks of memory, should use Protocol Extension 4, the RBI protocol, or Protocol Extension 7, the Memory Access Protocol.

A2.1. Use of Multiple Terminal Addresses.

The use of multiple terminal addresses on a single Remote Terminal increases the number of directly addressable registers by seven for each terminal address used. The method extends the register address space by combining the register load address with the terminal address field to identify the target register.

This technique can be applied only if there are a sufficient number of terminal addresses available. However, a significant advantage of the method is its simplicity. All of the available registers are directly accessible using single interrogation words.

256-PE2-18-R Register Load Instruction - Multiple Terminal Addresses

Where the register load instruction address range is extended by using multiple terminal addresses, adjacent addresses shall be used within each physical terminal.

A2.2. Use of Indirection Techniques.

Indirection extends the address range by using one of the directly accessible registers as a pointer to other registers. For example, a single 16-bit register could be configured as a pointer to sixteen banks of six 16-bit registers as shown in Figure 47. This extends the addressing range to 96 registers even without using decoding on the pointer register output. Once the pointer register is configured, each of the registers in the bank becomes directly addressable. Therefore, this method can still offer reasonable efficiency if registers are commonly accessed in groups.

Indirection has the advantages of being very flexible, and offering large increases in addressing range for minimal addition of hardware in the Remote Terminal. It also uses only one terminal address.

Against this, indirection requires the use of more complicated procedures by the Bus Controller in order to configure and maintain the pointer register. Furthermore, registers will not always be directly accessible from the Bus Controller, they may require two interrogations to access them. This may make it necessary to impose special operating requirements to ensure that a consistent and reliable means of access is achieved.

The following recommendations apply when indirection is being used:

256-PE2-19-R Register Load Instruction - Indirection Register

Where the register load instruction address range is extended by indirection techniques, the indirection register shall be located at register load address 1 (001 binary).

256-PE2-20-R Register Load Instruction - Indirection Bank Addresses

Where the register load instruction address range is extended by indirection techniques, the bank address value loaded into the indirection register shall be a binary value which identifies only one bank of registers.

256-PE2-21-R Register Load Instruction - Indirection Bank Size

Where the register load instruction address range is extended by indirection techniques, each indirectly addressed register bank shall contain up to six registers.

Note 1: It may be necessary to extend the register load instruction rejection criteria, for example to reject all subsequent register load instructions to register load addresses other than 001 if a pointer register load fails. Local error flags may also need to be raised.

Note 2: Rejection of register load instructions to non-implemented registers must take into account the current setting of the pointer register.

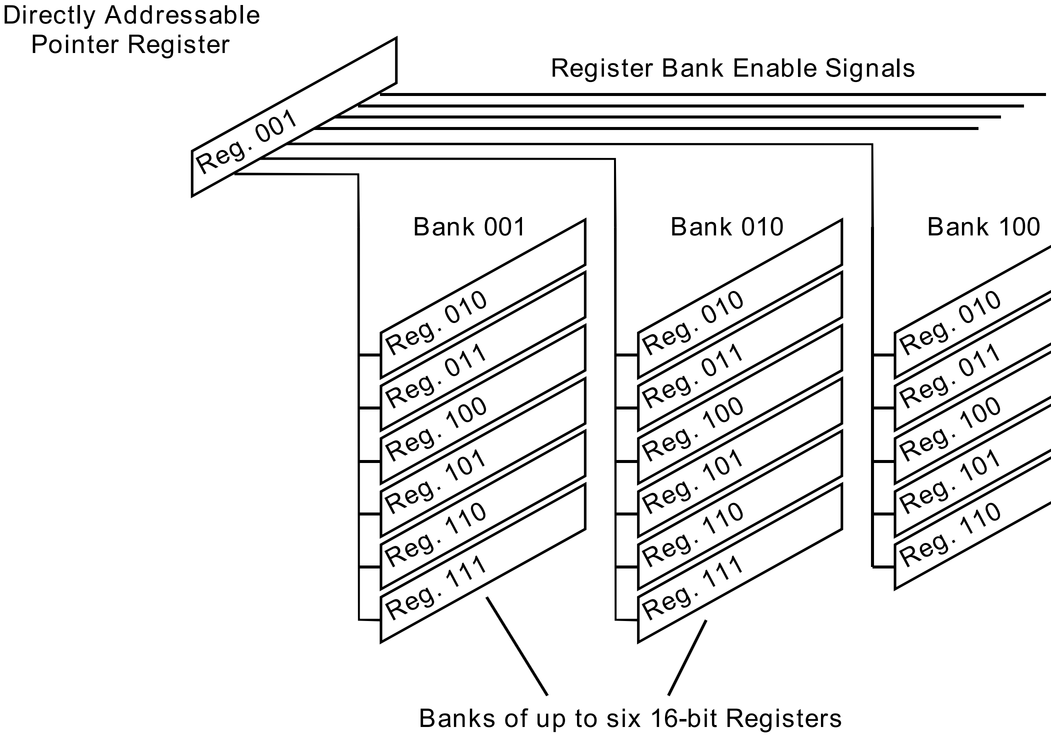

Figure 47 - Example of Extending Addressing Range Using a Pointer.

A2.3. Use of Two Interrogations per Instruction.

The third technique uses two register load instructions in succession. The first instruction carries a 16-bit address, the second instruction carries the 16-bit data to be written to that address. This can be considered an extension of the indirection technique described above but, instead of the pointer indicating a bank of registers, the pointer is now fully decoded to indicate one 16-bit register.

> Note: The use of multiple interrogations per instruction can be used in other cases, e.g. in Protocol Extension 1 to increase the number of data acquisition channels which can be identified. However, this technique may introduce complications due the identification of interrogations which make up an instruction and ensuring correct handling of instructions in which not all of the interrogations are received.

Within a layered architecture such as the 4-201 Data Handling System, multiple instructions sent from the Bus Controller cannot be guaranteed to be placed in adjacent 4-255 Data Bus slots. Users should therefore not rely on a fixed slot relationship, such as adjacency, to associate instructions carried in multiple interrogations. The preferred approach for all 4-255 Data Bus protocol extensions is to use a single instruction in each interrogation or to make each interrogation of a multiple interrogation instruction uniquely identifiable.

It is therefore recommended that the first register load instruction (containing the address) should use register load address 001, and the second register load instruction (containing the data) should use register load address 010. This allows the two instructions to be clearly distinguished and obviates the need for time-outs and special rejection criteria in the Remote Terminal. Using this scheme, register load instructions to register load address 001 always write to the pointer while register load instructions to register load 010 always write to the register referenced by the pointer. Single register load instructions to other register load addresses can be treated as normal register load instructions.

The following recommendations apply when two instructions are used to perform a single register load operation.

256-PE2-22-R Two Instruction Register Load Instruction – Address Load Instruction

The address load instruction shall be a register load instruction with the register load address 001 and the register load data set to the address of the register to be loaded.

256-PE2-23-R Two Instruction Register Load Instruction – Data Load Instruction

The data load instruction shall be a register load instruction with the register load address 010 and the register load data set to the value of the data to be loaded into the register referenced by the address pointer.