INTERNATIONAL STANDARD

ISO 8571-4

First edition
1988-10-01

AMENDMENT 2
1993-08-15

Information processing systems – Open Systems Interconnection – File Transfer, Access and Management –

Part 4 :
File Protocol Specification

AMENDMENT 2 : Overlapped access

Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Transfert, accès et gestion de fichiers –

Partie 4 : Spécification du protocole de fichiers

AMENDEMENT 2 : Chevauchement d'accès

Reference number
ISO 8571-4:1988/Amd.2:1993 (E)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 2 to International Standard ISO 8571-4:1988 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

ISO 8571 consists of the following parts, under the general title *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management*:

- *Part 1 : General introduction*
- *Part 2 : Virtual Filestore Definition*
- *Part 3 : File Service Definition*
- *Part 4 : File Protocol Specification*
- *Part 5 : Protocol Implementation Conformance Statement Proforma*

Printed in Switzerland

Information processing systems – Open Systems Interconnection – File Transfer, Access and Management –

Part 4:
File Protocol Specification

AMENDMENT 2 : Overlapped access

0. Introduction

Clause 0 provides an introduction to this amendment. The text in this clause is not intended for inclusion in ISO 8571 part 4.

0.1 General

ISO 8571 part 4 provides specifications of the protocols that support the internal and external file service interfaces.

This amendment extends these protocol specifications to provide support for the services offered by overlapped access.

0.2 Rationale

The objective in introducing overlapped access is to allow more efficient access to structured files when a single initiator has a need to perform many reading and updating operations; the serial nature of the current FTAM data transfer services introduces a significant control overhead if the FADUs are small. In this context, an FADU is small if its transmission time is comparable with the time to complete a confirmed service on the association (the association's round trip delay).

0.3 Summary

The current design envelope that there should be at most one file selection per association and one file open per file selection is maintained. If access to more than one file is to be overlapped, more than one association is necessary, The overlapped access takes place within a constant set of presentation contexts established as at present when the file is opened, or previously.

Two different degrees of overlap have been identified. Firstly, requests for future accesses may be issued whilst a previously requested BDT action is in progress, allowing the creation of a queue of read and write requests. In general, PCI relating to a given BDT action may be overlapped with other BDT actions, subject to restrictions; this is called consecutive access. Secondly, read and write actions can be performed in parallel, so that both directions of data transfer are exploited at any one time. Requests are then taken from the queue whenever either direction of transfer becomes free. This is called concurrent access.

The transfer of a single FADU, specified in a single F-READ request has the same interpretation as in ISO 8571. The resultant effect on the virtual filestore of a set of overlapped requests using consecutive access shall be the same as that of the equivalent set of requests issued in series; the service provided is serializable. If concurrent access is used then the resultant effect of a set of write actions on the virtual filestore, is also serializable. However, due to the non-determinism introduced by the use of concurrent access, it is also possible that in some uses of the service, the data transferred as a result of a read action is not consistent with the current state of the file.

1. Scope

This amendment makes no additions to clause 1.

2. Field of application

This amendment makes no additions to clause 2.

3. References

This amendment makes no additions to clause 3.

4. Definitions and abbreviations

This amendment makes no additions to clause 4.

Section one: General

5. Overview of the protocol

5.2 Services assumed by the basic file protocol

Amend table 1.

5.6 Protocol functional units

Add to end of list.

k) consecutive access functional unit

l) concurrent access functional unit

FTAM Functional Unit	Session Functional Unit	Presentation Functional Unit
Kernel(4)	Kernel Duplex Optionally: Resynch(1) Minor synch(2)	Kernel Duplex Optionally: Resynch(1) Minor synch(2) Context Management(3)
Recovery	Minor Synch Symmetric Synchronize (5)	Minor Synch Symmetric Synchronize (5)
Restart	Minor Synch Resynchronize Symmetric Synchronize (5)	Minor Synch Resynchronize Symmetric Synchronize (5)

NOTES

5 The Symmetric Synchronize functional unit is used when overlapped access is in use. Pending the specification of presentation symmetric synchronisation services, recovery mechanisms will not be available for use during overlapped access.

Section two: Basic file protocol

6. State of the association provided

6.2 Additional state information

6.2.3 Next state indicator

Replace second sentence.

The defined values are the state names "initialised", "selected", "normal data transfer idle", "consecutive data transfer idle", "concurrent data transfer idle", and "unset".

6.2.5

Add as last paragraph:

In overlapped access, the bulk transfer number identifies the bulk data transfer that can be cancelled, restarted or recovered. Thus, the bulk transfer number is incremented only when there is an outstanding data transfer request and the previous data transfer cannot be cancelled, restarted or recovered. If an initiator and a responder have different bulk transfer numbers then it is the bulk data transfer associated with the smaller of the two numbers that is cancelled, restarted or recovered.

6.2.6 Transfer number

In overlapped access, the transfer number identifies the bulk data transfer within a sequence of transfers from one data transfer idle state to a next data transfer idle state within an open regime. It is set to zero at each data transfer idle state.

In concurrent overlapped access, two transfer numbers are maintained - one for reads and one for writes.

7. File protocol data units

This amendment makes no additions to clause 7.

8. File initiating entity actions

9. File responding entity actions

10. File general actions

This amendment makes no additions to clause 10.

Section three: The basic bulk data transfer protocol

11. State of bulk data transfer activity

Re-label sub-clause 11.2.

11.2 Additional state of the entities (without overlapped access)

Replace first paragraph.

The following sub-clauses define the items of state information associated with the basic protocol entities for the purposes of bulk data transfer without overlapped access.

Add the following sub-clauses.

11.3 Additional state of the entities (with consecutive access)

The following sub-clauses define the items of state information associated with the basic protocol entities for the purposes of bulk data transfer with consecutive access.

11.3.1 Current Transfer Number

The current transfer number indicates the transfer number of the bulk data transfer procedure that is currently considered "in progress".

11.3.2 Start Bulk Transfer Number

The start bulk transfer number indicates the bulk transfer number of the first in a sequence of overlapped data transfers. The start bulk transfer number is used to calculate the bulk transfer number and transfer number for cancel, recover and restart.

11.3.3 Checkpoint identifier expected

The checkpoint identifier expected reflects the sequence of checkpoints within bulk data, and is reset by the start of the bulk data transfer, and by error recovery mechanisms. The value is incremented when a checkpoint is made. The value is an integer in the range 1 to 999998. Initially, the value is determined by the state of the association.

The checkpoint identifier expected applies only to FTAM regimes for which the use of the restart and/or recovery functional units have been successfully negotiated.

11.3.4 First next synchronisation point number

The first next synchronisation point number reflects the sequence of events in the supporting synchronisation services for a sender. The number is the serial number of the next session synchronisation point to be issued by the session service provider. The value is an integer in the range 0 to 999998. Initially on a newly created session connection, the value is 1.

The first next synchronisation point number applies only to FTAM regimes for which the use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

11.3.5 Second next synchronisation point number

The second next synchronisation point number reflects the sequence of events in the supporting synchronisation services for a receiver. The number is the serial number of the next session synchronisation point to be issued by the session service provider. The value is an integer in the range 0 to 999998. Initially on a newly created session connection, the value is 1.

The second next synchronisation point number applies only to FTAM regimes for which the use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

Table 5 - Protocol Data Units

Name	Carried by	Functional units
F-CHECK request (see note 2)	P-SYNC-MINOR request	recovery, restart
F-CHECK response (see note 2)	P-SYNC-MINOR response	recovery, restart

NOTES

1 The data value corresponds to an F-DATA request service primitive. There is no F-DATA request PDU as such.

2 If overlapped access is not in use then the F-CHECK request and response primitives are mapped directly onto the P-SYNC-MINOR request and response primitives, with no additional syntax.

11.3.6 Synchronisation offset

The synchronisation offset is a constant established when a read or write bulk data transfer is initiated or recovered, which gives the difference between the checkpoint identifier expected and the next resynchronisation point number.

The synchronisation offset applies only to FTAM regimes for which use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

NOTE - Except during the issue of a checkpoint, or performance of the restart procedure the value of the synchronisation offset is equal to the difference between the expected checkpoint number and the next synchronisation point number.

11.3.7 Outstanding Checkpoint counter

The outstanding checkpoint counter records the number of checkpoints which are unacknowledged.

11.3.8 Read/Write Indicator

The read/write indicator records whether the current bulk data transfer is to or from the initiator. The value is set upon the beginning of the data transfer regime. The defined values are "reading", "writing" and "unset". The initial value is unset.

11.3.9 Discard Indicator

The discard indicator is used to signal that data received during cancellation or before recovery is invalid and should be thrown away. If the recovery or restart functional units are selected and presentation resynchronisation functional unit has been successfully negotiated, it is used in conjunction with session resynchronisation to produce a recovery without user visibility of the error; otherwise it is used during the cancel phase. The defined values are "unset" and "set". Initially, the value is "unset".

11.3.10 Transfer Request Queue

The transfer request queue records the transfer number and transfer type (read or write) of all outstanding data transfers.

11.3.11 Transfer End Queue

The transfer end queue records the transfer numbers of all data transfers, already recorded on the transfer request queue, for which a transfer end PDU request or response has been issued.

11.3.12 Data End Queue

The data end queue records the transfer numbers of all data transfers, already recorded on the data request queue, for which a data end PDU has been issued or received.

11.3.13 Read and write checkpoint tables

The read and write checkpoint tables (two separate tables are kept) record checkpointing information for each ongoing bulk data transfer. The following information is kept for each currently active bulk data transfer: transfer number, bulk transfer number, checkpoint expected, synchronisation offset, and checkpoint counter. The size of the table is that of the transfer window negotiated when the file is opened.

11.3.14 Last transfer end confirm indicator

The last transfer end confirm indicator records the last transfer end received by the initiator. The last transfer end confirm received is sent to the responder on a transfer end request PDU. It is also included on cancel, recover and restart PDUs. The responder uses the information in removing items from the transfer end response queue and in the re-issuing of transfer-end response after a session resynchronisation in the direction of responder to initiator.

11.3.15 Last transfer end indication indicator

The last transfer end indication indicator records the last transfer end indication received by the responder. It is included on cancel, recover and restart PDUs. The initiator uses the information in the re-issuing of transfer end requests after a session resynchronisation in the direction of initiator to responder.

11.4 Additional state of the entities (with concurrent access)

The following sub-clauses define the items of state information associated with the basic protocol entities for the purposes of bulk data transfer with concurrent access.

11.4.1 Current Read Transfer Number and Current Write Transfer Number

The current transfer read number and the current transfer write number indicate the transfer numbers of the read and write data transfer procedures that are in progress.

11.4.2 Start Bulk Transfer Number

The start bulk transfer number indicates the bulk transfer number of the first in a sequence of overlapped data transfers. The start bulk transfer number is used to calculate the bulk transfer number and transfer number for cancel, recover, and restart.

11.4.3 Checkpoint identifier expected

The checkpoint identifier expected reflects the sequence of checkpoints within bulk data, and is reset by the start of the bulk data transfer, and by error recovery mechanisms. The value is incremented when a checkpoint is made. The value

is an integer in the range 1 to 999998. Initially, the value is determined by the state of the association.

The checkpoint identifier expected applies only to FTAM regimes for which the use of the restart and/or recovery functional units have been successfully negotiated.

11.4.4 First next synchronisation point number

The first next synchronisation point number reflects the sequence of events in the supporting synchronisation services for a sender. The number is the serial number of the next session synchronisation point to be issued by the session service provider. The value is an integer in the range 0 to 999998. Initially on a newly created session connection, the value is 1.

The first next synchronisation point number applies only to FTAM regimes for which the use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

11.4.5 Second next synchronisation point number

The second next synchronisation point number reflects the sequence of events in the supporting synchronisation services for a receiver. The number is the serial number of the next session synchronisation point to be issued by the session service provider. The value is an integer in the range 0 to 999998. Initially on a newly created session connection, the value is 1.

The second next synchronisation point number applies only to FTAM regimes for which the use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

11.4.6 Synchronisation offset

The synchronisation offset is a constant established when a read or write bulk data transfer is initiated or recovered, which gives the difference between the checkpoint identifier expected and the next resynchronisation point number.

The synchronisation offset applies only to FTAM regimes for which use of the presentation symmetric synchronisation functional unit has been successfully negotiated.

NOTE - Except during the issue of a checkpoint, or performance of the restart procedure the value of the synchronisation offset is equal to the difference between the expected checkpoint number and the next synchronisation point number.

11.4.7 Outstanding Checkpoint counter

The outstanding checkpoint counter records the number of checkpoints which are unacknowledged.

11.4.8 Read Indicator and Write Indicator

The read indicator and the write indicator record whether or not there are read or write data transfers in progress. The defined values are "reading"/"writing" and "unset". The initial value is unset.

11.4.9 Discard indicator

The discard indicator is used to signal that data received during cancellation or before recovery is invalid and should be thrown away. If the recovery or restart functional units are selected and presentation resynchronisation functional unit has been successfully negotiated, it is used in conjunction with session resynchronisation to produce a recovery without user visibility of the error; otherwise it is used during the cancel phase. The defined values are "unset" and "set". Initially, the value is "unset".

11.4.10 Transfer Read Request Queue and Transfer Write Request Queue

The transfer read request queue and the transfer write request queue record the transfer numbers of all outstanding read and write data transfers.

11.4.11 Transfer End Read Queue and Transfer End Write Queue

The transfer end read queue and the transfer end write queue record the transfer numbers of all read and write data transfers, already recorded on the transfer request read (write) queue, for which a transfer end PDU request or response has been issued.

11.4.12 Data End Read Queue and Data End Write Queue

The data end read queue and the data end write queue record the transfer numbers of all read and write data transfers, all ready recorded on the transfer request read (write) queue, for which a data end PDU has been issued/ received.

11.4.13 Read and write checkpoint tables

The read and write checkpoint tables (two separate tables are kept) record checkpointing information for each ongoing bulk data transfer. The following information is kept for each currently active bulk data transfer: transfer number, bulk transfer number, checkpoint expected, synchronisation offset, and checkpoint counter. The size of the table is that of the transfer window negotiated when the file is opened.

11.4.14 Last transfer end confirm indicator

The last transfer end confirm indicator records the last

transfer end read (write) received by the initiator. The last transfer end confirm received is sent to the responder on a transfer end request PDU. It is also included on cancel, recover and restart PDUs. The responder uses the information in removing items from the transfer end response queue and in the re-issuing of transfer-end response after a session resynchronisation in the direction of responder to initiator.

11.4.15 Last transfer end indication indicator

The last transfer end indication indicator records the last transfer end read (write) indication received by the responder. It is included on cancel, recover and restart PDUs. The initiator uses the information in the re-issuing of transfer end requests after a session resynchronisation in the direction of initiator to responder.

12. Bulk data transfer protocol data units

Amend table 5.

13. Bulk data transfer initiating entity actions

14. Bulk data transfer responding entity actions

15. Bulk data transfer sending entity actions

16. Bulk data transfer receiving entity actions

17. Bulk data transfer general actions

Section four: The error recovery protocol

18. Protocol mechanisms

19. Specification of the error control protocol

Section five: Abstract Syntax

20. Abstract Syntax Definition

20.3 ASN.1 Module Definition

Amend Figure 7.

Amend Figure 8.

Amend Figure 9.

Amend Figure 10.

21. Application Context Name

This amendment makes no additions to clause 21.

22. Conformance

This amendment makes no additions to clause 22.

```
ISO8571-FTAM  DEFINITIONS  ::=

BEGIN

   Functional-Units  ::= [4]  IMPLICIT  BIT  STRING
                  {  read                                ( 2),
                     write                               ( 3),
                     file-access                         ( 4),
                      limited-file-management            ( 5),
                      enhanced-file-management           ( 6),
                     grouping                            ( 7),
                     fadu-locking                        ( 8),
                     recovery                            ( 9),
                      restart-data-transfer              (10),
                      limited-filestore-management       (11),
                      enhanced-filestore-management      (12),
                     object-manipulation                 (13),
                     group-manipulation                  (14),
                     consecutive-access                  (15),
                     concurrent-access                   (16)  }

END
```

Figure 7 - FTAM regime PDUs

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

  F-OPEN-request ::= SEQUENCE {
     processing-mode           [0] IMPLICIT BIT STRING
                                   { f-read       (0),
                                     f-insert          (1),
                                     f-replace         (2),
                                     f-extend          (3),
                                     f-erase           (4) } DEFAULT { f-read },
     contents-type             [1] CHOICE {
                 unknown       [0] IMPLICIT NULL,
                 proposed      [1] Contents-Type-Attribute },
     concurrency-control             Concurrency-Control OPTIONAL,
     shared-ASE-information     Shared-ASE-Information OPTIONAL,
     enable-fadu-locking       [2] IMPLICIT BOOLEAN DEFAULT FALSE,
     activity-identifier        Activity-Identifier OPTIONAL,
           - Only used in the recovery functional unit.
     recovery-mode             [3] IMPLICIT INTEGER
                 { none (0),
                   at-start-of-file (1),
                   at-any-active-checkpoint (2) } DEFAULT none,
     remove-contexts           [4] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
     define-contexts           [5] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
           - The following are conditional on the negotiation of the consecutive overlap or
           - concurrent overlap functional units.
     degree-of-overlap         Degree-Of-Overlap OPTIONAL,
     transfer-window           [7] IMPLICIT INTEGER OPTIONAL }

  F-OPEN-response ::= SEQUENCE {
     state-result        State-Result DEFAULT success,
     action-result             Action-Result DEFAULT success,
     contents-type             [1] Contents-Type-Attribute,
     concurrency-control       Concurrency-Control OPTIONAL,
     shared-ASE-information     Shared-ASE-Information OPTIONAL,
     diagnostic                Diagnostic OPTIONAL,
     recovery-mode                   [3] IMPLICIT INTEGER
                 { none (0),
                   at-start-of-file (1),
                   at-any-active-checkpoint (2) } DEFAULT none,
     presentation-action       [6] IMPLICIT BOOLEAN DEFAULT FALSE,
           - This flag is set if the responder is going to follow this response by a P-ALTER-CONTEXT
           -  exchange.
           - The following are conditional on the negotiation of the consecutive access or
           - concurrent access functional units.
     degree-of-overlap         Degree-Of-Overlap OPTIONAL,
     transfer-window           [7] IMPLICIT INTEGER OPTIONAL }
```

Figure 8 - File selection and file open regime PDUs

```
F-RECOVER-request ::= SEQUENCE {
    activity-identifier                 Activity-Identifier,
    bulk-transfer-number                [0] IMPLICIT INTEGER,
          - If concurrent access was in use then this parameter indicates the read bulk
          -  transfer.
    requested-access              Access-Request,
    access-passwords              Access-Passwords OPTIONAL,
    recovery-point                      [2] IMPLICIT INTEGER DEFAULT 0,
          - zero indicates beginning of file point after last checkpoint indicates end of file
    remove-contexts               [3] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
    define-contexts               [4] IMPLICIT SET OF Abstract-Syntax-Name OPTIONAL,
          - The following are conditional on the negotiation of overlapped access.
    concurrent-bulk-transfer-number     [7] IMPLICIT INTEGER OPTIONAL,
          - conditional on use of concurrent access
    concurrent-recovery-point     [8] IMPLICIT INTEGER OPTIONAL,
          - conditional on use of concurrent access. Zero indicates beginning of file
          - point after last checkpoint indicates end of file
    last-transfer-end-read-response [9] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-response    [10] IMPLICIT INTEGER OPTIONAL }

F-RECOVER-response ::= SEQUENCE {
    state-result       State-Result Default success,
    action-result            Action-Result DEFAULT success,
    contents-type            [1] Contents-Type-Attribute,
    recovery-point           [2] IMPLICIT INTEGER DEFAULT 0,
          - Zero indicates beginning of file; point after last checkpoint indicates end of file
    diagnostic         Diagnostic OPTIONAL,
    presentation-action      [6] IMPLICIT BOOLEAN DEFAULT FALSE,
          - This flag is set if the responder is going to follow this response
          - by a P-ALTER-CONTEXT exchange.
          - The following are conditional on the negotiation of overlapped access.
    concurrent-recovery-point     [8] IMPLICIT INTEGER OPTIONAL,
          - conditional on use of concurrent access. Zero indicates beginning of file; point after
          - last checkpoint indicates end of file
    last-transfer-end-read-request  [9] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-request [10] IMPLICIT INTEGER OPTIONAL }

END
```

Figure 8 (continued) - File selection and file open regime PDUs

Amend Figure 9.

```
ISO8571-FTAM DEFINITIONS ::

BEGIN

   F-READ-request ::= SEQUENCE {
      file-access-data-unit-identity  FADU-Identity,
      access-context                        Access-Context,
      fadu-lock                             FADU-Lock OPTIONAL,
            - The following is conditional on the negotiation of consecutive of concurrent access.
      transfer-number                       [0] IMPLICIT INTEGER OPTIONAL }

   F-WRITE-request ::= SEQUENCE {
      file-access-data-unit-operation [0] IMPLICIT INTEGER
                        { insert     (0),
                          replace    (1),
                          extend     (2) },
      file-access-data-unit-identity  FADU-Identity,
      fadu-lock                             FADU-Lock OPTIONAL,
            - The following is conditional on the negotiation of consecutive or concurrent access.
      transfer-number                       [1] IMPLICIT INTEGER OPTIONAL }

   F-TRANSFER-END-request ::= SEQUENCE {
      shared-ASE-information                Shared-ASE-Information OPTIONAL,
            - The following are conditional on the negotiation of consecutive or concurrent access.
      request-type                          Request-Type OPTIONAL,
      transfer-number                       [0] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-read-response [1] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-write-response      [2] IMPLICIT INTEGER OPTIONAL }

   F-TRANSFER-END-response ::= SEQUENCE {
      action-result            Action-Result DEFAULT success,
      shared-ASE-information   Shared-ASE-Information OPTIONAL,
      diagnostic               Diagnostic OPTIONAL,
            - The following are conditional on the negotiation of consecutive or concurrent access.
      request-type             Request-Type OPTIONAL,
      transfer-number          [0] IMPLICIT INTEGER OPTIONAL }

   F-CANCEL-request ::= SEQUENCE {
      action-result                         Action-Result DEFAULT success,
      shared-ASE-information                Shared-ASE-Information OPTIONAL,
      diagnostic                            Diagnostic OPTIONAL,
            - The following are conditional on the negotiation of consecutive or concurrent access.
      request-type                    Request-Type,
      transfer-number                       [0] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-read-request  [1] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-read-response [2] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-write-request [3] IMPLICIT INTEGER OPTIONAL,
      last-transfer-end-write-response      [4] IMPLICIT INTEGER OPTIONAL }

   F-CANCEL-response ::= SEQUENCE {
      action-result                         Action-Result DEFAULT success,
      shared-ASE-information                Shared-ASE-Information OPTIONAL,
```

Figure 9 - Bulk data transfer PDUs

```
    diagnostic                          Diagnostic OPTIONAL,
          -- The following are conditional on the negotiation of consecutive or concurrent access.
    request-type                  Request-Type OPTIONAL,
    transfer-number                     [0] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-read-request  [1] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-read-response [2] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-request [3] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-response    [4] IMPLICIT INTEGER OPTIONAL }

F-CHECK-request ::= SEQUENCE {
    checkpoint-identifier    [0] IMPLICIT INTEGER,
    transfer-number    [1] IMPLICIT INTEGER }

F-CHECK-response ::= SEQUENCE {
    checkpoint-identifier    [0] IMPLICIT INTEGER,
    transfer-number    [1] IMPLICIT INTEGER }

F-RESTART-request ::= SEQUENCE {
    checkpoint-identifier               [0] IMPLICIT INTEGER,
          -- The following are conditional on the negotiation of consecutive or concurrent access.
    request-type                  Request-Type OPTIONAL,
    transfer-number               [1] IMPLICIT INTEGER,
    last-transfer-end-read-request  [2] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-read-response [3] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-request [4] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-response    [5] IMPLICIT INTEGER OPTIONAL }

F-RESTART-response ::= SEQUENCE {
    checkpoint-identifier               [0] IMPLICIT INTEGER,
          -- The following are conditional on the negotiation of consecutive or concurrent access.
    request-type                  Request-Type OPTIONAL,
    transfer-number               [1] IMPLICIT INTEGER,
    last-transfer-end-read-request  [2] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-read-response [3] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-request [4] IMPLICIT INTEGER OPTIONAL,
    last-transfer-end-write-response    [5] IMPLICIT INTEGER OPTIONAL }

END
```

Figure 9 (continued) - Bulk data transfer PDUs

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

   Degree-Of-Overlap ::= [Application 30] IMPLICIT INTEGER
            { normal      (0),
              consecutive (1),
              concurrent  (2) }

   Request-Type ::= [Application 31] IMPLICIT INTEGER
            { read (0),
              write      (1) }

END
```

Figure 10 - Application wide types

Section six: Conformance

Annex A
Protocol state tables
(This annex forms part of the standard.)

A.1 Introduction

Add note after third paragraph:

Note - for concurrent access, there are separate state tables for read and write procedures, for both the FERPM and the Basic FPM.

A.1.1 System model

Add last paragraph.

In addition, for concurrent overlap, both the FPM and the FERPM are considered to consist of two sub-entities of read FPM and write FPM, and read FERPM and write FERPM. The additional local signals for concurrent overlap are:

q) L-TRERQ-R - signal indicating a re-issued F-TRANSFER-END(read) request primitive;

r) L-TRERQ-W -signal indicating a re-issued F-TRANSFER-END(write) request primitive;

s) L-TRERP-W - signal indicating a re-issued F-TRANSFER-END(write) response primitive;

t) L-TRERP-R - signal indicating a re-issued F-TRANSFER-END(read) response primitive;

u) L-WRTRQ - signal indicating a re-issued F-WRITE request primitive;

v) L-REARQ - signal indicating a re-issued F-READ request primitive;

w) L-OPEN - signal to a sub-entity of successful processing of F-OPEN primitive;

x) L-CLOSE - signal to a sub-entity of successful processing of F-CLOSE primitive;

y) L-RECVR - signal to a sub-entity to go into or out of the RECOVER-WT state;

z) L-PASIVE - signal to a sub-entity to go into the PASSIVE state.

A.1.2 Incoming states

Add last paragraph.

h) for concurrent overlap, the following suffices indicate the data transfer types of primitives and PDUs:

"R" Read

"W" Write

A.1.4 States

Add last paragraph.

The prefix "Q-" indicates the use of consecutive access. The prefixes "K-R-" & "K-W-" indicate the use of concurrent access and are with respect to read and writes respectively.

A.1.5 Predicates

Add to functional units of FPM.

U15: consecutive overlap
U16: concurrent overlap

A.1.8 Additional State Information

Replace first sentence.

The tables make use of the indicators and other state variables defined in: 6.2, 11.2 and 11.3.

Add last paragraph.

In overlapped access the following symbols are used:

TN - transfer number
RTN - read transfer number
WTN - write transfer number
BTN - bulk transfer number
SBTN - start bulk transfer number
RSTTN - restart transfer number
TE - transfer end

Add the following sub-clauses:

A.1.8.1 FPM in overlapped access

Each transfer request is placed in a queue. It is removed from the queue when a F-TRANSFER-END response is issued or a F-TRANSFER-END confirm is received.

The current state of the FPM always represents the state of the transfer that is currently on the top of the request queue. The FPM places each F-DATA-END primitive that is issued or received on a queue and checks the next request in the request queue to set the read/write indicator(s) for the next transfer.

During the exchange of F-TRANSFER-END primitives, the initiating FPM places the F-TRANSFER-END request in a queue and the responding FPM places the F-TRANSFER-END response in a queue (these are saved for possible re-issuing when a re-synchronisation occurs). When a F-TRANSFER-END confirm is received then the F-TRANSFER-END request is removed from its queue. A F-TRANSFER-END response is removed from its queue when the transfer end response parameter on a F-TRANSFER-END request indicates its receipt by the initiator.

The transfer-end queues, the data-end queues and the read/write indicator(s) are used to determine the next state of the FPM.

To make the state tables manageable for concurrent overlap the basic FPM is considered to consist of the two sub-entities of read FPM and write FPM. Each sub-entity maintains its own states and communicates with the other via global variables. The operation of the sub-entities are independent apart from during cancel and restart. The cancellation or restart of a write may destroy control information of data transfer and transfer end requests for reads; the cancellation or restart of a read may destroy control information of transfer end responses for writes. These have to be saved and re-issued at the completion of resynchronisation to maintain the independence of read and write activities.

A.1.8.2 FERPM in overlapped access

The FERPM is divided into two sub-entities of read and write in concurrent access. It is assumed that under normal operation the read sub-entity will look after the FTAM regime and the file management regime. The write sub-entity stays in the PASSIVE state and enters into the XFER-IDLE state when it receives the local signal L-OPEN (from the read sub-entity) and it will go back to the PASSIVE state when it receives the locals signal L-CLOSE (read sub-entity). Either sub-entity can perform recovery of the FTAM regime and the file management regime. The sub-entity that is performing the recovery will send the local signal L-RECVR, to put the other entity into the RECOVER-WT state. When recovery is completed, the recovering sub-entity will again issue a L-RECVR signal to inform the other sub-entity that recovery is completed. It is assumed that each sub-entity will be informed by different local signals for type 1 and type 2 errors. The same signal will be sent to both sub-entities when type 3 errors occur. The re-sending of any control information is done at the end of the restart phase.

When a write sub-entity receives a type 2 error it will issue a Cancel in write and a read sub-entity will issue a Cancel in read when it receives a type 2 error. Recovery actions do not start until both cancels are completed. The entity that completes the cancel first will be put into the RECOVER-WT state. An indicator is set to remember which entity is in the wait state.

The FERPM is given the task of re-issuing F-TRANSFER-END requests and responses and F-READ and F-WRITE requests after a session resynchronisation. The cancellation of a write will require the initiator to re-issue any possible read requests and transfer end (read) requests. The cancellation of a read will require the responder to re-issue any possible transfer end (write) responses. In addition, during error recovery the initiator will resend any write requests from the transfer request queue and transfer end (write) requests from its docket as well as any from its transfer end queue. The responder will resend and transfer end (read) response from its docket and from its transfer end queue. It is assumed that F-TRANSFER-END primitives will be re-issued before transfer request primitives.

To reduce the complexity of the state tables, all processing of user primitives are held until the completion of error recovery.

A.1.9 Additional processing indicators for the FERPM

During recovery from class II and III errors in concurrent access, the FERPM uses the following extra indicators to co-ordinate events between the two sub-entities.

a) Error3 indicator - it is set when a sub-entity has sent a local L-ERROR3 signal to the other sub-entity to indicate that it is unable to recover from a class II error and is treating the error as a class III error. It is unset when recovery is completed.

b) Write Cancel indicator - it is set when the write sub-entity has issued a CANCEL when an error II occurs. It is unset when recovery is completed.

c) Read Cancel indicator - it is set when the read sub-entity has issued a CANCEL when an error II occurs. It is unset when recovery is completed.

d) Recovery indicator - It is set to READ if a read sub-entity is passive during the recovery from class II/III errors. It is set to WRITE if a write sub-entity is passive during the recovery from class II/III errors.

A.2 FTAM regime management protocol machine (Kernel functional unit)

...

A.3 File regime management protocol machine

...

A.4 Bulk data transfer protocol machine

A.4.1 States - bulk data transfer

Relabel A.4.1:

A.4.1.1 Normal access (non-overlapped)

Add the following sub-clauses.

A.4.1.2 Consecutive access

Q-DXFRIDLE	Data transfer idle in consecutive overlap
Q-READ	Read data transfer in consecutive overlap
Q-I-READ-ENDING	Read transfer end, wait for F-TRANSFER-END request primitive from the internal file service user in consecutive overlap
Q-READ-ENDING	Read transfer ending, wait for transfer end request PDU in consecutive overlap
Q-I-R-XFER-ENDING	Wait for F-TRANSFER-END response primitive after READ from the internal file service user in consecutive overlap
Q-R-XFER-ENDING	Wait for TRANSFER-END response PDU after READ in consecutive overlap
Q-WRITE	Write data transfer in consecutive overlap
Q-I-WRT-ENDING	Write TRANSFER-END, wait for TRANSFER-END request PDU in consecutive overlap
Q-WRT-ENDING	Write transfer-end, wait for transfer end request PDU.
Q-I-W-XFER-ENDING	Wait for F-TRANSFER-END response primitive after WRITE from the internal file service user in consecutive overlap
Q-W-XFER-ENDING	Wait for TRANSFER-END response PDU after write in consecutive overlap
Q-CANCEL-PD	Cancel pending, wait for cancel response PDU in consecutive overlap
Q-I-CANCEL-PD	Cancel pending, wait for F-CANCEL response primitive from the internal file service user in consecutive overlap
Q-RRESTART-PD	Read restart pending, wait for restart response PDU in read operation in consecutive overlap
Q-I-RRESTART-PD	Read restart pending, wait for F-RESTART response primitive in read operation from the internal file service user in consecutive overlap

Q-WRESTART-PD Write restart pending, wait for restart response PDU in write operation in consecutive overlap

Q-I-WRESTART-PD Write restart pending, wait for F-RESTART response primitive in write operation from the internal file service user in consecutive overlap

Q-XFRIDLE-REC Data transfer file during recovery in consecutive overlap

Q-REA-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in consecutive overlap

Q-RES-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in consecutive overlap

Q-CAN-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, cancel requested in consecutive overlap

Q-WRT-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive after WRITE in consecutive overlap

Q-REA-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive after READ in consecutive overlap

Q-RES-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, restart requested in consecutive overlap

Q-CAN-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, cancel requested in consecutive overlap

Q-WRT-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive after WRITE in consecutive overlap

Q-RESTART-CAN-PD Restart cancel pending, wait for cancel or restart response PDU in consecutive overlap

A.4.1.3 Concurrent access

States for READ entity:

K-R-DXFRIDLE Data transfer idle in concurrent overlap in READ
K-READ Read data transfer in concurrent overlap in READ
K-I-READ-ENDING Read transfer end, wait for F-TRANSFER-END request primitive from the internal file service user in concurrent overlap in READ
K-READ-ENDING Read transfer ending, wait for F-TRANSFER-END request PDU in concurrent overlap in READ
K-I-RXFER-ENDING Wait for F-TRANSFER-END response primitive after READ from the internal file service user in concurrent overlap in READ
K-R-XFER-ENDING Wait for F-TRANSFER-END response PDU after READ in concurrent overlap in READ
K-RCANCEL-PD Cancel pending, wait for cancel response PDU in concurrent overlap in READ
K-I-RCANCEL-PD Cancel pending, wait for F-CANCEL response primitive from the internal file service user in concurrent overlap in READ
K-RRESTART-PD Read restart pending, wait for restart response PDU in read operation in concurrent overlap in READ
K-I-RRESTART-PD Read restart pending, wait for F-RESTART response primitive in read operation from the internal file service user in concurrent overlap in READ
K-R-XFRIDLE-REC Data transfer file during recover in concurrent overlap in READ
K-REA-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in concurrent overlap in READ
K-RRES-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in concurrent overlap in READ
K-RCAN-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, cancel requested in concurrent overlap in READ
K-RRES-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, restart requested in concurrent overlap in READ
K-RCAN-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, cancel requested in concurrent overlap in READ
K-REA-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive after READ in concurrent overlap in READ
K-RRESTART-CAN-PD Restart cancel pending, wait for cancel or restart response PdU in concurrent overlap in READ

States for WRITE entity:

K-W-DXFRIDLE Data transfer idle in concurrent overlap in WRITE
K-WRITE Write data transfer in concurrent overlap in WRITE
K-I-WRT-ENDING Write transfer end, wait for F-TRANSFER-END request PDU in concurrent overlap in WRITE

K-WRT-ENDING Write transfer-end, wait for F-TRANSFER-ENDING request PDU in concurrent overlap in WRITE
K-I-WXFER-ENDING Wait for F-TRANSFER-END response primitive after WRITE from the internal file service user in concurrent overlap in WRITE
K-W-XFER-ENDING Wait for F-TRANSFER-END response PDU after WRITE in concurrent overlap in WRITE
K-WCANCEL-PD Cancel pending, wait for cancel response PDU in concurrent overlap in WRITE
K-I-WCANCEL-PD Cancel pending, wait for F-CANCEL response primitive from the internal file service user in concurrent overlap in WRITE
K-WRESTART-PD Write restart pending, wait for restart response PDU in write operation in concurrent overlap in WRITE
K-I-WRESTART-PD Write restart pending, wait for F-RESTART response primitive in write operation from the internal file service user in concurrent overlap in WRITE
K-W-XFRIDLE-REC Data transfer file during recover in concurrent overlap in WRITE
K-WRT-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in concurrent overlap in WRITE
K-WRES-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, restart requested in concurrent overlap in WRITE
K-WCAN-SYMIN-PD Wait for first P-SYNC-MINOR indication primitive, cancel requested in concurrent overlap in WRITE
K-WRES-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, restart requested in concurrent overlap in WRITE
K-WCAN-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive, cancel requested in concurrent overlap in WRITE
K-WRT-SYMCF-PD Wait for first P-SYNC-MINOR confirm primitive after WRITE in concurrent overlap in WRITE
K-WRESTART-CAN-PD Restart cancel pending, wait for cancel or restart response PdU in concurrent overlap in WRITE

A.4.4 Specific actions - bulk data transfer

Relabel A.4.4:

A.4.4.1 Normal access (non-overlapped)

Add the following sub-clause.

A.4.4.2 Consecutive and Concurrent access

[2] Add a PDU to the current PSDU and terminate the current PSDU.

[15] Preserve the PDU for further processing

[16] Add a PDU to the current PSDU and optionally terminate the current PSDU, according to local decision.

[18] Add the data given on the F-DATA request to the current PSDU. Optionally (depending upon local system considerations) terminate the PSDU.

[21] Unset discard indicator.

[25] Terminate the current PSDU.

[44] Increment the bulk transfer number.

[47] Preserve the primitive parameters for subsequent processing. In some states, it happens that when F-CANCEL primitive is preserved, there is already an F-RESTART primitive preserved: in this case, the F-RESTART primitive shall be discarded.

[401] Place request in request queue if request not already in the queue.

[402] Add a PDU to the current PSDU and terminate the current PSDU or optionally send the PDU when the current PSDU is terminated depending upon local system considerations.

[403] Use as parameter the user data from P-SYMIN.

[404] Place transfer end request in TE request queue if request not already in the queue.

[405] Remove first TE request from TE request queue and update the last TE confirm received and remove corresponding

entry for TN received in checkpoint table.

[406] Remove first transfer request from transfer request queue.

[407] Place current TN in DE request queue.

[408] Remove first DE request from DE request queue.

[409] Set indicator to read.

[410] Set indicator to write.

[411] Mark the next transfer request (to show that the current TN has been updated in the checkpoint table)

[412] Add 1 to current transfer number and in checkpoint table corresponding to TN, set TN, BTN to be the current TN and BTN and set checkpoint counter to zero.

[413] Set BTN to be SBTN.

[414] Place current TN in DE indication queue.

[415] Remove first DE indication from DE indication queue.

[416] Send PDU with parameters: TN set to BTN - SBTN and TN of last TE confirm (or confirms) received as user data on a P-Resync request (or response) with second item set as abandon. (*Init read)

[417] Send PDU with parameters: TN set to BTN - SBTN and TN of last TE request (or requests) issued as user data on a P-Resync request (or response) with first item set as abandon. (* init write)

[418] Send PDU with parameters: TN set to BTN - SBTN and TN of last TE response (or response) issued as user data on a P-Resync request (or response) with first item set as abandon. (*resp read)

[419] Send PDU with parameters: TN set to BTN - SBTN and TN of last TE indication (or indications) received as user data on a P-Resync request (or response) with second item set as abandon. (*resp write)

[420] Clear queues and variables except BTN and the NSPNs.

[421] Reissue TE read requests and remove any TE write requests from TE request queue (or queues).

[425] PDU with TN and the checkpoint identifier is sent as user data on a P-RESYNC response primitive with first resync item set to restart and the first resync point serial number equal to those received on the P-RESYNC indication

[426] PDU with TN and the checkpoint identifier is sent as user data on a P-RESYNC request primitive with second resync item set to restart and the second resync point serial number equal to the sum of the checkpoint identifier, on the F-RESTART primitive, and the synchronization offset of the corresponding TN plus 1.

[427] PDU with TN and the checkpoint identifier is sent as user data on a P-RESYNC request primitive with first resync item set to restart and the first resync point serial number equal to the sum of the checkpoint identifier, on the F-RESTART primitive, and the synchronization offset of the corresponding TN plus 1.

[428] PDU with TN and the checkpoint identifier is sent as user data on a P-RESYNC response primitive with second resync item set to restart and the second resync point serial number equal to those received on the P-RESYNC indication.

[429] Place Transfer-end response in TE response queue if response is not already in the queue.

[430] Remove corresponding TE response from TE response queue and remove corresponding entry from checkpoint entry table.

[431] Update the TN for the last transfer end indication.

[432] Use as parameter the TN and the TN of the last transfer end confirm (or confirms) received. (*init read)

[433] Use as parameter the TN and the TN of the last transfer end request (or requests) issued. (init write)

[434] If the response queue is not empty, remove TE responses from its queue up to TN of confirm received on the indication.

[435] Use as parameters the TN and the TN of the last transfer end indication (or indications) received. (*resp write)

[436] Use as parameter the TN and the TN of the last transfer end response (or responses) issued. (*resp read)

[438] Send PDU as user data with BTN = smaller of the BTN and the (transfer number on Cancel + SBTN) on a P-RESYNC request or response primitive with second item set to abandon..

[439] Send PDU as user data with BTN = smaller of the BTN and the (transfer number on Cancel + SBTN) on a P-RESYNC request or response primitive with first item set to abandon..

[440] If the DE queues are not empty, remove from the end of queues entries up to but not including the entry for TN, the smallest TN corresponding to the BTN

[441] Reissue Read requests with TN > current TN and remove any write requests from transfer request queue (or queues).

[444] Use (BTN received -SBTN) as the transfer number parameter.

[445] Set user data to TN and checkpoint identifier of I-CHECK request (or response) primitive.

[446] If there is no change in TN then use the TN and the checkpoint identifier received as parameters; otherwise use the new TN and the (final checkpoint identifier of the new TN +1) as parameters.

[447] Clear R/W indicator.

[449] The smaller of the BTN and the (TN received + SBTN) becomes the new BTN and the smallest TN in the checkpoint entry table corresponding to the new BTN becomes the new TN.

[451] Use for the sync point serial number, the checkpoint received plus the offset for the TN specified in F-CHECK.

[452] Use the TN in user data of P-SYMCF and use for checkpoint the sync point serial number minus the offset for the TN.

[453] Set discard indicator and set the outstanding checkpoint counter(s) for entries >= TN to zero.

[454] Increment the checkpoint identifier expected for the TN specified in F-CHECK.

[455] Increment the outstanding checkpoint counter for the TN specified in F-CHECK.

[456] Decrement the outstanding checkpoint counter for the TN specified in F-CHECK according to the checkpoint number confirmed (see 15.3.2).

[457] Set the checkpoint identifier expected for the corresponding TN to one.

[458] Set the synchronization offset for the corresponding TN to the value of the second NSPN minus the checkpoint identifier expected.

[459] Set the checkpoint identifier expected to the value negotiated plus one for the TN.

[460] Discard any user data not yet delivered and set outstanding checkpoint counter(s) for entries >= TN to zero.

[461] Use TN as the parameter.

[462] The checkpoint identifier for the TN shall be equal to the value received on the PDU.

[463] Reissue TE write response and remove any TE read response from TE response queue (or queues).

[464] Add 1 to the first NSPN (sender).

[465] Add 1 to the second NSPN (receiver).

[466] Set the first NSPN equal to the synch point serial number negotiated with the presentation service provider.

[467] Set the second NSPN equal to the synch point serial number negotiated with the presentation service provider.

[468] Store the synchronization point serial number provided by the presentation service provider as the first NSPN.

[469] Store the synchronization point serial number provided by the presentation service provider as the second NSPN.

[470] Set the synchronization offset for the corresponding TN to the value of the first NSPN minus the checkpoint identifier expected.

[471] For all but the first entry in the checkpoint table that has a BTN = current BTN (if any), increment the BTN by 1.

Note - TN stands for RTN for concurrent overlap read entity and WTN for concurrent overlap write.

A.4.5 Predicates - bulk data transfer

Relabel A.4.5:

A.4.5.1 Normal access (non-overlapped)

Add the following sub-clause.

A.4.5.2 Consecutive and Concurrent access

P8: The context is not within the defined context set.

P14: The synchronization point exceeds 999 998, exceeding the session limitation.

P15: Discard indicator is set.

P27: The outstanding checkpoint counter exceeds the agreed maximum.

P400: The consecutive overlap functional unit has been negotiated.

P401: The Symm Sync functional unit of Session has been negotiated.

P402: The expected checkpoint identifier for the TN equals the sync point serial number received minus the offset for the TN.

P403: The next request is marked (indicating that TN has already been updated in the checkpoint table)

P404: Next transfer is read.

P405: Next transfer is write.

P406: Number in transfer queue exceeds limit set in open.

P407: The symm resync functional unit has been negotiated.

P408: The Data end request queue is empty.

P409: Read indicator is set.

P410: Transfer request queue is empty.

P411: The data end indication queue is empty.

P412: Last transfer end confirm TN received < last transfer end response TN issued.

P413: Cancel read or restart read or transfer-end read.

P414: Cancel write or restart write or transfer-end write.

P415: Last transfer end indication TN received < last transfer end request TN issued.

P416: TN on TRERP PDU = TN of last transfer indication received.

P417: Cancel PDU preserved

P418: The transfer end request queue is not empty.

P419: Complete re-issue of transfer end requests

P420: Complete re-issue of transfer end response.

P421: TN > last TN in TE request queue.

P422: Last transfer end indication TN received > TN of last item in TE response queue.

P423: The TE response is already in the response queue and the TN of Transfer end response > TN of last TN confirm received by the initiating entity.

P424: The write indicator is set.

P425: On the Restart, Recover or Cancel primitive, the TN confirm (or confirms) received by initiator < the TN of last item in TE response queue (or queues)

P426: On the Restart, Recover or Cancel primitive, the TN indication (or indications) received by the responder < the TN of last item in TE request queue (or queues)

P427: TN of transfer end primitive = TN of last item in TE queue (or the last item removed from the queue if the TE queue is empty)

P428: TN in TRERQ PDU <= last transfer end indication TN received.

P429: TN > Last TE end indication + 1.

P430: Next TE request in queue is read.

P431: There are transfer requests with TN > current TN in the transfer request queue (or queues).

P432: THe TE request is already in a TE request queue and the TN of transfer end request > the last TN indication received by the responding entity.

P433 Complete re-issue of transfer request.

P434: There is no next transfer request.

P436: Request already in queue.

P437: TN=1

P447: Both activity types are cleared

Note - TN stands for RTN for concurrent overlap read entity and WTN for concurrent overlap write entity.

A.4.6 Initiating state tables - bulk data transfer

Relabel A.4.6:

A.4.6.1 Normal access (non-overlapped) - Initiator

Relabel A.4.7:

A.4.6.2 Detailed entries - normal access

Add the following sub-clauses.

A.4.6.3 Consecutive access - Initiator

STATE / EVENT	Q-DXFRIDLE	Q-DXFRIDLE-REC	Q-READ-SYMIN-PD	Q-RESET-SYMIN-PD	Q-CANCEL-SYMIN-PD	Q-READ	Q-I-READ-ENDING	Q-R-XFER-ENDING	Q-WRT-SYMCF-PD	Q-RESET-SYMCF-PD	Q-CANCEL-SYMCF-PD	Q-WRITE	Q-I-WRT-ENDING	Q-W-XFER-ENDING	Q-CANCEL-PD	Q-I-CANCEL-PD	Q-RRESTART-PD	Q-WRESTART-PD	Q-I-RRESTART-PD	Q-I-WRESTART-PD	Q-RESTART-CAN-PD
I-REARQ	1	35	45			45	64	56	47			47	47	66							
I-WRTRQ	2	36	46			46	46	65	48			48	48	65							
I-DATRQ									4			4		51							
I-DAERQ									5			5		52							
DATIN						6	49	49							41		41				
DAERQ						8	50	50							41		41				
I-TRERQ							9						10								
TRERP								11						3							
I-CANRQ			31	31		12	12	54	34	34		7					38	38	12	7	
CANRP															13						
CANRQ						14			14			14	14	14	15		14	14	14	14	39
I-CANRP																16					
I-CHKRQ									17			17									
P-SYMCF									32	60	7	18	37	37	41						
P-SYMIN			29	21	12	19	42	42							41						
I-CHKRP						20	20	63													
I-RESRQ			30			21	21	59	33			22		60							
RESRP																	23	24			54
RESRQ						25		62	26			26	26	26			43	44			
I-RESRP																			27	28	

A.4.6.4 Detailed entries - consecutive access

1:	P400&(~P401|~P407):	L-ERRABT	=>samestate
	P400&P401&P407:	[420],[401],[409],[44],[412],[413],[457],[458],REARQ[2]	=>Q-REA-SYMIN-PD
2:	P400&(~P401|~P407):	L-ERRABT	=>samestate
	P400&P401&P407:	[420],[401],[410],[44],[412],[413],WRTRQ[2],[464],	
	P14:	L-ERRABT	=>samestate
	~P14:	P-SYMRQ,[457],[470]	=>Q-WRT-SYMCF-PD
3:	P15:		=>samestate
	~P15:	I-TRECF,[405],[406],	
	P410:		=>Q-DXFRIDLE
	~P410&P409:	[412],[411],[457],[458]	=>Q-READ
	~P410&P424&P408&P427:		=>Q-WRITE
	~P410&P424&~P408&P427:		=>Q-I-WRT-ENDING
	~P410&~P430&~P427:		=>Q-W-XFER-ENDING
4:	P8:	L-ERRABT	=>samestate
	~P8:	P-DATRQ[18]	=>samestate
5:		DAERQ[16],[407]	=>Q-I-WRT-ENDING
6:	P15:		=>samestate
	~P15:	I-DATIN	=>samestate
7:		[453],[460],CANRQ[417]	=>Q-CANCEL-PD
8:	P15:		=>samestate
	~P15:	I-DAEIN,[414],	
	~P404&~P405:	[447]	=>Q-I-READ-ENDING
	P404&~P403:	[412],[411],[409],[457],[458]	=>Q-I-READ-ENDING
	P404&P403:	[409]	=>Q-I-READ-ENDING
	P405:	[410]	=>Q-I-READ-ENDING
9:	P432:	TRERQ[402]	=>samestate
	~P432:	[415],TRERQ[432][402],[404],[471],	
	P404&~P403:	[44],[409],[412],[411],[457],[458]	=>Q-R-XFER-ENDING
	P404&P403:	[44],[409]	=>Q-R-XFER-ENDING
	P405:	[44],[410],[412],[411],[457],[470]	=>Q-R-XFER-ENDING
	~P404&~P405:	[447]	=>Q-R-XFER-ENDING
10:		[408],TRERQ[402],[432],[404],[471],	
	P404:	[44],[409]	=>Q-W-XFER-ENDING
	P405&~P403:	[44],[410],[412],[411],[457],[470]	=>Q-W-XFER-ENDING
	P405&P403:	[44],[410]	=>Q-W-XFER-ENDING
	~P404&~P405:	[447]	=>Q-W-XFER-ENDING
11:	P15:		=>samestate
	~P15:	I-TRECF,[405],[406],	
	P410&P427:		=>Q-DXFRIDLE
	P410&~P427:		=>samestate
	~P410&P409&P411&P427:		=>Q-READ
	~P410&P424&P408&P427:		=>Q-WRITE
	~P410&P424&~P408&P427:		=>Q-I-WRT-ENDING
	~P410&P409&~P411&P427:		=>Q-I-READ-ENDING
	~P410&~P427&P430		=>Q-R-XFER-ENDING
	~P410&~P427&~P430		=>Q-W-XFER-ENDING

12:	[453],[460],CANRQ[416]	=>Q-CANCEL-PD
13: ~P414&P413:	[449],[469],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
P414&~P413:	[449],[468],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
14: ~P414&P413:	[449],[469],[453],[460],I-CANIN[461]	=>Q-I-CANCEL-PD
P414&~P413:	[449],[468],[453],[460],I-CANIN[461]	=>Q-I-CANCEL-PD
15: ~P413&P414:	[449],CANRP[417],[468],[21],I-CANCF[461],[440]	=>Q-XFRIDLE
~P414&P413:	[449],CANRP[416],[469],[21],I-CANCF[461],[440]	=>Q-XFRIDLE
16: ~P413&P414:	[468],[21],CANRP[417],[440]	=>Q-XFRIDLE
~P414&P413:	[469],[21],CANRP[416],[440]	=>Q-XFRIDLE
17:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate
18:	I-CHKCF[452]	=>samestate
19: ~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
20:	P-SYMRP[451][445]	=>samestate
21:	[453],RESRQ[426]	=>Q-RRESTART-PD
22:	[453],RESRQ[427]	=>Q-WRESTART-PD
23:	[449],[467],[459],[458],[21],	
P418:	I-RESCF[446]	=>Q-R-XFER-ENDING
~P418:	I-RESCF[446]	=>Q-READ
24:	[449],[466],[459],[470],[21],	
P418:	I-RESCF[446]	=>Q-W-XFER-ENDING
~P418:	I-RESCF[446]	=>Q-WRITE
25:	[449],I-RESIN[446],[460]	=>Q-I-RRESTART-PD
26:	[449],I-RESIN[446]	=>Q-I-WRESTART-PD
27: P418:	[467],[459],[458],RESRP[428],[440]	=>Q-R-XFER-ENDING
~P418&P411:	[467],[459],[458],RESRP[428],[440]	=>Q-READ-ENDING
~P418&~P411:	[467],[459],[458],RESRP[428],[440]	=>Q-READ
28: P418:	[466],[459],[470],RESRP[425],[440]	=>Q-W-XFER-ENDING
~P418&P408:	[466],[459],[470],RESRP[425],[440]	=>Q-I-WRT-ENDING
~P418&~P408:	[466],[459],[470],RESRP[425],[440]	=>Q-WRITE
29:	[465],[458],P-SYMRP	=>Q-READ
30:	[453],[47]	=>Q-RES-SYMIN-PD
31:	[453],[47]	=>Q-CAN-SYMIN-PD
32:	[470]	=>Q-WRITE
33:	[453],[47]	=>Q-RES-SYMCF-PD
34:	[453],[47]	=>Q-CAN-SYMCF-PD
35:	REARQ[2],[458],[409]	=>Q-READ-SYMIN-PD

36:	[458],WRTRQ[2],[410],[464],	
P14:	L-ERRABT	=>samestate
~P14:	P-SYMRQ	=>Q-WRT-SYMCF-PD
37: ~P421:	L-ERRABT	=>samestate
P421:	I-CHKCF[452]	=>samestate
38:	[47]	=>Q-RESTART-CAN-PD
39: ~P413&P414:	[449],CANRP[417],[466],[21],I-CANCF[461],[440]	=>Q-XFRIDLE
~P414&P413:	[449],CANRP[416],[467],[21],I-CANCF[461],[440]	=>Q-XFRIDLE
41:		=>samestate
42: ~P409:	L-ERRABT	=>samestate
P409:	[455],	
~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
43:	[449],[467],[459],[458],RESRP[428],I-RESCF[446],[440],[43],	
P418:		=>Q-R-XFER-ENDING
~P411&~P418:		=>Q-READ-ENDING
P411&~P418:		=>Q-READ
44:	[449],[466],[459],[470],RESRP[425],I-RESCF[446],[440],[43],	
P418:		=>Q-W-XFER-ENDING
~P418&~P408:		=>Q-I-WRT-ENDING
~P418&P408:		=>Q-WRITE
45:	[401],REARQ[2]	=>samestate
46:	[401],WRTRQ[2]	=>samestate
47:	[401],REARQ[402]	=>samestate
48:	[401],WRTRQ[402]	=>samestate
49: ~P409:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DATIN	=>samestate
50: ~P409:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DAEIN,[414],	
~P404&~P405:	[447]	=>samestate
P404&~P403:	[412],[411],[409],[457],[458]	=>samestate
P404&P403:	[409]	=>samestate
P405:	[410]	=>samestate
51: ~P424:	L-ERRABT	=>samestate
P8:	L-ERRABT	=>samestate
~P8:	P-DATRQ[18]	=>samestate
52: ~P424:	L-ERRABT	=>samestate
P424:	DAERQ[16],[407]	=>samestate
54: P413:	[453],[460],CANRQ[416]	=>Q-CANCEL-PD
P414:	[453],[460],CANRQ[417]	=>Q-CANCEL-PD
56: P434:	[401],[409],[44],[412],[411],[457],[458],REARQ[402]	=>samestate
~P434:	[401],REARQ[402]	=>samestate

59: ~P409&~P424:	L-ERRABT	=>samestate
P409:	[453],RESRQ[426],[460]	=>Q-RRESTART-PD
P424:	[453],RESRQ[427]	=>Q-WRESTART-PD
60: ~P424:	L-ERRABT	=>samestate
P424:	[453],RESRQ[427]	=>Q-WRESTART-PD
62: ~P409|~P424:	L-ERRABT	=>samestate
P409:	[449],I-RESIN[446],[460]	=>Q-I-RRESTART-PD
P424:	[449],[460],I-RESIN[446]	=>Q-I-WRESTART-PD
63: ~P409:	L-ERRABT	=>samestate
P409:	[456],P-SYMRP[451][455]	=>samestate
64: P434:	[401],[409],[412],[411],[457],[458],REARQ[402]	=>samestate
~P434:	[401],REARQ[402]	=>samestate
65: P434:	[401],[410],[44],[412],[411],[457],[470],WRTRQ[402]	=>samestate
~P434:	[401],WRTRQ[402]	=>samestate
66: P434:	[401],[44],[409],REARQ[402]	=>samestate
~P434:	[401],REARQ[402]	=>samestate

A.4.6.5 Concurrent access - Initiator

A.4.6.5.1 Concurrent read access - Initiator

STATE / EVENT	K-R-DXFRIDLE	K-R-DXFRIDLE-REC	K-REA-SYMIN-PD	K-RRES-SYMIN-PD	K-RCAN-SYMIN-PD	K-READ	K-I-READ-ENDING	K-R-XFER-ENDING	K-RCANCEL-PD	K-I-RCANCEL-PD	K-RRESTART-PD	K-I-RRESTART-PD	K-RRESTART-CAN-PD
I-REARQ	1	35	47			47	64	56					
DATIN						6	49	49	41		41		
DAERQ						8	50	50	41		41		
I-TRERQ-R							9						
TRERP-R								11					
I-CANRQ-R			31	31		12	12	54			38	12	
CANRP-R									13				
CANRQ-R						14			15		14	14	39
I-CANRP-R										16			
P-SYMIN			29	21	12	19	42	42	41				
I-CHKRP						20	20	63					
I-RESRQ-R			30			21	21	59					
RESRP-R											23		54
RESRQ-R						25		62			43		
I-RESRP-R												27	

A.4.6.5.2 Detailed entries - concurrent read access

1:	P440&(~P401|~P407):	L-ERRABT	=>samestate
	P440&P401&P407:	[420],[401],[409],[44],[412],[413],[457],[458]REARQ[2]	=>K-REA-SYMIN-PD
6:	P15:		=>samestate
	~P15:	I-DATIN	=>samestate
8:	P15:		=>samestate
	~P15:	I-DAEIN,[414],	
	~P404:	[447]	=>K-I-READ-ENDING
	P404&~P403:	[412],[411],[409],[457],[458]	=>K-I-READ-ENDING
	P404&P403:	[409]	=>K-I-READ-ENDING
9:	P432:	TRERQ-R[402]	=>samestate
		[415],TRERQ-R[432][402],[404],[471],	

P404&~P403:	[44],[409],[412],[411],[457],[458]	=>K-R-XFER-ENDING
P404&P403:	[44],[409]	=>K-R-XFER-ENDING
~P404:	[447]	=>K-R-XFER-ENDING
11: P15:		=>samestate
~P15:	I-TRECF-R,[405],[406],	
P410&P427:		=>K-R-DXFRIDLE
P410&~P427:		=>samestate
~P410&P409&P411&P427:		=>K-READ
~P410&P409&~P411&P427:		=>K-I-READ-ENDING
~P410&~P427:		=>K-R-XFER-ENDING
12:	[453],[460],CANRQ-R[416]	=>K-RCANCEL-PD
13:	[449],[469],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
14:	[449],[469],[453],[460],I-CANIN-R[461]	=>K-I-RCANCEL-PD
15:	[449],CANRP-R[416],[469],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
16:	[21],CANRP-R[416],[440],[469]	=>K-R-DXFRIDLE
19: ~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
20:	[456],P-SYMRP[451][445]	=>samestate
21:	[453],RESRQ-R[426],[460]	=>K-RRESTART-PD
23:	[449],[467],[459],[458],[21],	
P418:	I-RESCF-R[446]	=>K-R-XFER-ENDING
~P418:	I-RESCF-R[446]	=>K-READ
25:	[449],I-RESIN-R[446],[460]	=>K-I-RRESTART-PD
27: P418:	[467],[459],[458],RESRP-R[428],[440]	=>K-R-XFER-ENDING
~P418&P411:	[467],[459],[458],RESRP-R[428],[440]	=>K-READ-ENDING
~p418&~p411:	[467],[459],[458],RESRP-R[428],[440]	=>K-READ
29:	[465],[458],P-SYMRP	=>K-READ
30:	[453],[47]	=>K-RRES-SYMIN-PD
31:	[453],[47]	=>K-RCAN-SYMIN-PD
35:	REARQ[2],[458],[409]	=>K-REA-SYMIN-PD
38:	[47]	=>K-RRESTART-CAN-PD
39:	[449],CANRP-R[416],[467],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
41:		=>samestate
42: ~P409:	L-ERRABT	=>samestate
P409:	[455],	
~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
43:	[449],[467],[459],[458],[21], RESRP-R[428],I-RESCF-R[446],[440],	
P418:		=>K-R-XFER-ENDING
~P411&~P418:		=>K-READ-ENDING

P411&~P418:		=>K-READ
47:	[401],REARQ[402]	=>samestate
49: ~P409:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DATIN-R	=>samestate
50: ~P409:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DAEIN-R,[414],	
~P404:	[447]	=>samestate
P404&~P403:	[412],[411],[409],[457],[458]	=>samestate
P404&P403:	[409]	=>samestate
54: P409: [453],[460],CANRQ-R[416]		=>K-RCANCEL-PD
~P409:	L-ERRABT	=>samestate
56: P434:	[401],[409],[44],[412],[411],[457],[458],REARQ[402]	=>samestate
~P434:	[401],REARQ[402]	=>samestate
59: ~P409	L-ERRABT	=>samestate
P409:	[453],RESRQ-R[426],[460]	=>K-RRESTART-PD
62: ~P409:	L-ERRABT	=>samestate
P409:	[449],I-RESIN-R[446],[460]	=>K-I-RRESTART-PD
63: ~P409:	L-ERRABT	=>samestate
P409:	[456],P-SYMRP[451][455]	=>samestate
64: P434:	[401],[409],[412],[411],[457],[458],REARQ[402]	=>samestate
~P434:	[401],REARQ[402]	=>samestate

A.4.6.5.3 Concurrent write access - Initiator

STATE / EVENT	K-W-DXFRIDLE	K-WDXFRIDLE	K-WRITE	K-WREAD	K-WCANCEL	K-WRITING	K-WEND	K-WCANCEL	K-WCANCEL	K-WREAD	K-WREAD	K-WRITING
I-WRTRQ	2	36	48			48	46	65				
I-DATRQ			4			4		51				
I-DAERQ			5			5		52				
I-TRERQ-W							19					
TRERP-W								3				
I-CANRQ-W			34	34		7				38	7	
CANRP-W									13			
CANRQ-W			14			14	14	14	15	14	14	39
I-CANRP-W												
I-CHKRQ-W			17			17						
P-SYMCF			32	60	7	18	37	37	41			
I-RESRQ-W			33			22		60				
RESRP-W										24		54
RESRQ-W			26			26	26	26		44		
I-RESRP-W											28	

A.4.6.5.4 Detailed entries - concurrent write access

2:	P440&(~P401|~P407):	L-ERRABT	=>samestate
	P440&P401&P407:	[420],[401],[410],[44],[412],[413],WRTRQ[2],[464],	
	P14:	L-ERRABT	=>samestate
	~P14:	P-SYMRQ,[457],[470]	=>K-WRT-SYMCF-PD
3:	P15:		=>samestate
	~P15:	I-TRECF-W,[405],[406],	
	P410:		=>K-W-DXFRIDLE
	~P410&P424&P408&P427:		=>K-WRITE
	~P410&P424&~P408&P427:		=>K-I-WRT-ENDING
	~P410&~P430&~P427:		=>K-W-XFER-ENDING
4:	P8:	L-ERRABT	=>samestate
	~P8:	P-DATRQ[18]	=>samestate
5:		DAERQ[16],[407]	=>K-I-WRT-ENDING

7:	[453],[460],CANRQ-W[417]	=>K-WCANCEL-PD
10	[408],TRERQ-W[402],[432],[404],[471],	
P405&~P403:	[44],[410],[412],[457],[470]	=>K-W-XFER-ENDING
P405&P403:	[44],[410]	=>K-W-XFER-ENDING
~P405:	[447]	=>K-W-XFER-ENDING
13:	[449],[468],[21],I-CANCF-W[461],[440]	=>K-W-DXFRIDLE
14:	[449],[468],[453],[460],I-CANIN-W[461]	=>K-I-WCANCEL-PD
15: P418:	[449],CANRP-W[417],[468],[21],I-CANCF-W[461]	=>K-W-XFRIDLE
~P418:	[449],CANRP-W[417],[468],[21],I-CANCF-W[461],[440]	=>K-W-XFRIDLE
16:	[468],[21],CANRP-W[417],[440]	=>K-W-XFRIDLE
17:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate
18:	I-CHKCF[452]	=>samestate
22:	[453],RESRQ-W[427]	=>K-WRESTART-PD
24:	[449],[466],[459],[470],[21],	
P418:	I-RESCF-W[446]	=>K-W-XFER-ENDING
~P418:	I-RESCF-W[446]	=>K-WRITE
26:	[449],I-RESIN-W[446]	=>K-I-WRESTART-PD
28: P418:	[466],[459],[470],RESRP-W[425],[440]	=>K-W-XFER-ENDING
~P418&P408:	[466],[459],[470],RESRP-W[425],[440]	=>K-I-WRT-ENDING
~P418&~P408:	[466],[459],[470],RESRP-W[425],[440]	=>K-WRITE
32:	[458]	=>K-WRITE
33:	[453],[47]	=>K-WRES-SYMCF-PD
34:	[453],[47]	=>K-WCAN-SYMCF-PD
36:	[458],WRTRQ[2],[410],[464],	
P14:	L-ERRABT	=>samestate
~P14:	P-SYMRQ	=>K-WRT-SYMCF-PD
37: ~P421:	L-ERRABT	=>samestate
P421:	I-CHKCF[452]	=>samestate
38:	[47]	=>K-WRESTART-CAN-PD
39:	[449],CANRP-W[417],[466],[21],I-CANCF-W[461],[440]	=>K-W-XFRIDLE
41:		=>samestate
44:	[449],[466],[459],[470],	
	RESRP-W[425],I-RESCF-W[446],[440],	
P418:		=>K-W-XFER-ENDING
~P408&~P418:		=>K-I-WRT-ENDING
~P418&P408:		=>K-WRITE
46:	[401],WRTRQ[2]	=>samestate

48:	[401],WRTRQ[402]	=>samestate
51: ~P424:	L-ERRABT	=>samestate
P8:	L-ERRABT	=>samestate
~P8:	P-DATRQ[18]	=>samestate
52: ~P424:	L-ERRABT	=>samestate
P424:	DAERQ[16],[407]	=>samestate
54: P424:	[453],[460],CANRQ-W[417]	=>K-WCANCEL-PD
~P424:	L-ERRABT	=>samestate
60: ~P424:	L-ERRABT	=>samestate
P424:	[453],RESRQ-W[427]	=>K-WRESTART-PD
65: P434:	[410],[44],[412],[411],[457],[470],WRTRQ[402],[401]	=>samestate
~P434:	WRTRQ[402],[401]	=>samestate

A.4.7 Responding entity state tables - bulk data transfer

Relabel A.4.8:

A.4.7.1 Normal access (non-overlapped) - Responder

Relabel A.4.9:

A.4.7.2 Detailed entries - normal access

Add the following sub-clauses.

A.4.7.3 Consecutive access - Responder

STATE / EVENT	Q-DXFRIDLE	Q-DXFRIDLE-REC	Q-READ-SYMCF-PD	Q-RES-SYMCF-PD	Q-CAN-SYMCF-PD	Q-READ	Q-READ-ENDING	Q-IR-XFER-ENDING	Q-WRT-SYMIN-PD	Q-RES-SYMIN-PD	Q-CAN-SYMIN-PD	Q-WRITE	Q-WRT-ENDING	Q-IW-XFER-ENDING	Q-CANCEL-PD	Q-I-CANCEL-PD	Q-RRESTART-PD	Q-WRESTART-PD	Q-I-RRESTART-PD	Q-I-WRESTART-PD	Q-RESTART-CAN-PD
REARQ	1	36	47			47	64	64	47			47	47	47							
WRTRQ	2	37	48			48	48	65	48			48	48	63							
I-DATRQ			4			4	51	51													
I-DAERQ			5			5	52	52													
DATIN								49				6		49	41			41			
DAERQ								50				8		50	41			41			
TRERQ							9	9					10		41			41			
I-TRERP								11						43							
I-CANRQ			30	30		12		54	33	33		7	7	7			39	39	12	7	
CANRP															13						
CANRQ			14	14		14	14					14			44		14	14	14	14	40
I-CANRP																15					
I-CHKRQ			16			16	42	42													
P-SYMCF			28	20	12	18	38	38							41						
P-SYMIN									31	34	35	17			41						
I-CHKRP												19	19								
I-RESRQ			29			20	60	59	32			21	21	59							
RESRP																	22	23			54
RESRQ			24			24	24	62				25		25			45	46			
I-RESRP																			26	27	

A.4.7.4 Detailed entries - consecutive access

1:	P400&(~P401|~P407):	L-ERRABT	=>samestate
	P400&P401&P407&P437:	[420],[401],[409],[44],[412],[413],I-REAIN,[464],	
	P400&P401&P407&~P437:	[401],[409],[44],[412],I-REAIN,[464],	
	P14:	L-ERRABT	=>samestate
	~P14:	P-SYMRQ,[457],[470]	=>Q-REA-SYMCF-PD
2:	P400&(~P401|~P407):	L-ERRABT	=>samestate
	P400&P401&P407&P437:	[420],[401],[410],[44],[412],[413],[457],[458],I-WRTIN	=>Q-WRT-SYMIN-PD
	P400&P401&P407&~P437:	[401],[410],[44],[412],[457],[458],I-WRTIN	=>Q-WRT-SYMIN-PD
4:	~P8:	P-DATRQ[18]	=>samestate
	P8:	L-ERRABT	=>samestate
5:		DAERQ[2],[407],[471],	
	~P404&~P405:	[447]	=>Q-I-READ-ENDING
	P405:	[44],[410]	=>Q-I-READ-ENDING
	P404&~P403:	[44],[412],[411],[409],[457],[470]	=>Q-I-READ-ENDING
	P404&P403:	[44],[409]	=>Q-I-READ-ENDING
6:	P15:		=>samestate
	~P15:	I-DATIN	=>samestate
7:		[453],[460],CANRQ[419]	=>Q-CANCEL-PD
8:	P15:		=>samestate
	~P15:	I-DAEIN,[414]	=>Q-WRT-ENDING
9:	P15:		=>samestate
	~P15:	[408],I-TREIN,[434],[431],	
	P405&~P403:	[410],[412],[411],[457],[458]	=>Q-I-RXFER-ENDING
	P405&P403:	[410]	=>Q-I-RXFER-ENDING
	~P405:		=>Q-I-RXFER-ENDING
10:	P15:		=>samestate
	~P15:	[415],I-TREIN,[434],[431],	
	P405&~P403:	[410],[412],[411],[457],[458]	=>Q-I-WXFER-ENDING
	P405&P403:	[410]	=>Q-I-WXFER-ENDING
	P404:	[409]	=>samestate
	~P405&~P404:	[447]	=>Q-I-WXFER-ENDING
11:	P423:	TRERP[402]	=>samestate
	~P423:	TRERP[435][402],[429],[406],	
	P410&P416:		=>Q-DXFRIDLE
	P410&~P416:		=>samestate
	~P410&P409&P408&P416:		=>Q-READ
	~P410&P409&~P408&P416:		=>Q-READ-ENDING
	~P410&P424&P411&P416:		=>Q-WRITE
	~P410&P424&~P411&P416:		=>Q-WRT-ENDING
	~P410&~P416&P404:		=>Q-I-RXFER-ENDING
	~P410&~P416&P405:		=>Q-I-WXFER-ENDING
12:		[453],[460],CANRQ[418]	=>Q-CANCEL-PD
13:	P413:	[449],[468],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
	P414:	[449],[469],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE

14: P413:	[449],[468],[453],[460],I-CANIN[461]	=>Q-I-CANCEL-PD
P414:	[449],[469],[453],[460],I-CANIN[461]	=>Q-I-CANCEL-PD
15: ~P413&P414:	[21],CANRP[419],[469],[440]	=>Q-DXFRIDLE
P413&~P414:	[21],CANRP[418],[468],[440]	=>Q-DXFRIDLE
16:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate
17: ~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
18:	I-CHKCF,[452]	=>samestate
19:	P-SYMRP[451][445]	=>samestate
20:	[453],RESRQ[427]	=>Q-RRESTART-PD
21:	[453],RESRQ[426]	=>Q-WRESTART-PD
22:	[449],[466],[459],[470],[21],	
P422:	I-RESCF[446]	=>Q-I-RXFER-ENDING
~P422:	I-RESCF[446]	=>Q-READ
23:	[449],[467],[459],[458],[21],	
P422:	I-RESCF[446]	=>Q-I-WXFER-ENDING
~P422:	I-RESCF[446]	=>Q-WRITE
24:	[449],I-RESIN[446]	=>Q-I-RRESTART-PD
25:	[449],[460],I-RESIN[446]	=>Q-I-WRESTART-PD
26: P422:	[466],[459],[470],RESRP[425],[440]	=>Q-I-RXFER-ENDING
~P422&P408:	[466],[459],[470],RESRP[425],[440]	=>Q-READ-ENDING
~P422&~P408:	[466],[459],[470],RESRP[425],[440]	=>Q-READ
27: P422:	[467],[459],[458],RESRP[428],[440]	=>Q-I-WXFER-ENDING
~P422&P411:	[467],[459],[458],RESRP[428],[440]	=>Q-WRT-ENDING
~P422&~P411:	[467],[459],[458],RESRP[428],[440]	=>Q-WRITE
28:	[470]	=>Q-READ
29:	[453],[47]	=>Q-RES-SYMCF-PD
30:	[453],[47]	=>Q-CAN-SYMCF-PD
31:	[465],[458],P-SYMRP	=>Q-WRITE
32:	[453],[47]	=>Q-RES-SYMIN-PD
33:	[453],[47]	=>Q-CAN-SYMIN-PD
34:	[465],P-SYMRP,[453],RESRQ[426]	=>Q-WRESTART-PD
35:	[465],P-SYMRP,[453],CANRQ[419]	=>Q-CANCEL-PD
36:	[470],I-REAIN,[464],[409]	
P14:	L-ERRABT	=>samestate
~P14:	P-SYMRQ	=>Q-READ-SYMCF-PD
37:	I-WRTIN,[458],[410]	=>Q-WRT-SYMIN-PD

38: P429:	I-CHKCF[452]	=>samestate
~P429:	L-ERRABT	=>samestate
39:	[453],[47]	=>Q-RESTART-CAN-PD
40: ~P413&P414:	[449],CANRP[419],[467],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
P413&~P414:	[449],CANRP[418],[466],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
41:		=>samestate
42: ~P409:	L-ERRABT	=>samestate
P409:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate
43: P423:	TRERP[402],[435]	=>samestate
~P423:	TRERP[402],[435],[429],[406],[471],	
P410:		=>Q-DXFRIDLE
~P410&P409&~P403:	[44],[457],[470],[412],[411]	=>Q-READ
~P410&P409&P403:	[44]	=>Q-READ
~P410&P424&~P411&P416:	[44]	=>Q-WRT-ENDING
~P410&P424&P411&P416:	[44]	=>Q-WRITE
~P410&~P416:	[44]	=>Q-I-WXFER-ENDING
44: ~P413&P414:	[449],CANRP[419],[469],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
~P414&P413:	[449],CANRP[418],[468],[21],I-CANCF[461],[440]	=>Q-DXFRIDLE
45:	[449],[466],[459],[470],RESRP[425],I-RESCF[446],[440],[21],	
P422:		=>Q-I-RXFER-ENDING
~P408:		=>Q-READ-ENDING
P408&~P422:		=>Q-READ
46:	[449],[467],[459],[458],RESRP[428],I-RESCF[446],[440],[21],	
P422:		=>Q-I-WXFER-ENDING
~P411:		=>Q-WRT-ENDING
P411&~P422:		=>Q-WRITE
47:	[401],	
~P406:	I-REAIN	=>samestate
P406:	L-ERRABT	=>samestate
48:	[401],	
~P406:	I-WRTIN	=>samestate
P406:	L-ERRABT	=>samestate
49: ~P424:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DATIN	=>samestate
50: ~P424:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DAEIN,[414]	=>samestate
51: P8|~P409:	L-ERRABT	=>samestate
~P8&P409:	P-DATRQ[18]	=>samestate
52: ~P409:	L-ERRABT	=>samestate
P409:	DAERQ[2],[407],[471],	
P404&~P403:	[44],[412],[411],[457],[470]	=>samestate
P404&P403:	[44]	=>samestate
P405:	[44],[410]	=>samestate

~P404&~P405:	[447]	=>samestate
54: ~P409|~P424:	L-ERRABT	=>samestate
P409:	[453],[460],CANRQ[418]	=>Q-CANCEL-PD
P424:	[453],[460],CANRQ[419]	=>Q-CANCEL-PD
59: ~P409|~P424:	L-ERRABT	=>samestate
P409:	[453],RESRQ[427]	=>Q-RRESTART-PD
P424:	[453],RESRQ[426]	=>Q-WRESTART-PD
60: P409:	[453],RESRQ[427]	=>Q-RRESTART-PD
~P409:	L-ERRABT	=>samestate
62: ~P409|~P424:	L-ERRABT	=>samestate
P409:	[449],I-RESIN[446]	=>Q-I-RRESTART-PD
P424:	[449],[460],I-RESIN[446]	=>Q-I-WRESTART-PD
63: P434:	[401],[410],[412],[411],[457],[458],	
~P434:	[401],	
~P406:	I-WRTIN	=>samestate
P406:	L-ERRABT	=>samestate
64:	[401],	
P434&~P406:	[409],[44],[412],[411],[457],[470],I-REAIN	=>samestate
~P434&~P406:	I-REAIN	=>samestate
P406:	L-ERRABT	=>samestate
65: P434:	[401],[410],[44],[412],[411],[457],[458],	
~P434:	[401],	
~P406:	I-WRTIN	=>samestate
P406:	L-ERRABT	=>samestate

A.4.7.5 Concurrent access - Responder

A.4.7.5.1 Concurrent read access - Responder

STATE / EVENT	K-R-DXFRIDLE	K-R-DXFRIDLE-REC	K-R-READ-SYMCF-PD	K-R-RESS-SYMCF-PD	K-R-RCAN-SYMCF-PD	K-READ	K-READ-ENDING	K-I-RXFER-ENDING	K-RCANCEL-PD	K-I-RCANCEL-PD	K-RRESTART-PD	K-I-RRESTART-PD	K-RRESTART-CAN-PD
REARQ	1	36	47			47	64	64					
I-DATRQ			4			4	51	51					
I-DAERQ			5			5	52	52					
TRERQ-R							9	9	41				
I-TRERP-R								11					
I-CANRQ-R			30	30		12		54			39	12	
CANRP-R									13				
CANRQ-R			14	14		14	14		44		14	14	40
I-CANRP-R										15			
I-CHKRQ			16			16	42	42					
P-SYMCF			28	20	12	18	38	38	41				
I-RESRQ-R			29			20	60	60					
RESRP-R											22		54
RESRQ-R			24			24	62	62			45		
I-RESRP-R												26	

A.4.7.5.2 Detailed entries - concurrent read access

1:	P440&(~P401|~P407):	L-ERRABT	=>samestate
	P440&P401&P407&P437:	[420],[401],[409],[44],[412],[413],I-REAIN,[464],	
	P440&P401&P407&~P437:	[401],[409],[44],[412],I-REAIN,[464],	
	P14:	L-ERRABT	=>samestate
	~P14:	P-SYMRQ,[457],[470]	=>K-READ-SYMCF-PD
4:	~P8:	P-DATRQ[18]	=>samestate
	P8:	L-ERRABT	=>samestate
5:		DAERQ[2],[407],[471],	
	~P404:	[447]	=>K-I-READ-ENDING
	P404&~P403:	[44],[412],[411],[409],[457],[470]	=>K-I-READ-ENDING
	P404&P403:	[44],[409]	=>K-I-READ-ENDING

9: P15:		=>samestate
~P15	[408],I-TRERQ-R,[434],[431],	
P404&~P403:	[412],[411],[457],[470],	=>K-I-RXFER-ENDING
11 P423:	TRERP-R[402],	
~P423:	TRERP-R[402],[435],[429],[406],	
P410&P416:		=>K-R-DXFRIDLE
P410&~P416:		=>samestate
~P410&P409&P408&P416:		=>K-READ
~P410&P409&~P408&P416:		=>K-READ-ENDING
~P410&~P416:		=>K-I-RXFER-ENDING
12:	[453],CANRQ-R[418]	=>K-RCANCEL-PD
13:	[449],[468],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
14:	[449],[468],[453],I-CANIN-R[461]	=>K-I-RCANCEL-PD
15:	[21],CANRP-R[418],[468],[440]	=>K-R-DXFRIDLE
16:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate
18:	I-CHKCF[452]	=>samestate
20:	[453],RESRQ-R[427]	=>K-RRESTART-PD
22:	[449],[466],[459],[470],[21],	
P422:	I-RESCF-R[446]	=>K-RXFER-ENDING
~P422:	I-RESCF-R[446]	=>K-READ
24:	[449],I-RESIN-R[446]	=>K-I-RRESTART-PD
26: P422:	[466],[459],[470],RESRP-R[425],[440]	=>K-I-RXFER-ENDING
~P422&P408:	[466],[459],[470],RESRP-R[425],[440]	=>K-READ-ENDING
~P422&~P408:	[466],[459],[470],RESRP-R[425],[440]	=>K-READ
28:	[470]	=>K-READ
29:	[453],[47]	=>K-RRES-SYMCF-PD
30:	[453],[47]	=>K-RCAN-SYMCF-PD
36:	[470],I-REAIN,[464],[409],	
P14:	L-ERRABT	=>samestate
~P14:	P-SYMRQ	=>K-REA-SYMCF-PD
38: P429:	I-CHKCF[452]	=>samestate
~P429:	L-ERRABT	=>samestate
39:	[453],[47]	=>K-RRESTART-CAN-PD
40:	[449],CANRP-R[418],[466],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
41:		=>samestate
42: ~P409:	L-ERRABT	=>samestate
P409:	[25],[454],[464],	
~P14:	P-SYMRQ[445]	=>samestate
P14:	L-ERRABT	=>samestate

44:	[449],CANRP-R[418],[468],[21],I-CANCF-R[461],[440]	=>K-R-DXFRIDLE
45:	[449],[466],[459],[470],[2], RESRP-R[425],I-RESCF-R[446],[440],	
P422:		=>K-I-RXFER-ENDING
~P408:		=>K-READ-ENDING
P408&~P422:		=>K-READ
47:	[401],	
~P406:	I-REAIN	=>samestate
P406:	L-ERRABT	=>samestate
51: P8|~P409:	L-ERRABT	=>samestate
~P8&P409:	P-DATRQ[18]	=>samestate
52: ~P409:	L-ERRABT	=>samestate
P409:	DAERQ[2],[407],[471],	
P404&~P403:	[44],[412],[411],[457],[470]	=>samestate
P404&P403:	[44]	=>samestate
~P404:	[447]	=>samestate
54: P409:	[453],CANRQ-R[418]	=>K-RCANCEL-PD
~P409:	L-ERRABT	=>samestate
60: P409:	[453],RESRQ-R[427]	=>K-RRESTART-PD
~P409:	L-ERRABT	=>samestate
62: ~P409:	L-ERRABT	=>samestate
P409:	[449],I-RESIN-R[446]	=>K-I-RRESTART-PD
64: P434:	[401],[409],[44],[412],[411],[457],[470],	
~P434:	[401],	
~P406:	I-REAIN	=>samestate
P406:	L-ERRABT	=>samestate

A.4.7.5.3 Concurrent write access - Responder

STATE / EVENT	K-W-DXFRIDLE	K-W-DXFRIDLE-REC	K-WRT-SYMIN-PD	K-WRES-SYMIN-PD	K-WCAN-SYMIN-PD	K-WRITE	K-WRT-ENDING	K-I-WXFER-ENDING	K-WCANCEL-PD	K-I-WCANCEL-PD	K-WRESTART-PD	K-I-WRESTART-PD	K-WRESTART-CAN-PD
WRTRQ	2	37	48			48	48	65					
DATRQ						6		49	41		41		
DAERQ						8		50	41		41		
TRERQ-W							10		41		41		
I-TRERP-W								43					
I-CANRQ-W			33	33		7	7	7			39	7	
CANRP-W									13				
CANRQ-W						14			44		14	14	40
I-CANRP-W										15			
P-SYMIN			31	34	35	17			41				
I-CHKRP						19	19						
I-RESRQ-W			32			21	21	59					
RESRP-W											23		54
RESRQ-W						25		25			46		
I-RESRP-W												27	

A.4.7.5.4 Detailed entries - concurrent write access

2:	P440&(~P401|~P407):	L-ERRABT	=>samestate
	P440&P401&P407&P437:	[420],[401],[410],[44],[412],[413],[457],[458],I-WRTIN	=>K-WRT-SYMIN-PD
	P440&P401&P407&~P437:	[401],[410],[44],[412],[457],[458],I-WRTIN	=>K-WRT-SYMIN-PD
6:	P15:		=>samestate
	~P15:	I-DATIN	=>samestate
7:		[453],[460],CANRQ-W[419]	=>K-WCANCEL-PD
8:	P15:		=>samestate
	~P15:	I-DAEIN,[414]	=>K-WRT-ENDING
10:	P15:		=>samestate
	~P15:	[415],I-TREIN-W,[434],[431],	
	P405&P403:	[410]	=>K-I-WXFER-ENDING
	P405&~P403:	[410],[412],[411],[457],[458]	=>K-I-WXFER-ENDING

~P405:	[447]	=>K-I-WXFER-ENDING
13:	[449],[469],[21],I-CANCF-W[461],[440]	=>K-W-DXFRIDLE
14:	[449],[469],[453],[460],I-CANIN-W[461]	=>K-I-WCANCEL-PD
15:	[21],CANRP-W[419],[469],[440]	=>K-W-DXFRIDLE
17: ~P27&P402:	I-CHKIN[403],[465],[454]	=>samestate
P27|~P402:	L-ERRABT	=>samestate
19:	P-SYMRP[451][445]	=>samestate
21:	[453],[460],RESRQ-W[426]	=>K-WRESTART-PD
23:	[449],[467],[459],[458],[21],	
P422:	I-RESCF-W[446]	=>K-I-WXFER-ENDING
~P422:	I-RESCF-W[446]	=>K-WRITE
25:	[449],[460],I-RESIN-W[446]	=>K-I-WRESTART-PD
27: P422:	[467],[459],[458],RESRP-W[428],[440]	=>K-I-WXFER-ENDING
~P422&P411:	[467],[459],[458],RESRP-W[428],[440]	=>K-WRT-ENDING
~P422&~P411:	[467],[459],[458],RESRP-W[428],[440]	=>K-WRITE
31:	[465],[458],P-SYMRP	=>K-WRITE
32:	[453],[47]	=>K-WRES-SYMIN-PD
33:	[453],[47]	=>K-WCAN-SYMIN-PD
34:	[465],P-SYMRP,[453],RESRQ-W[426]	=>K-WRESTART-PD
35:	[465],P-SYMRP,[453],CANRQ-W[419]	=>K-WCANCEL-PD
37:	I-WRTIN,[458],[410]	=>K-WRT-SYMIN-PD
39:	[453],[47]	=>K-WRESTART-CAN-PD
40:	[449],CANRP-W[419],[467],[21],I-CANCF-W[461],[440]	=>K-W-DXFRIDLE
41:		=>samestate
43: P423:	TRERP-W[402]	=>samestate
~P423:	TRERP-W[402],[429],[406],[471],	
P410:		=>K-W-DXFRIDLE
~P410&P424&~P411&P416:	[44]	=>K-WRT-ENDING
~P410&P424&P411&P416:	[44]	=>K-WRITE
~P410&~P416:	[44]	=>K-I-WXFER-ENDING
44:	[449],CANRP-W[419],[469],[21],I-CANCF-W[461],[440]	=>K-W-DXFRIDLE
46:	[449],[467],[459],[458],[21], RESRP-W[428],I-RESCF-W[446],[440],	
P422:		=>K-I-WXFER-ENDING
~P411:		=>K-WRT-ENDING
P411&~P422:		=>K-WRITE
48:	[401],	
~P406:	I-WRTIN	=>samestate
P406:	L-ERRABT	=>samestate
49: ~P424:	L-ERRABT	=>samestate

P15:		=>samestate
~P15:	I-DATIN	=>samestate
50: ~P424:	L-ERRABT	=>samestate
P15:		=>samestate
~P15:	I-DAEIN,[414]	=>samestate
54: ~P424:	L-ERRABT	=>samestate
P424:	[453],[460],CANRQ-W[419]	=>K-WCANCEL-PD
59: ~P424:	L-ERRABT	=>samestate
P424:	[453],[460],RESRQ-W[426]	=>K-WRESTART-PD
65:		
P434:	[401],[410],[44],[412],[411],[457],[458],	
~P434:	[401],	
P406:	L-ERRABT	=>samestate
P~406:	I-WRTIN	=>samestate

A.5 File error recovery protocol machine (FERPM)

A.5.1 States - file error recovery

Relabel A.5.1:

A.5.1.1 Normal access (non-overlapped)

Add the following sub-clauses.

A.5.1.2 Consecutive access

Q-PASSIVE	The basic file protocol is currently in operation but no transfer of file contents is in progress. It is used when the FERPM becomes inactive in normal error free activity, or when it cannot provide its services any more after issuing an L-ERRABT signal.
Q-XFER-IDLE	Data transfer idle.
Q-XFER	Normal transfer of file contents has been established.
Q-RE-ISSUE	After a CANCEL, RECOVER or RESTART, the initiating FERPM reissues TRANSFER-END requests and TRANSFER requests while the responding FERPM reissues TRANSFER-END responses
Q-RESTART-PD	Restart pending, wait for an I-RESCF event as an F-RESTART confirm primitive from the IFS.
Q-RESTART	A restart of a data transfer is in progress. A checkpoint identifier has been negotiated, and data following that checkpoint, but prior to the error, is being retransmitted.
Q-CANCEL-PD	Cancel pending, wait for an I-CANCF event as an F-CANCEL confirm primitive from the IFS.
CLOSE-EX	Close expected, wait for an I-CLOIN event as an F-CLOSE indication primitive from the IFS.
DESELECT-EX	Deselect expected, wait for an I-DESIN event as an F-DESELECT indication primitive from the IFS.
RECOVER-PD	Recover pending, wait for an I-RECCF event as an F-RECOVER confirm primitive from the IFS.
SEL-PD	SELECT pending, wait for an I-SELCF event as an F-SELECT confirm primitive from the IFS.
OPN-PD	OPEN pending, wait for an I-OPNCF event as an F-OPEN indication primitive form the IFS.
SEL-EX	SELECT expected, wait for an I-SELIN event as an F-SELECT indication primitive from the IFS.
OPN-EX	OPEN expected, wait for an I-OPNIN event as an F-OPEN indication primitive form the IFS.
XFER-EX	Data transfer expected, wait for an I-REAIN/I-WRITIN as an F-READ/F-WRITE indication primitive from the IFS.
INIT-EX	Initiation expected, wait for an I-ININ event as an F-INITIALIZE indication primitive from the IFS.
CLOSE-PD	Close pending, wait for an I-CLOCF event as an F-CLOSE confirm primitive from the IFS.
DESELECT-PD	Deselect pending, wait for an I-DESCF event as an F-DESELECT confirm primitive from the IFS.
INIT-PD	Initialize pending, wait for an I-INICF event as an F-INITIALIZE confirm primitive from the IFS.

A.5.1.3 Concurrent access

R-PASSIVE	The basic file protocol is currently in operation but no transfer of file contents is in progress. It is used when the Read sub-entity of FERPM becomes inactive in normal error free activity, or when it cannot

	provide its services any more after issuing an L-ERRABT signal.
R-XFER-IDLE	Data transfer idle for the Read sub-entity of FERPM.
R-XFER	Normal transfer of file contents has been established for the Read sub-entity of FERPM.
R-REISSUE	After a CANCEL, RECOVER or RESTART, the initiating read sub-entity of FERPM reissues TRANSFER-END requests and TRANSFER requests while the responding read sub-entity of FERPM reissues TRANSFER-END responses
R-RESTART-PD	Restart pending, wait for an I-RESCF event as an F-RESTART confirm primitive from the IFS by the read sub-entity.
R-RESTART	A restart of a data transfer is in progress. A checkpoint identifier has been negotiated, and data following that checkpoint, but prior to the error, is being retransmitted for the read sub-entity.
R-CANCEL-PD	Cancel pending, wait for an I-CANCF event as an F-CANCEL confirm primitive from the IFS by the read sub-entity of FERPM.
R-RECOVER-WT	Recovery by the write sub-entity is in progress. The read sub-entity is waiting for the L-RECVR signal from the write sub-entity.
W-PASSIVE	The basic file protocol is currently in operation but no transfer of file contents is in progress. It is used when the Write sub-entity of FERPM becomes inactive in normal error free activity, or when it cannot provide its services any more after issuing an L-ERRABT signal.
W-XFER-IDLE	Data transfer idle for the Write sub-entity of FERPM.
W-XFER	Normal transfer of file contents has been established for the Write sub-entity of FERPM.
W-REISSUE	After a CANCEL, RECOVER or RESTART, the initiating Write sub-entity of FERPM reissues TRANSFER-END requests and TRANSFER requests while the responding Write sub-entity of FERPM reissues TRANSFER-END responses
W-CREISSUE	Cancel pending while re-issuing transfer requests and transfer end requests by the write sub-entity of FERPM.
W-RECOVER-WT	Recovery by the read sub-entity is in progress. The write sub-entity is waiting for the L-RECVR signal from the read sub-entity.
W-RESTART-PD	Restart pending, wait for an I-RESCF event as an F-RESTART confirm primitive from the IFS by the Write sub-entity.
W-RESTART	A restart of a data transfer is in progress. A checkpoint identifier has been negotiated, and data following that checkpoint, but prior to the error, is being retransmitted for the Write sub-entity.
W-CANCEL-PD	Cancel pending, wait for an I-CANCF event as an F-CANCEL confirm primitive from the IFS by the Write sub-entity of FERPM.
CLOSE-EX	Close expected, wait for an I-CLOIN event as an F-CLOSE indication primitive from the IFS by either sub-entity.
DESELECT-EX	Deselect expected, wait for an I-DESIN event as an F-DESELECT indication primitive from the IFS by either sub-entity.
RECOVER-PD	Recover pending, wait for an I-RECCF event as an F-RECOVER confirm primitive from the IFS by either sub-entity.
SEL-PD	SELECT pending, wait for an I-SELCF event as an F-SELECT confirm primitive from the IFS by either sub-entity.
OPN-PD	OPEN pending, wait for an I-OPNCF event as an F-OPEN indication primitive form the IFS by either sub-entity.

SEL-EX	SELECT expected, wait for an I-SELIN event as an F-SELECT indication primitive from the IFS by either sub-entity.
OPN-EX	OPEN expected, wait for an I-OPNIN event as an F-OPEN indication primitive form the IFS by either sub-entity.
XFER-EX	Data transfer expected, wait for an I-REAIN/I-WRITIN as an F-READ/F-WRITE indication primitive from the IFS by either sub-entity.
INIT-EX	Initiation expected, wait for an I-ININ event as an F-INITIALIZE indication primitive from the IFS by either sub-entity.
CLOSE-PD	Close pending, wait for an I-CLOCF event as an F-CLOSE confirm primitive from the IFS by either sub-entity.
DESELECT-PD	Deselect pending, wait for an I-DESCF event as an F-DESELECT confirm primitive from the IFS by either sub-entity.
INIT-PD	Initialize pending, wait for an I-INICF event as an F-INITIALIZE confirm primitive from the IFS by either sub-entity.
REC-EX	Recover expected, wait for an I-RECIN event as an F-RECOVER indication primitive from the IFS by either sub-entity.

A.5.2 Incoming events - file error recovery

Add the following sub-clause.

A.5.2.1 Normal access (non-overlapped)

Relabel A.5.2.1:

A.5.2.1.1 Incoming events from the external file service user - normal access

Relabel A.5.2.2:

A.5.2.1.2 Incoming events from the internal file service - normal access

Relabel A.5.2.3:

A.5.2.1.3 Incoming events from the Local System Environment

Add the following sub-clauses.

A.5.2.2 Consecutive and concurrent access

A.5.2.2.1 Incoming events from the external file service user - access

F-OPNRQ	F-OPEN request primitive
F-CLORQ	F-CLOSE request primitive
F-REARQ	F-READ request primitive
F-WRTRQ	F-WRITE request primitive
F-DATRQ	F-DATA request primitive
F-DAERQ	F-DATA-END request primitive
F-TRERQ	F-TRANSFER-END request primitive
F-ANYRQ	F-INITIALIZE,F-READ-ATTRIBUTE,

	F-CHANGE-ATTRIBUTE,F-SELECT,F-CREATE,F-DESELECT, F-LOCATE,F-ERASE,F-BEGIN-GROUP,F-END-GROUP and F-U-ABORT request primitives
F-OPNRP	F-OPEN response primitive
F-CLORP	F-CLOSE response primitive
F-TRERP	F-TRANSFER-END response primitive
F-ANYRP	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP and F-END-GROUP response primitives
F-CANRQ	F-CANCEL request primitive
F-CANRP	F-CANCEL response primitive

A.5.2.2.2 Incoming events from the internal file service - access

I-OPNCF	F-OPEN confirm primitive
I-CLOCF	F-CLOSE confirm primitive
I-DATIN	F-DATA indication primitive
I-DAEIN	F-DATA-END indication primitive
I-CHKIN	F-CHECK indication primitive
I-CHKCF	F-CHECK confirm primitive
I-TRECF	F-TRANSFER-END confirm primitive
I-ANYCF	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP and F-END-GROUP confirm primitive
I-ANYIN	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP,F-END-GROUP,F-U-ABORT and F-P-ABORT indication primitives
I-REAIN	F-READ indication primitive
I-WRTIN	F-WRITE indication primitive
I-CLOIN	F-CLOSE indication primitive
I-TREIN	F-TRANSFER-END indication primitive
I-CANIN	F-CANCEL indication primitive
I-CANCF	F-CANCEL confirm primitive
I-OPNIN	F-OPEN indication primitive
I-RESIN	F-RESTART indication primitive
I-RESCF	F-RESTART confirm primitive
I-RESIN-R	F-RESTART (read) indication primitive
I-RESIN-W	F-RESTART (write) indication primitive
I-RESCF-R	F-RESTART (read) confirm primitive
I-RESCF-W	F-RESTART (write) confirm primitive
I-RECIN	F-RECOVER indication primitive
I-RECCF	F-RECOVER confirm primitive
I-RECIN-R	F-RECOVER (read) indication primitive
I-RECIN-W	F-RECOVER (write) indication primitive
I-RECCF-R	F-RECOVER (read) confirm primitive
I-RECCF-W	F-RECOVER (write) confirm primitive
I-CANIN-R	F-CANCEL(read) indication primitive
I-CANIN-W	F-CANCEL(write) indication primitive
I-CANCF-R	F-CANCEL(read) confirm primitive
I-CANCF-W	F-CANCEL(write) confirm primitive
I-TREIN-R	F-TRANSFER-END(read) indication primitive
I-TREIN-W	F-TRANSFER(write) indication primitive
I-TRECF-R	F-TRANSFER-END(read) confirm primitive
I-TRECF-W	F-TRANSFER-END(write) confirm primitive

A.5.2.2.3 Incoming events from the Local System Environment

L-TRERQ-R	Re-issued F-TRANSFER-END(read) request primitive
L-TRERQ-W	Re-issued F-TRANSFER-END(write) request primitive

L-TRERP-W Re-issued F-TRANSFER-END(write) response primitive
L-TRERP-R Re-issued F-TRANSFER-END(read) response primitive
L-WRTRQ Re-issued F-WRITE request primitive
L-REARQ Re-issued F-READ request primitive
L-ERROR1 Class 1 error (as defined in 18.1)
L-ERROR1-R Class I error (as defined in 18.1) to Read sub-entity
L-ERROR1-W Class I error (as defined in 18.1) to Write sub-entity
L-ERROR2 Class II error (as defined in 18.1)
L-ERROR2-R Class II error (as defined in 18.1) to Read sub-entity
L-ERROR2-W Class II error (as defined in 18.1) to Write sub-entity
L-ERROR3 Class III error (as defined in 18.1)
L-GIVEUP Signal to give up recovery process
L-DATRQ Re-issued F-DATA request primitive
L-CHKRQ Re-issued F-CHECK request primitive
L-DAERQ Re-issued F-DATA-END request primitive
L-EORIN Indicates that all checkpoint identifiers and "data end" markers in the docket have been used, and all data to be resent have been sent.
L-OPEN Signal to a sub-entity of successful processing of F-OPEN primitive.
L-CLOSE Signal to a sub-entity of successful processing of F-CLOSE primitive
L-RECVR Signal to a sub-entity to go into or out of the RECOVER-WT state
L-PASIVE Signal to a sub-entity to go into the PASSIVE state.

A.5.3 Outgoing events - file error recovery

Add the following sub-clause.

A.5.3.1 Normal access (non-overlapped)

Relabel A.5.3.1:

A.5.3.1.1 Outgoing events from the external file service user - normal access

Relabel A.5.3.2:

A.5.3.1.2 Outgoing events from the internal file service - normal access

Relabel A.5.3.3:

A.5.3.1.3 Outgoing events from the Local System Environment - overlapped access

Add the following sub-clauses.

A.5.3.2 Consecutive and concurrent access

A.5.3.2.1 Outgoing events from the external file service user - overlapped access

F-OPNCF F-OPEN confirm primitive
F-ANYCF F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP and F-END-GROUP confirm primitives
F-CLOCF F-CLOSE confirm primitive
F-DATIN F-DATA indication primitive
F-DAEIN F-DATE-END indication primitive
F-TRECF F-TRANSFER-END confirm primitive
F-OPNIN F-OPEN indication primitive
F-WRTIN F-WRITE indication primitive
F-CLOIN F-CLOSE indication primitive

F-REAIN	F-READ indication primitive
F-TREIN	F-TRANSFER-END indication primitive
F-ANYIN	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP,F-END-GROUP,F-U-ABORT and F-P-ABORT indication primitives
F-CANIN	F-CANCEL indication primitive
F-CANCF	F-CANCEL confirm primitive

A.5.3.2.2 Outgoing events from the internal file service - overlapped access

I-OPNRQ	F-OPEN request primitive
I-CLORQ	F-CLOSE request primitive
I-DATRQ	F-DATA request primitive
I-DAERQ	F-DATA-END request primitive
I-CHKRQ	F-CHECK request primitive
I-TRERQ	F-TRANSFER-END request primitive
I-REARQ	F-READ request primitive
I-WRTRQ	F-WRITE request primitive
I-ANYRQ	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP,F-END-GROUP and F-U-ABORT request primitives
I-OPNRP	F-OPEN response primitive
I-CLORP	F-CLOSE response primitive
I-TRERP	F-TRANSFER-END response primitive
I-ANYRP	F-INITIALIZE,F-READ-ATTRIBUTE,F-CHANGE-ATTRIBUTE, F-SELECT,F-CREATE,F-DESELECT,F-LOCATE,F-ERASE, F-BEGIN-GROUP and F-END-GROUP response primitives
I-CHKRP	F-CHECK response primitive
I-CANRQ	F-CANCEL request primitive
I-CANRP	F-CANCEL response primitive
I-RESRQ	F-RESTART request primitive
I-RESRP	F-RESTART response primitive
I-RESRQ-R	F-RESTART (read) request primitive
I-RESRQ-W	F-RESTART (write) request primitive
I-RESRP-R	F-RESTART (read) response primitive
I-RESRP-W	F-RESTART (write) response primitive
I-RECRQ	F-RECOVER request primitive
I-RECRP	F-RECOVER response primitive
I-RECRQ-R	F-RECOVER (read) request primitive
I-RECRQ-W	F-RECOVER (write) request primitive
I-RECRP-R	F-RECOVER (read) response primitive
I-RECRP-W	F-RECOVER (write) response primitive
I-CANRQ-R	F-CANCEL(read) request primitive
I-CANRQ-W	F-CANCEL(write) request primitive
I-CANRP-R	F-CANCEL(read) response primitive
I-CANRP-W	F-CANCEL(write) response primitive
I-TRERQ-W	F-TRANSFER-END(write) request primitive
I-TRERQ-R	F-TRANSFER-END (read) request primitive
I-TRERP-R	F-TRANSFER-END(read) response primitive
I-TRERP-W	F-TRANSFER-END(write) response primitive

A.5.3.2.3 Outgoing events from the Local System Environment - overlapped access

L-ERROR2	Signal indicating class II error
L-ERROR2-R	Signal indicating class II error to read sub-entity
L-ERROR2-W	Signal indicating class II error to write sub-entity
L-ERROR3	Signal indicating class III error
L-RESEND	Resend data request to the local system

L-STPRSD Stop resend data request to the local system
L-SUSPND Suspend issuing of F-DATRQ, F-DAERQ by the local system
L-RESUME Resume issuing of F-DATRQ, F-DAERQ by the local system
L-ERRABT signal FPM to issue an F-P-ABORT request PDU, with permanent error value
L-PABORT Signal FPM to issue a F-P-ABORT request PDU, and a F-P-ABORT indication primitive to the FERPM, both with transient error value
L-HOLD Hold primitives from external user
L-UNHOLD Process any held primitives
L-OPEN Signal to a sub-entity of successful processing of F-OPEN primitive.
L-CLOSE Signal to a sub-entity of successful processing of F-CLOSE primitive
L-RECVR Signal to a sub-entity to go into or out of the recovery-wt state
L-PASIVE Signal to a sub-entity to go into the PASSIVE state.

A.5.4 Specific actions -file error recovery

Relabel A.5.4:

A.5.4.1 Normal access (non-overlapped)

Add the following sub-clauses.

A.5.4.2 Consecutive and concurrent access

A.5.4.2.1 Normal procedure

[55] Delete the docket.

[56] Set the activity state indicator to "finished".

[57] Set the activity state indicator to "in-progress".

[58] Set the activity type indicator to "read".

[59] Set the activity type indicator to "write".

[60] Add the checkpoint identifier to the list in the docket. Increment the count of outstanding checkpoints by one.

[61] Delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued. Decrement the count of outstanding checkpoints by the number of the deleted identifiers.

[63] Set the activity state indicator to "starting".

[64] Set the activity state indicator to "data transfer finished".

[66] Use the last checkpoint identifier in the docket as the parameter

[67] Increment the checkpoint identifier count.

[68] Set the checkpoint identifier count to zero.

[69] Record the bulk data transfer number in the docket (the number is maintained by the FPM).

[70] Record which data values have been already delivered to the user.

[72] Use as parameter the checkpoint identifier.

[76] Record which data values have been sent.

[77] Create a docket and record in it the activity identifier, the information needed to issue or check an I-INIREQ (including the locations of the initiator and the responder), the recovery mode, the access context, the presentation context,

a null checkpoint list and set the activity state indicator to "starting".

[201] Include a state result indicating failure.

[204] Store "data end" marker in the docket.

[206] Delete "data end" marker in the docket.

[213] Store "transfer end" marker (reflecting information on the PDU) in the docket.

[214] Delete"transfer end" marker from docket.

[215] Clear activity type indicator.

[216] Clear the activity state indicator.

[421] Reissue TE read requests and remove any TE write requests from TE request queue (or queues).

[432] Use as parameter the TN and the TN of the last transfer end confirm (or confirms) received. (*Init read)

[441] Reissue Read requests with TN > current TN and remove any write requests from transfer request queue (or queues).

[463] Reissue TE write response and remove any TE read response from TE response queue (or queues)

[500] Increment the checkpoint identifier for the current TN

[501] Set the checkpoint identifier count for the TN received to zero.

[502] Add the checkpoint identifier to the appropriate list in the docket and increment the count of outstanding checkpoints by one for the TN received or issued.

[503] For the TN received or issued, delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued and decrement the count of outstanding checkpoints by the number of the deleted identifiers.

[504] Use the current TN and the last checkpoint identifier in the docket for that TN as the parameters.

[505] Create a docket and record in the docket the activity identifier, the information needed to issue or check an I-INIRQ (including the locations of the initiator and the the responder), the recovery mode, the access mode, the access context, the presentation context, a null checkpoint list or if the consecutive overlap access is used than a null checkpoint list for each entry in the checkpoint table and set the activity state indicator to "starting"; Or, if concurrent overlap access is used than a null checkpoint list for each entry of both the read checkpoint table and the write checkpoint table and set both activity state indicators to "starting".

[506] Record the bulk data transfer number in the docket for the TN.

[507] Use as parameter the current TN.

[508] Store "data end" marker in the docket for the current TN.

[509] Delete "data end" marker and "transfer-end" in the docket for the TN.received or issued.

[510] Add 1 to current TN;

[511] Set current TN to 0; In concurrent case, set both current TN for read (RTN) and write (WTN) to 0.

[512] Set the activity state indicator (or indicators) to "finished".

[513] Clear docket of any read 'data end' and read 'transfer end' markers if read cancel and any write 'data end' and write 'transfer end' markers if write cancel.

[514] Clear 'read' activity type indicator and set the read activity state indicator to "starting" if read cancel; or clear the

write activity type indicator and set the write state indicator to "starting" if write cancel.

[515] Store 'transfer end' marker (reflecting information on the PDU) in the docket for the TN received or issued.

[516] Set current TN to zero.

[532] Use as parameter the last transfer end confirm(s) received.

[533] Use as parameter the last transfer end indication(s) received.

[548] Clear Cancel indicators.

[550] Let the other sub-entity process the primitive or signal.

A.5.4.2.2 Error recovery

[55] Delete the docket.

[56] Set the activity state indicator to "finished".

[57] Set the activity state indicator to "in-progress".

[58] Set the activity type indicator to "read".

[59] Set the activity type indicator to "write".

[60] Add the checkpoint identifier to the list in the docket. Increment the count of outstanding checkpoints by one.

[61] Delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued.

[63] Set the activity state indicator to"starting"

[64] Set the activity state indicator to "data transfer finished".

[67] Increment the checkpoint identifier count.

[68] Set the checkpoint identifier count to zero.

[69] Record the bulk data transfer number in the docket (the number is maintained by the FPM).

[70] Record which data values have been already delivered to the user.

[72] Use as parameter the checkpoint identifier.

[73] Use as the parameter the oldest (first) checkpoint identifier in the list in the docket.

[75] Use as the diagnostics parameter "activity identifier unknown".

[76] Record which data values have been sent.

[200] Wait for 2 recommended retry time seconds.

[201] Include a state result indicating failure.

[202] Use the data value specified by the L-DATRQ signal, representing previous F-DATA request primitives, as the parameter.

[204] Store "data end" marker in the docket.

[206] Delete "data end" marker in the docket.

[207] Include an action result indicating transient error.

[209] Recreate select and open regimes.

[210] Include an action result indicating failure on next response with diagnostic of damage to open/select regime.

[211] Include a diagnostic of damage to open/select regime.

[212] Use as the parameter the bulk data transfer number recorded in the docket.

[213] Store "transfer end" marker (reflecting information on the PDU) in the docket.

[214] Delete "transfer end" marker from docket.

[215] Clear activity type indicator.

[216] Clear the activity state indicator.

[421] Reissue TE read requests and remove any TE write requests from TE request queue (or queues).

[432] Use as parameter the TN and the TN of the last transfer end confirm (or confirms) received. (*init read)

[433] Use as parameter the TN and the TN of the last transfer end request (or requests) issued. (init write)

[435] Use as parameters the TN and the TN of the last transfer end indication (or indications) received. (*resp write)

[436] Use as parameter the TN and the TN of the last transfer end response (or responses) issued.

[441] Reissue Read requests with TN > current TN and remove any write requests from transfer request queue (or queues).

[463] Reissue TE write response and remove any TE read response from TE response queue (or queues)

[500] Increment the checkpoint identifier for the current TN.

[501] Set the checkpoint identifier count for the TN received to zero.

[502] Add the checkpoint identifier to the appropriate list in the docket and increment the count of outstanding checkpoints by one for the TN received or issued.

[503] For the TN received or issued, delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued and decrement the count of outstanding checkpoints by the number of the deleted identifiers.

[504] Use the current TN and the last checkpoint identifier in the docket for the TN as the parameters.

[505] Create a docket and record in the docket the activity identifier, the information needed to issue or check an I-INIRQ (including the locations of the initiator and the the responder), the recovery mode, the access mode, the access context, the presentation context, a null checkpoint list or if the consecutive overlap access is used than a null checkpoint list for each entry in the checkpoint table and set the activity state indicator to "starting"; Or, if concurrent overlap access is used than a null checkpoint list for each entry of both the read checkpoints table and the write checkpoint table and set both activity state indicators to "starting".

[507] Use as parameter the current TN.

[508] Store 'data end' marker in the docket for the current TN.

[509] Delete 'data end' marker and 'transfer-end' marker in the docket for the TN received.

[511] Set current TN to 0; In concurrent case, set both current TN for read (RTN) and write (WTN) to 0.

[512] Set the activity state indicator (or the indicators) to "finished".

[513] Clear docket of any read 'data end' and read 'transfer end' markers if read cancel and any write 'data end' and write 'transfer end' markers if write cancel.

[514] Clear 'read' activity type indicator and the read activity state indicator if read cancel; and the write activity type indicator and the write state indicator if write cancel.

[515] Store 'transfer end' marker (reflecting information on the PDU) in the docket for the TN received.

[516] Set current TN to zero.

[520] The issuer of the I-RESRQ or I-RECRQ primitive identifies a TN and a checkpoint identifier which is
a) for the sender, the last point acknowledged.
b) for the receiver, the last point received and secured.

[521] The issuer of the I-RESRP or I-RECRP primitive identifies a TN and a checkpoint identifier which is
a) for the sender, equal to the value provided by the issuer of the request, and
b) for the receiver, the last points received and secured.

[522] Mark the TNs and the checkpoint identifiers to be reissued.

[523] Use as the parameter the TN and the checkpoint identifier of the primitive received.

[524] Use as the parameter the TN and the checkpoint identifier negotiated.

[525] Re-issue TE requests from TE request Queue (or queues).

[526] Re-issue Transfer (Read/Write) requests with TN > current TN from Transfer request queue or queues.

[527] Re-issue TE response from TE response queue (or queues).

[528] Add 1 to RSTTN.

[529] Record the TN as RSTTN.

[530] Use as the parameter the TN and the checkpoint identifier available in the docket that is both
a) after the checkpoint identifier negotiated, and
b) not yet issued during this restart of the BDT.

[531] Discard all held primitives except F-CANRQ; process F-CANRQ, if any.

[532] Use as parameter the last transfer end confirm(s) received.

[533] Use as parameter the last transfer end indication(s) received.

[534] Use RSTTN as a parameter.

[535] Set recovery wait indicator to write.

[536] Set recovery wait indicator to read

[537] Clear recovery wait indicator.

[538] For both read and write , record separately the TNs as RSTTNs and record for the initiating FERPM the RTN received (or issued) and for the responding FERPM the WTN received (or issued). And, for each receiver, delete from the docket all checkpoint identifiers smaller than the parameter in the primitive received or issued. Decrement the count of outstanding checkpoints by number of the deleted identifiers.

[539] The issuer of the I-RECRQ primitive identifies two sets of TNs and checkpoint identifiers which are:
a) for the sender, the last TN and identifier acknowledged.
b) for the receiver, the last TN and identifier received and secured.

The BTN is set to BTN = TN + SBTN.

[540] The issuer of the I-RECRP primitive identifies two sets of TNs and checkpoint identifiers which are:
a) for the sender, equal to the value provided by the issuer of the request, and
b) for the receiver, the last TN and identifier received and secured.
The BTN is set to BTN = TN + SBTN.

[541] Clear 'read' activity type indicator.

[542] Clear 'write' activity type indicator.

[543] Use the last checkpoint identifier in the docket for the received TN as the parameter.

[544] Set error3 indicator.

[545] Clear error3 indicator.

[546] Set read cancel indicator.

[547] Set write cancel indicator.

[548] Clear cancel and error3 indicators.

[550] Let the other sub-entity process the primitive or signal.

A.5.5 Predicates -file error recovery

Relabel A.5.5:

A.5.5.1 Normal access (non-overlapped)

Add the following sub-clauses.

A.5.5.2 Consecutive and concurrent access

A.5.5.2.1 Normal procedure

P23: The incoming response or confirm primitive has a state result indicating success.

P24: There are checkpoints that must be confirmed.

P29: The activity state indicator is set to "in-progress" and the activity type to "read".

P30: The activity state indicator is set to "in-progress" and the activity type to "write".

P31: The activity type indicator is set to "read".

P32: The activity type indicator is set to "write".

P33: The activity type indicator is set to "starting".

P34: The activfity state indicator is set to "finished".

P37: A checkpoint identifier is to be inserted in the data stream.

P99: The transfer service class has been negotiated.

P103: The incoming request or indication primitive has an action result indicating transient error.

P105: The activity state indicator is set to "data transfer finished".

P400: The consecutive overlap functional unit has been negotiated.

P401: The Symm Sync functional unit of Session has been negotiated.

P404: Next transfer is read.

P405: Next transfer is write.

P409: Read indicator is set.

P413: Cancel read or restart read or transfer-end read.

P414: Cancel write or restart write or transfer-end write.

P419: Complete re-issue of transfer end requests

P420: Complete re-issue of transfer end response.

P424: The write indicator is set.

P425: On the restart, recover or cancel primitive, TN confirm (or confirms) received by initiator < TN of last item in TE response queue (or queues)

P426: On the restart, recover or cancel primitive, TN indication (or indications) received by the responder < TN of last item in TE request queue (or queues)

P430: Next TE request in queue is read.

P431: There are transfer requests with TN > current TN in the transfer request queue (or queues).

P433: Complete re-issue of transfer request.

P434: There is no next transfer request.

P436: Request already in queue.

P440: The concurrent overlap functional unit has been negotiated.

P447: Both activity types are cleared

NOTE: TN represents the transfer number in consecutive overlap and the read transfer number (RTN) or the write transfer number (WTN) in concurrent overlap.

A.5.5.2.2 Error recovery

P23: The incoming response or confirm primitive has a state result indicating success.

P24: There are checkpoints that must be confirmed.

P26: An F-DATA indication primitive corresponding to the incoming I-DATIN has already been issued to the user.

P28: The data value available is after the negotiated checkpoint.

P29: The activity state indicator is set to "in-progress" and the activity type to "read".

P30: The activity state indicator is set to "in-progress" and the activity type to "write".

P31: The activity type indicator is set to "read".

P32 The activity type indicator is set to "write".

P33: The activity state indicator is set to "starting".

P34: The activity state indicator is set to "finished".

P37: A checkpoint identifier is to be inserted in the data stream.

P39: The entity has the docket for the activity identifier recovered, the current initiator matches the location, and the activity can be resumed.

P100: The recommended retry time field is present on the diagnostic parameter.

P101: The specified recovery is possible and is required.

P102: The specified association is possible.

P103: The incoming request or indication primitive has an action result indicating transient error.

P104: The "transfer end" marker is stored in the docket.

P105: The activity state indicator is set to "data transfer finished".

P106: The "data end" marker is stored in the docket.

P400: The consecutive overlap functional unit has been negotiated.

P401: The Symm Sync functional unit of Session has been negotiated.

P404: Next transfer is read.

P405: Next transfer is write.

P409: Read indicator is set.

P413: Cancel read or restart read or transfer-end read.

P414: Cancel write or restart write or transfer-end write.

P419: Complete re-issue of transfer end requests.

P420: Complete re-issue of transfer end response.

P424: The write indicator is set.

P425: On the Restart, Recover or Cancel primitive, the TN confirm (or confirms) received by initiator < the TN of last item in TE response or queue (or queues)

P426: On the Restart, Recover or Cancel primitive, the TN indication (or indications) received by the responder < the TN of last item in TE request queue (or queues)

P430: Next TE request in queue is read.

P431: There are transfer requests with TN > current TN in the transfer request queue (or queues).

P433: Complete re-issue of transfer request.

P434: There is no next transfer request.

P435: Both cancel indicators are not set.

P436: Request already in queue.

P440: The concurrent overlap functional unit has been negotiated.

P441: Current TN=RSTTN

P442: A F-CANRQ primitive is to be processed.

P443: The transfer request corresponding to RSTTN is a Read request.

P444: Recovery wait indicator is set to read.

P445: Recovery wait indicator is set to write.

P446: Error3 indicator is set.

P447: Both activity types are cleared.

P448: An F-TRANSFER-END indication primitive corresponding to the incoming I-TREIN has already been issued to the user.

A.5.6 Initiating entity state tables - FERPM - (Normal procedure)

Relabel A.5.6:

A.5.6.1 Normal access (non-overlapped) - Initiator

Relabel A.5.7:

A.5.6.2 Detailed entries - normal access (non-overlapped procedure)

Add the following sub-clauses.

A.5.6.3 Consecutive access - Initiator

STATE / EVENT	Q-PASSIVE	Q-XFER-IDLE	Q-XFER
F-OPENRQ	1		
I-OPNCF	2		
F-ANYRQ	3	3	
I-ANYCF	4	4	
F-CLORQ		5	
I-CLOCF		6	
F-REARQ		7	23
F-WRTRQ		8	24
F-DATRQ			9
I-DATIN			10
I-CHKCF			11
I-CHKIN			12
F-DAERQ			13
I-DAEIN			14
F-TRERQ			15
I-TRECF			16
F-CANRQ			17
I-CANCF			18
I-CANIN			19
F-CANRP			20

A.5.6.4 Detailed entries - consecutive access (normal procedure)

1: P400:	[511],[505],I-OPNRQ	=>samestate
2: P23:	F-OPNCF	=>Q-XFER-IDLE
~P23:	[55],F-OPNCF[201]	=>samestate
3:	I-ANYRQ	=>samestate
4:	F-ANYCF	=>samestate
5:	[56],I-CLORQ	=>samestate

6:	[55],F-CLOCF	=>Q-PASSIVE
7: P400&P33:	[57],[58],[501],I-REARQ	=>Q-XFER
P400&P34:	L-ERRABT	=>samestate
8: P400&P33:	[57],[59],[501],I-WRTRQ	=>Q-XFER
P400&P34:	L-ERRABT	=>samestate
9: P37&P400:	[500],[502],I-CHKRQ[72][507],	
[76],I-DATRQ	=>samestate	
10:	[70],F-DATIN	=>samestate
11: P400:	[503]	=>samestate
12: P400:	[502],	
P24&P400:	I-CHKRP,[503],	=>samestate
13: P400:	[508],I-DAERQ	=>samestate
14: P24&P400:	I-CHKRP[504],[503],	
P400:	[508],F-DAEIN	=>samestate
15: P434&P400:	[64],	
P400&P404:	[58],	
P400&P405:	[59],	
P400:	[515],I-TRERQ[432]	=>samestate
16: P434&P400:	[63],[215],[509],F-TRECF,[516]	=>Q-XFER-IDLE
P400:	[509],F-TRECF	=>samestate
17: P400:	I-CANRQ[532]	=>samestate
18: P400:	[215],[63],[509],F-CANCF,[516]	=>Q-XFER-IDLE
19:	F-CANIN	=>samestate
20: P400:	[215],[63],[509],I-CANRP,[532],[516]	=>Q-XFER-IDLE
23: P400:	[501],I-REARQ,	
P105:	[58],[57],	=>samestate
24: P400:	[501],I-WRTRQ,	
P105:	[59],[57],	=>samestate

A.5.6.5 Concurrent access - Initiator

A.5.6.5.1 Concurrent read access - Initiator

STATE / EVENT	R-PASSIVE	R-XFER-IDLE	R-XFER
F-OPNRQ	1		
I-OPNCF	2		
F-ANYRQ	3	3	
I-ANYCF	4	4	
F-CLORQ		5	
I-CLOCF		6	
F-REARQ		7	23
I-DATIN			10
I-CHKIN			12
I-DAEIN			14
F-TRERQ			15
F-CANRQ			17
F-CANRP			20
I-TRECF-R			25
I-CANCF-R			27
I-CANIN-R			19

A.5.6.5.2 Detailed entries - concurrent read access (normal procedure)

1: P440:	[511],[505],[548],I-OPNRQ	=>samestate
2: P23:	F-OPNCF,L-OPEN	=>R-XFER-IDLE
~P23:	[55],F-OPNCF[201]	=>samestate
3:	I-ANYRQ	=>samestate
4:	F-ANYCF	=>samestate
5: P447:	[512],I-CLORQ	=>samestate
~P447:	[55],L-ERRABT	=>samestate
6:	[55],F-CLOCF,L-CLOSE	=>R-PASSIVE
7: P440&P33:	[57],[58],[501],I-REARQ	=>R-XFER
P440&P34:	[55],L-ERRABT	=>samestate
10:	[70],F-DATIN	=>samestate

12: P440: P24&P440:	[502], I-CHKRP,[503],	=>samestate
14: P24&P440: P440:	I-CHKRP[504],[503], [508],F-DAEIN	=>samestate
15: P434&P440&P413: P440&P413:	[64], [515],I-TRERQ-R[432]	=>samestate
17: P413&P440:	I-CANRQ-R[532]	=>samestate
19:	F-CANIN	=>samestate
20: P440&P413:	[513],[514],I-CANRP-R,[532],[516]	=>R-XFER-IDLE
23: P440&~P31 P440&P31: P105:	[55],L-ERRABT [501],I-REARQ, [57], =>samestate	=>samestate
25: P434&P413: ~P434&P413:	[63],[215],[509],F-TRECF,[516] [509],F-TRECF	=>R-XFER-IDLE =>samestate
27:	[513],[514],F-CANCF,[516]	=>R-XFER-IDLE

A.5.6.5.3 Concurrent write access - Initiator

STATE / EVENT	W-PASSIVE	W-XFER-IDLE	W-XFER	W-REISSUE
F-OPNRQ	30			
I-OPNCF	30			
F-ANYRQ	30	30		
I-ANYCF	30	30		
F-CLORQ		30		
I-CLOCF		30		
F-WRTRQ		8	24	
F-DATRQ			9	
I-CHKCF			11	
F-DAERQ			13	
F-TRERQ			15	
L-TRERQ-R				21
L-REARQ				22
I-TRECF-W			25	
I-CANCF-W			28	
I-CANIN-W			19	
F-CANRQ			17	
F-CANRP			20	
L-OPEN	31			
L-CLOSE		32		

A.5.6.5.4 Detailed entries - concurrent write access (normal procedure)

8: P440&P33:	[57],[59],[501],I-WRTRQ	=>W-XFER
P440&P34:	[55],L-ERRABT	=>samestate
9: P37&P440:	[500],[502],I-CHKRQ[72][507], [76],I-DATRQ	=>samestate
11: P440:	[503]	=>samestate
13: P440:	[508],I-DAERQ	=>samestate
15: P434&P440&P414:	[64],	
P440&P414:	[515],I-TRERQ-W[432]	=>samestate

17: P414&P440:	I-CANRQ-W[532]	=>samestate
19:	F-CANIN	=>samestate
20: P440&P414:	[513],[514],I-CANRP-W[532],	
P426&P414:	[421],L-HOLD	=>W-REISSUE
~P426&P431&P414:	[441],L-HOLD	=>W-REISSUE
~P426&~P431&P414:	[516]	=>W-XFER-IDLE
21:	I-TRERQ-R,	
~P419:		=>samestate
P419&P431:	[441]	=>samestate
P419&~P431:	[516],L-UNHOLD	=>W-XFER-IDLE
22:	I-REARQ,	
~P433:		=>samestate
P433:	L-UNHOLD,[516]	=>W-XFER-IDLE
24: P440&~P32:	[55],L-ERRABT	=>samestate
P440&P32:	[501],I-WRTRQ,	
P105:	[57],	=>samestate
25: P434:	[63],[215],[509],F-TRECF,[516]	=>W-XFER-IDLE
~P434:	[509],F-TRECF	=>samestate
28:	[513],[514],F-CANCF,	
P426:	[421],L-HOLD	=>W-REISSUE
~P426&P431:	[441],L-HOLD	=>W-REISSUE
~P426&~P431:	[516]	=>W-XFER-IDLE
30:	[550]	=>samestate
31:		=>W-XFER-IDLE
32:		=>W-PASSIVE

A.5.7 Responding entity state table - FERPM - (normal procedure)

Relabel A.5.8:

A.5.7.1 Normal access (non-overlapped) - Responder

Relabel A.5.9:

A.5.7.2 Detailed entries - normal access (normal procedure)

Add the following sub-clauses.

A.5.7.3 Consecutive access - Responder

STATE / EVENT	Q-PASSIVE	Q-XFER-IDLE	Q-XFER
I-OPENIN	1		
F-OPNRP	2		
F-ANYRP	3	3	
I-ANYIN	4	4	
I-CLOIN		5	
F-CLORP		6	
I-REAIN		7	23
I-WRTIN		8	24
F-DATRQ			9
I-DATIN			10
I-CHKCF			11
I-CHKIN			12
F-DAERQ			13
I-DAEIN			14
I-TREIN			15
F-TRERP			16
F-CANRQ			17
I-CANCF			18
I-CANIN			19
F-CANRP			20

A.5.7.4 Detailed entries - consecutive access (normal procedure)

1: P400:	[511],[505],F-OPNIN	=>samestate
2: P23:	I-OPNRP	=>Q-XFER-IDLE
~P23:	[55],F-OPNRP	=>samestate
3:	I-ANYRP	=>samestate
4:	F-ANYIN	=>samestate
5: ~P103:	[56],F-CLOIN	=>samestate
6:	[55],I-CLORP	=>Q-PASSIVE
7: P400&P33:	[57],[58],[501],F-REAIN	=>Q-XFER
P400&P34:	L-ERRABT	=>samestate
8: P400&P33:	[57],[59],[501],F-WRTIN	=>Q-XFER
P400&P34:	L-ERRABT	=>samestate
9: P37&P400:	[500],[502],I-CHKRQ[72][507], [76],I-DATRQ	=>samestate
10:	[70],F-DATIN	=>samestate
11: P400:	[503]	=>samestate
12: P400:	[502],	
P24&P400:	I-CHKRP,[503],	=>samestate
13: P400:	[508],I-DAERQ	=>samestate
14: P24&P400:	I-CHKRP[504],[503],	
P400:	[508],F-DAEIN	=>samestate
15: P434&P400:	[64],	
P400&P405:	[59],	
P400:	F-TREIN	=>samestate
16: P434&P400:	[63],[215],[509],I-TRERP,[515],[516]	=>Q-XFER-IDLE
P400&P404:	[58],	
P400:	[509],I-TRERP,[515]	=>samestate
17: P400:	I-CANRQ[533]	=>samestate
18: P400:	[215],[63],[509],[512],F-CANCF,[516]	=>Q-XFER-IDLE
19:	F-CANIN	=>samestate
20: P400:	[215],[63],[509],I-CANRP[533],[516]	=>Q-XFER-IDLE
23: P400:	[501],	
~P436:	F-REAIN,	
P105:	[58],[57],	=>samestate
24: P400:	[501],	

~P436:	F-WRTIN,
P105:	[59],[57],

=>samestate

Note -The condition "P103" in entry 5 is handled in A5.1.7.1 in entry 16

A.5.7.5 Concurrent access - Responder

A.5.7.5.1 Concurrent read access - Responder

STATE / EVENT	R-PASSIVE	R-XFER-IDLE	R-XFER	R-REISSUE
I-OPENIN	1			
F-OPNRP	2			
F-ANYRP	3	3		
I-ANYIN	4	4		
I-CLOIN		5		
F-CLORP		6		
I-REAIN		7	23	
F-DATRQ			9	
I-CHKCF			11	
F-DAERQ			13	
F-TRERP			16	
F-CANRQ			17	
F-CANRP			20	
L-TRERP-W				21
I-TREIN-R			25	
I-CANCF-R			27	
I-CANIN-R			19	

A.5.7.5.2 Detailed entries - concurrent read access (normal procedure)

1: P440:	[511],[505],F-OPNIN	=>samestate
2: P23:	I-OPNRP,L-OPEN	=>R-XFER-IDLE
~P23:	[55],F-OPNRP	=>samestate
3:	I-ANYRP	=>samestate
4:	F-ANYIN	=>samestate
5: ~P103&P447:	[512],F-CLOIN	=>samestate
~P103&~P447:	[55],L-ERRABT	=>samestate
6:	[55],I-CLORP,L-CLOSE	=>R-PASSIVE
7: P440&P33:	[57],[58],[501],F-REAIN	=>R-XFER
P440&P34:	[55],L-ERRABT	=>samestate

9: P37&P440:	[500],[502],I-CHKRQ[72][507], [76],I-DATRQ	=>samestate
11: P440:	[503]	=>samestate
13: P440:	[508],I-DAERQ	=>samestate
16: P434&P440&P413:	[63],[215],[509],I-TRERP-R,[515],[516]	=>R-XFER-IDLE
~P434&P413&P440:	[509],I-TRERP-R,[515]	=>samestate
17: P413&P440:	I-CANRQ-R[533]	=>samestate
19:	F-CANIN	=>samestate
20: P440&P413:	[513],[514],-CANRP[533],	
P425&P413:	[463],L-HOLD	=>R-REISSUE
~P425&P413:	[516]	=>R-XFER-IDLE
21:	I-TRERP-W,	
~P420:		=>samestate
P420:	L-UNHOLD,[516]	=>R-XFER-IDLE
23: ~P440&~P31:	[55],L-ERRABT	=>samestate
P440&P31:	[501],	
P105:	[57],	
~P436:	F-REAIN,	=>samestate
25: P434:	[64], F-TREIN	=>samestate
27:	[513],[514],F-CANCF,	
P425:	[463],L-HOLD	=>R-REISSUE
~P425:	[516]	=>R-XFER-IDLE

Note -The condition "P103" in entry 5 is handled in A5.1.7.1 in entry 16

A.5.7.5.3 Concurrent write access - Responder

STATE / EVENT	W-PASSIVE	W-XFER-IDLE	W-XFER
I-OPENIN	30		
F-OPNRP	30		
F-ANYRP	30	30	
I-ANYIN	30	30	
I-CLOIN		30	
F-CLORP		30	
I-WRTIN		8	
I-DATIN			10
I-CHKIN			12
I-DAEIN			14
F-TRERP			16
F-CANRQ			17
F-CANRP			20
I-TREIN-W			26
I-CANCF-W			28
I-CANIN-W			19
L-OPEN	31		
L-CLOSE		32	

A.5.7.5.4 Detailed entries - concurrent write access (normal procedure)

8: P440&P33:	[57],[59],[501],F-WRTIN	=>W-XFER
P440&P34:	[55],L-ERRABT	=>samestate
10:	[70],F-DATIN	=>samestate
12: P440:	[502],	
P24&P440:	I-CHKRP,[503],	=>samestate
14: P24&P440:	I-CHKRP[504],[503],	
P440:	[508],F-DAEIN	=>samestate
16: P434&P440&P414:	[63],[215],[509],I-TRERP-W,[515],[516]	=>W-XFER-IDLE
P414&P440:	[509],I-TRERP-W,[515]	=>samestate

17: P414&P440:	I-CANRQ-W[533]	=>samestate
19:	F-CANIN	=>samestate
20: P440&P414:	[513],[514],I-CANRP[533],[516]	=>W-XFER-IDLE
24: P440&~P32:	[55],L-ERRABT	=>samestate
P440&P32:	[501],	
~P436:	F-WRTIN,	
P105:	[57],	=>samestate
26: P434:	[64],	
	[510],F-TREIN	=>samestate
28:	[513],[514],F-CANCF,	
P435:	[516]	=>XFER-IDLE
~P435:	[516]	=>samestate
30:	[550]	=>samestate
31:		=>W-XFER-IDLE
32:		=>W-PASSIVE

Note -The condition "P103" in entry 5 is handled in A5.1.7.1 in entry 16

A.5.8 Initiating entity state tables - FERPM - (Class 1 errors)

Relabel A.5.10:

A.5.8.1 Normal access (non-overlapped) - Initiator

Relabel A.5.11:

A.5.8.2 Detailed entries - normal access (class 1 errors)

Add the following sub-clauses.

A.5.8.3 Consecutive access - Initiator

STATE / EVENT	Q-XFR	Q-RESTART-PD	Q-RESTART	Q-REISSUE
L-ERROR1	1		1	
I-RESCF		2		
I-RESIN	3			
I-DATIN			4	
I-CHKCF			6	
I-DAEIN			7	
L-DATRQ			9	
L-CHKRQ			10	
L-DAERQ			11	
L-EORIN			12	
L-TRERQ				13
L-REARQ				14
L-WRTRQ				15
I-CANIN		16	16	16
F-CANRP		17	17	17

A.5.8.4 Detailed entries - consecutive access (class 1 errors)

1: P400&P101&P30 :	L-SUSPND,I-RESRQ[520][532],L-HOLD	=>Q-RESTART-PD
P400&P101&P29 :	I-RESRQ[520],L-HOLD	=>Q-RESTART-PD
P400&~P101 :	L-ERROR2	=>Q-XFER
2: P400&P29 :	[503],[529]	=>Q-RESTART

P400&P30 :	[522],[503],[529],L-RESEND[523]	=>Q-RESTART
3: P400&P29 :	I-RESRP[521],L-HOLD,[529]	=>Q-RESTART
P400&P30 :	L-HOLD,L-SUSPND,[522],[503],I-RESRP[521][532], L-RESEND[524],[529]	=>Q-RESTART
4: P400&P26 :		=>samestate
P400&~P26 :	F-DATIN,L-UNHOLD	=>Q-XFER
6: P400 :	[503]	=>samestate
7: P400&P24 :	I-CHKRP[543],[503],	
~P106 :	[508],F-DAEIN	=>Q-XFER
P106&P441&~P104 :	L-UNHOLD	=>Q-XFER
P106&~P441&~P104 :	[528]	=>samestate
P106&P441&P104 :	L-UNHOLD,I-TRERQ	=>Q-XFER
P106&~P441&P104 :	[528],I-TRERQ	=>samestate
9: P28 :	I-DATRQ[202]	=>samestate
~P28 :		=>samestate
10:	I-CHKRQ[530]	=>samestate
11: ~P441 :	[528],	
~P104 :	I-DAERQ	=>samestate
P104 :	I-DAERQ,I-TRERQ	=>samestate
12: ~P426&~P431 :	L-RESUME,L-UNHOLD	=>Q-XFER
P426 :	[525]	=>Q-RE-ISSUE
~P426&P431:	[526]	=>Q-RE-ISSUE
13:	I-TRERQ,	
~P419 :		=>samestate
P419&P431 :	[526]	=>samestate
P419&~P431 :	L-RESUME,L-UNHOLD	=>Q-XFER
14:	I-REARQ,	
~P433 :		=>samestate
P433 :	L-RESUME,L-UNHOLD	=>Q-XFER
15:	I-WRTRQ,	
~P433 :		=>samestate
P433 :	L-RESUME,L-UNHOLD	=>Q-XFER
16: P400&~P103&P30 :	L-STPRSD,	
P400&~P103&(P30|P29):	[531],	
~P442 :	F-CANIN,L-UNHOLD	=>samestate
P442 :	F-CANCF,[215],[63],[509],I-CANRP,[516],L-UNHOLD	=>Q-XFER-IDLE
17: P400 :	[215],[63],[509],I-CANRP,[516]	=>Q-XFER-IDLE

Note: Either P30 or P29 must be true for RESTART to occur.

A.5.8.5 Concurrent access - Initiator

A.5.8.5.1 Concurrent read access - Initiator

STATE / EVENT	R-XFR	R-RESTART-PD	R-RESTART
L-ERROR1-R	1		1
I-RESCF-R		2	
I-RESIN-R	3		
I-DATIN			4
I-CHKCF			6
I-DAEIN			7
I-CANIN-R		16	17
I-CANRP-R		16	17

A.5.8.5.2 Detailed entries - concurrent read access (class 1 errors)

1:	P440&P101&P29 :	I-RESRQ-R[520][532],L-HOLD	=>R-RESTART-PD
	P440&(~P101|~P29) :	L-ERROR2-R,L-ERROR2-W	=>R-XFER
2:	P440&P29 :	[503],[529]	=>R-RESTART
3:	P440&P29 :	I-RESRP-R[521][532],L-HOLD,[529]	=>R-RESTART
4:	P440&P26 :		=>samestate
	P440&~P26 :	F-DATIN,L-UNHOLD	=>R-XFER
6:	P440 :	[503]	=>samestate
7:	P440&P24 :	I-CHKRP[543],[503],	
	~P106 :	[508],F-DAEIN	=>R-XFER
	P106&~P441&~P104 :	[528]	=>samestate
	P106&P441&~P104 :	L-UNHOLD	=>R-XFER
	P106&P441&P104 :	L-UNHOLD,I-TRERQ-R	=>R-XFER
	P106&~P441&P104:	[528],I-TRERQ-R	=>samestate
16:	P440&P29:	[531],	
	~P442:	F-CANIN,L-UNHOLD	=>samestate
	P442:	F-CANCF,[215],[63],[509],I-CANRP-R[532], [516],L-UNHOLD	=>R-XFER-IDLE
17:	P440:	[215],[63],[509],I-CANRP-R[532],[516]	=>R-XFER-IDLE

A.5.8.5.3 Concurrent write access - Initiator

STATE / EVENT	W-XFR	W-RESTART-PD	W-RESTART	W-REISSUE	W-CREISSUE
L-ERROR1-W	1		1		
I-RESCF-W		2			
I-RESIN-W	3				
L-DATRQ			9		
L-CHKRQ			10		
L-DAERQ			11		
L-EORIN			12		
L-TRERQ-W				13	
L-REARQ				14	19
L-WRTRQ				15	20
I-CANIN-W		16	16	16	
F-CANRP-W		17	17	17	
L-TRERQ-R				18	21

A.5.8.5.4 Detailed entries - concurrent write access (class 1 errors)

1: P440&P101&P30 : L-SUSPND,I-RESRQ-W[520][532],L-HOLD =>W-RESTART-PD
P440&(~P101&~P30) : L-ERROR2-R,L-ERROR2-W =>W-XFER

2: P440&P30 : [522],[503],[529],L-RESEND[523] =>W-RESTART

3: P440&P30 : L-HOLD,L-SUSPND,[522],[503],
I-RESRP-W[521][532],L-RESEND[524],[529] =>W-RESTART

9: P28 : I-DATRQ[202] =>samestate
~P28 : =>samestate

10: I-CHKRQ[530] =>samestate

11: ~P441 : [528],
~P104 : I-DAERQ =>samestate
P104 : I-DAERQ,I-TRERQ-W =>samestate

12: ~P426&~P431 : L-RESUME,L-UNHOLD =>W-XFER
P426 : [525] =>W-REISSUE
~P426&P431 : [526] =>W-REISSUE

13: I-TRERQ-W,
~P419 : =>samestate

P419&P431 :	[526]	=>samestate
P419&~P431 :	L-RESUME,L-UNHOLD	=>W-XFER
14:	I-REARQ,	
~P433 :		=>samestate
P433 :	L-RESUME,L-UNHOLD	=>W-XFER
15:	I-WRTRQ,	
~P433 :		=>samestate
P433 :	L-RESUME,L-UNHOLD	=>W-XFER
16: P440&~P103 :	[531],	
~P442 :	F-CANIN,L-UNHOLD	=>samestate
P442 :	F-CANCF,[513],[514],I-CANRP-W[532],	
P426 :	[421]	=>CREISSUE
~P426&P431 :	[441]	=>CREISSUE
~P426&~P431 :	[63],[516],L-UNHOLD	=>W-XFER-IDLE
17: P440&P414 :	[513],[514],I-CANRP-W[532],	
P426 :	[421]	=>CREISSUE
~P426&P431 :	[441]	=>CREISSUE
~P426&~P431 :	[63],[516],L-UNHOLD	=>W-XFER-IDLE
18:	I-TRERQ-R,	
~P419:		=>samestate
P419&P431:	[526]	=>samestate
P419&~P431:	L-RESUME,L-UNHOLD	=>W-XFER
19:	I-REARQ,	
~P433:		=>samestate
P433:		[63],[516],L-UNHOLD =>W-XFER-IDLE
20:	I-WRTRQ,	
~P433:		=>samestate
P433:		[63],[516],L-UNHOLD =>W-XFER-IDLE
21:	I-TRERQ-R,	
~P419:		=>samestate
P419&P431:	[441]	=>samestate
P419&~P431:	[63],[516],L-UNHOLD	=>W-XFER-IDLE

A.5.9 Responding entity state table - FERPM - (class 1 errors)

Relabel A.5.12:

A.5.9.1 Normal access (non-overlapped) - Responder

Relabel A.5.13:

A.5.9.2 Detailed entries - normal access (class 1 errors)

Add the following sub-clauses.

A.5.9.3 Consecutive access - Responder

STATE / EVENT	Q-XFR	Q-RESTART-PD	Q-RESTART	Q-REISSUE
L-ERROR1	1		1	
I-RESCF		2		
I-RESIN	3			
I-DATIN			4	
I-CHKCF			6	
I-DAEIN			7	
L-DATRQ			9	
L-CHKRQ			10	
L-DAERQ			11	
L-EORIN			12	
I-TREIN			13	
L-TRERP				14
I-CANIN		15	15	15
F-CANRP		16	16	16

A.5.9.4 Detailed entries - consecutive access (class 1 errors)

1:	P400&P101&P29 :	L-SUSPND,I-RESRQ[520][533],L-HOLD	=>Q-RESTART-PD
	P400&P101&P30 :	I-RESRQ[520],L-HOLD	=>Q-RESTART-PD
	P400&~P101 :	L-ERROR2	=>Q-XFER
2:	P400&P30 :	[503],[529]	=>Q-RESTART

	P400&P29 :	[522],[503],[529],L-RESEND[523]	=>Q-RESTART
3:	P400&P30 :	I-RESRP[521][533],L-HOLD,[529]	=>Q-RESTART
	P400&P29 :	L-HOLD,L-SUSPND,[522],[503], I-RESRP[521],[533],L-RESEND[524],[529]	=>Q-RESTART
4:	P400&P26 :		=>samestate
	P400&~P26 :	F-DATIN,L-UNHOLD	=>Q-XFER
6:	P400 :	[503]	=>samestate
7:	P400&P24 :	I-CHKRP[543],[503],	
	~P106 :	[508],F-DAEIN	=>Q-XFER
	P106&P441&~P104 :	L-UNHOLD	=>Q-XFER
	P106&P441&P104 :		=>samestate
	P106&~P441 :		=>samestate
9:	P28 :	I-DATRQ[202]	=>samestate
	~P28 :		=>samestate
10:		I-CHKRQ[530]	=>samestate
11:	P441 :	I-DAERQ	=>samestate
	~P441 :	I-DAERQ,[528]	=>samestate
12:	~P425 :	L-RESUME,L-UNHOLD	=>Q-XFER
	P425:	[527]	=>Q-RE-ISSUE
13:	P104:	I-TRERP,	
	~P441:		=>samestate
	P441&P448:	L-UNHOLD	=>samestate
	P441&~P448:	F-TREIN,L-UNHOLD	=>Q-XFER
14:		I-TRERP,	
	~P420 :		=>samestate
	P420 :	L-RESUME,L-UNHOLD	=>Q-XFER
15:	P400&~P103&P29 :	L-STPRSD,	
	P400&~P103&(P30|P29):	[531],	
	~P442 :	F-CANIN,L-UNHOLD	=>samestate
	P442 :	F-CANCF,[215],[63],[509],I-CANRP,[516],L-UNHOLD	=>Q-XFER-IDLE
16:	P400 :	[215],[63],[509],I-CANRP,[516]	=>Q-XFER-IDLE

Note: Either P30 or P29 must be true for RESTART to occur.

A.5.9.5 Concurrent access - Responder

A.5.9.5.1 Concurrent read access - Responder

STATE / EVENT	R-XFR	R-RESTART-PD	R-RESTART	R-REISSUE	R-CREISSUE
L-ERROR1-R	1		1		
I-RESCF-R		2			
I-RESIN-R	3				
L-DATRQ			9		
L-CHKRQ			10		
L-DAERQ			11		
L-EORIN			12		
L-TREIN-R			13		
I-TRERP-R				14	
I-CANIN-R		15	15	15	
F-CANRP		16	16	16	
L-TRERP-W				17	18

A.5.9.5.2 Detailed entries - concurrent read access (class 1 errors)

1: P440&P101&P29 :	L-SUSPND,I-RESRQ-R[520][533],L-HOLD	=>R-RESTART-PD
P440&{~P101&~P29) :	L-ERROR2-R,L-ERROR2-W	=>R-XFER
2: P440&P29 :	[522],[503],[529],L-RESEND[523]	=>R-RESTART
3: P440&P29 :	L-HOLD,L-SUSPND,[522],[503],I-RESRP-R[521][533], L-RESEND[524],[529]	=>R-RESTART
9: P28 :	I-DATRQ[202]	=>samestate
~P28 :		=>samestate
10:	I-CHKRQ[530]	=>samestate
11: P441 :	I-DAERQ	=>samestate
~P441 :	I-DAERQ,[528]	=>samestate
12: ~P425 :	L-RESUME,L-UNHOLD	=>R-XFER
P425 :	[527]	=>R-REISSUE
13: P104:	I-TRERQ-R,	
~P441:		=>samestate
P441&P448	L-UNHOLD	=>R-XFER
P441&~P448:	F-TREIN,L-UNHOLD	=>R-XFER

14:	I-TRERP-R,	
~P420 :		=>samestate
P420 :	L-RESUME,L-UNHOLD	=>R-XFER
15: P440 :	L-STPRSD,[531],	
~P442 :	F-CANIN,L-UNHOLD	=>samestate
P442 :	F-CANCF,[513],[514],I-CANRP-R[533],	
P425 :	[463]	=>R-CREISSUE
~P425 :	[63],[516],L-UNHOLD	=>R-XFER-IDLE
16: P440&P413 :	[513],[514],I-CANRP-R[533],	
P425 :	[463]	=>R-CREISSUE
~P425 :	[63],[516],L-UNHOLD	=>R-XFER-IDLE
17:	I-TRERP-W,	
~P420 :		=>samestate
P420 :	L-RESUME,L-UNHOLD	=>R-XFER
18:	I-TRERQ-W,	
~P420:		=>samestate
P420:	[63],[516],L-UNHOLD	=>R-XFER-IDLE

A.5.9.5.3 Concurrent write access - Responder

STATE / EVENT	W-XFR	W-RESTART-PD	W-RESTART
L-ERROR1-W	1		1
I-RESCF-W		2	
I-RESIN-W	3		
I-DATIN			4
I-CHKCF			6
I-DAEIN			7
I-TREIN-W			13
I-CANIN-W		15	16
I-CANRP-W		15	16
I-CANIN-W		15	16
F-CANRP		15	16

A.5.9.5.4 Detailed entries - concurrent write access (class 1 errors)

1: P440&P101&P30 :	I-RESRQ-W[520][533],L-HOLD	=>W-RESTART-PD
P440&(~P101|~P30) :	L-ERROR2-R,L-ERROR2-W	=>W-XFER
2: P440&P30 :	[503],[529]	=>W-RESTART
3: P440&P30 :	I-RESRP[521][533],L-HOLD,[529]	=>W-RESTART
4: P440&P26 :		=>samestate
P440&~P26 :	F-DATIN,L-UNHOLD	=>W-XFER
6: P440 :	[503]	=>samestate
7: P440&P24 :	I-CHKRP[543],[503],	
~P106 :	[508],F-DAEIN	=>W-XFER
P106&~P441 :		=>samestate
P106&P441&~P104 :	L-UNHOLD	=>W-XFER
P106&P441&P104 :		=>samestate
13: P441 :	L-UNHOLD	=>W-XFER
~P441 :	[528]	=>samestate
15: P440&P30:	[531],	
~P442:	F-CANIN,L-UNHOLD	=>samestate
P442:	F-CANCF,[215],[63],[509],I-CANRP-W[533], [516],L-UNHOLD	=>W-XFER-IDLE
16: P440:	[215],[63],[509],I-CANRP-W[533],[516]	=>W-XFER-IDLE

A.5.10 Initiating entity state tables - FERPM - (Class II and III errors)

Relabel A.5.14:

A.5.10.1 Normal access (non-overlapped) - Initiator

Relabel A.5.15:

A.5.10.2 Detailed entries - normal access (class II and III errors)

Add the following sub-clauses.

STATE / EVENT	Q-XFER	Q-RESTART-PD	Q-RESTART	Q-CANCEL-PD	CLOSE-PD	DESLECT-PD	RECOVER-PD	INIT-PD	Q-PASSIVE	SEL-PD	OPN-PD	Q-XFER-IDLE
L-ERROR2	1	1	1									11
I-CANIN	2	2	2									
I-CANCF				3								
I-CLOCF					4							
I-DESCF						5						
I-RECCF							6					
L-ERROR3	7	7	7	7	7	7	7		7	7	7	7
I-PABIN	12	12	12	12	12	12	12		12	12	12	12
I-SELCF										9		
I-OPNCF											10	
I-INICF								8				

A.5.10.3 Consecutive access - Initiator

A.5.10.4 Detailed entries - consecutive access (class II and III errors)

1: P400&-P101:	L-ERROR3	=>samestate
P400&P101&P30:	L-SUSPND,I-CANRQ[207][433],L-HOLD	=>Q-CANCEL-PD
P400&P101&P29:	I-CANRQ[207][432],L-HOLD	=>Q-CANCEL-PD

2:	P103&P29&P400:	I-CANRP[432],I-CLORQ	=>CLOSE-PD
	P103&P30&P400:	L-SUSPND,I-CANRP[433],I-CLORQ	=>CLOSE-PD
3:		I-CLORQ	=>CLOSE-PD
4:		I-DESRQ	=>DESELECT-PD
5:	P400:	I-RECRQ[520][532]	=>RECOVER-PD
6:	P400&~P23&~P33&P34:	[55],F-CLOCF[211],L-ERRABT	=>Q-PASSIVE
	P400&~P23&P33&~P34:	L-HOLD,I-SELRQ	=>SEL-PD
	P400&~P23&101&~(P34|P33)-&(P30|P29):	I-RECRQ[520][532]	=>samestate
	P400&~P23&~P101&~(P34|P33)&(P30|P29):	[55],L-ERRABT	=>Q-PASSIVE
	P400&P23:	[503],[529],[209],	
	P443:	I-REARQ	=>Q-RESTART
	~P443:	[522],I-WRTRQ,L-RESEND[523]	=>Q-RESTART
7:	P400&P30:	L-SUSPND,	
	P400&P100&P101:	L-HOLD,L-PABORT,[200],I-INIRQ	=>INIT-PD
	P400&~P100&P101:	L-HOLD,L-PABORT,I-INIRQ	=>INIT-PD
	P400&~P101:	[55],L-ERRABT	=>Q-PASSIVE
8:	P23:	I-RECRQ[520][532]	=>RECOVER-PD
	~P23&P102:	[200],I-INIRQ	=>samestate
	~P23&~P102:	[55],L-ERRABT	=>Q-PASSIVE
9:	P23:	I-OPNRQ	=>OPN-PD
	~P23:	[55],L-ERRABT	=>Q-PASSIVE
10:	P23:	L-UNHOLD	=>Q-XFER-IDLE
	~P23:	[55],L-ERRABT	=>Q-PASSIVE
11:	~P101:	[55],L-ERRABT	=>Q-PASSIVE
	P101:	L-HOLD,I-CLORQ[211]	=>CLOSE-PD
12:	P400&P30:	L-SUSPND,	
	P400&P103&P100&P101:	L-HOLD,[200],I-INIRQ	=>INIT-PD
	P400&P103&~P100&P101:	L-HOLD,I-INIRQ	=>INIT-PD
	P400&(~P103|~P101):	[55],F-PABIN	=>Q-PASSIVE

NOTES

1 To avoid recovery procedures that never terminate the action in entry 6 by the condition P101 or ~P101 effectively allowing the local system to determine the number of times a recovery from any one error will be attempted.

2. The condition P101 in entries 12 and 7 includes the test when FQOS is zero but the FERPM is not null.

3. The condition ~P103 in entry 2 implies that this event is not relevant to the FERPM and that normal actions take over.

4. The predicates which model the activity state indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

A.5.10.5 Concurrent access - Initiator

STATE / EVENT	R-XFER	R-RESTART-PD	R-RESTART	R-CANCEL-PD	CLOSE-PD	DESLECT-PD	RECOVER-PD	INIT-PD	SEL-PD	OPN-PD	R-RECEIVER-WT	R-XFER-IDLE	R-PASSIVE
I-CANIN-R	2	2	2										
I-CANCF-R				3									
I-CLOCF					4						22		
I-DESCF						5					22		
I-RECCF							6				22		
L-ERROR3	7	7	7	7	7	7	7		7	7	22	7	7
I-PABIN	12	12	12	12	12	12	12		12	12	22	12	12
I-SELCF									9				
I-OPNCF										10			
I-INICF								8					
L-RECVR											23	24	
L-ERROR2-R	16	16	16									11	
L-PASIVE	20	20	20	20							20	20	

A.5.10.5.1 Concurrent read access - Initiator

A.5.10.5.2 Detailed entries - concurrent access (class II and III errors) - read

2:	P440&P103&P29:	I-CANRP-R[432],[546],	
	P435:	[536]	=>R-RECOVER-WT
	~P435&P30:	I-CLORQ	=>CLOSE-PD
	~P435&~P30:	I-CLORQ,L-RECVR	=>CLOSE-PD
3:		[546],	
	P435:	[536]	=>R-RECOVER-WT
	~P435:	I-CLORQ	=>CLOSE-PD
4:		I-DESRQ	=>DESELECT-PD
5:	P440:	I-RECRQ[539][532],[548]	=>RECOVER-PD

6:	P440&P23&P29&P445:	[538],[209],I-REARQ,L-RECVR	=>R-RESTART
	P440&P23&~P29&P445:	L-RECVR	=>R-XFER-IDLE
	P440&~P23&~P33&P34&P445:	[55],F-CLOCF[211],L-ERRABT,L-PASIVE	=>R-PASSIVE
	P440&~P23&P33&~P34:	L-HOLD,I-SELRQ	=>SEL-PD
	P440&~P23&P101&~(P34|P33)&(P30|P29):	I-RECRQ[539][532]	=>samestate
	P440&~P23&~P101&~(P34|P33)&(P30|P29)&P445:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
7:	P440&P100&P101:	L-HOLD,L-PABORT,[200],I-INIRQ	=>INIT-PD
	P440&~P100&P101:	L-HOLD,L-PABORT,I-INIRQ	=>INIT-PD
	P440&~P101:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
8.	P23:	I-RECRQ[539][532]	=>RECOVER-PD
	~P23&P102:	[200],I-INIRQ	=>samestate
	~P23&~P102&P445:	[55],L-ERRAB,L-PASIVE	=>R-PASSIVE
9:	P23:	I-OPNRQ	=>OPN-PD
	~P23&P445:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
10:	P23:	L-UNHOLD,L-RECVR	=>R-XFER-IDLE
	~P23&P445:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
11:	P447&~P101:	[55],L-ERRABT	=>R-PASSIVE
	P447&P101:	L-HOLD,I-CLORQ[211]	=>CLOSE-PD
	~P447&P101:	[536]	=>R-RECOVER-WT
12:	P440&P100&P101&P103:	L-HOLD,[200],I-INIRQ	=>INIT-PD
	P440&~P100&P101&P103:	L-HOLD,I-INIRQ	=>INIT-PD
	P440&(~P101|~P103):	[55],F-PABIN,L-PASIVE	=>R-PASSIVE
16:	P440&P101:	I-CANRQ-R[207][507][532],L-HOLD,[546]	=>R-CANCEL-PD
	P440&~P101&~P446:	L-ERROR3,[544]	=>samestate
	P440&~P101&P446:		=>samestate
20:		L-UNHOLD	=>R-PASSIVE
22;		[550]	=>samestate
23:	P444&P29:	[537],[538],I-REARQ	=>R-RESTART
	P444&~P29:	[537]	=>R-XFER-IDLE
24:			=>R-RECOVER-WT

NOTES

1 To avoid recovery procedures that never terminate the action in entry 6 by the condition P101 or ~P101 effectively allowing the local system to determine the number of times a recovery from any one error will be attempted.

2 The condition P101 in entries 12 and 7 includes the test when FQOS is zero but the FERPM is not null.

3. The condition ~P103 in entry 2 implies that this event is not relevant to the FERPM and that normal actions take over.

4. The predicates which model the activity state indicator and wait indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

A.5.10.5.3 Concurrent write access - Initiator

STATE / EVENT	W-XFER	W-RESTART-PD	W-RESTART	W-CANCEL-PD	CLOSE-PD	DESLECT-PD	RECOVER-PD	INIT-PD	SEL-PD	OPN-PD	W-RECEVER-WT	W-XFER-IDLE	W-PASSIVE
I-CANIN-W	13	13	13										
I-CANCF-W				14									
I-CLOCF					4						22		
I-DESCF						5					22		
I-RECCF							6				22		
L-ERROR3	18	18	18	18	7	7	7		7	7	22	18	18
I-PABIN	18	18	18	18	12	12	12		12	12	22	18	18
I-SELCF									9				
I-OPNCF										10			
I-INICF								8			22		
L-RECVR											15	25	
L-ERROR2-W	17	17	17									21	
L-PASIVE	19	19	19	19							19	19	

A.5.10.5.4 Detailed entries - concurrent access (class II and III errors) - write

4:		I-DESRQ	=>DESELECT-PD
5:	P440:	I-RECRQ[539][532],[548]	=>RECOVER-PD
6:	P440&P23&~P30&P444:	L-RECVR	=>W-XFER-IDLE
	P440&P23&P30&P444:	[538],[522],[209],I-WRTRQ,L-RESEND[523],L-RECVR	=>W-RESTART
	P440&~P23&~P33&P34&P444:	[55],F-CLOCF[211],L-ERRABT,L-PASIVE	=>W-PASSIVE
	P440&~P23&P33&~P34:	L-HOLD,I-SELRQ	=>SEL-PD
	P440&~P23&P101&~(P34|P33)&(P30|P29):	I-RECRQ[539][532]	=>samestate
	P440&~P23&~P101&~(P34|P33)&(P30|P29)&P444:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
7:	P440&P100&P101:	L-HOLD,L-PABORT,[200],I-INIRQ	=>INIT-PD
	P440&~P100&P101:	L-HOLD,L-PABORT,I-INIRQ	=>INIT-PD
	P440&~P101:	[55],L-ERRABT,L-PASSIVE	=>W-PASSIVE
8.	P23:	I-RECRQ[539],[532]	=>RECOVER-PD
	~P23&P102:	[200],I-INIRQ	=>samestate
	~P23&~P102&P444:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
9:	P23:	I-OPNRQ	=>OPN-PD

~P23&P444:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
10: P23:	L-UNHOLD,L-RECVR	=>W-XFER-IDLE
~P23&P444:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
12: P440&P100&P101&P103:	L-HOLD,[200],I-INIRQ	=>INIT-PD
P440&~P100&P101&P103:	L-HOLD,I-INIRQ	=>INIT-PD
P440&(~P101|~P103):	[55],F-PABIN,L-PASIVE	=>W-PASSIVE
13: P440&P103&P30:	I-CANRP-W[433],[547],L-SUSPND,	
P435:	[535]	=>W-RECOVER-WT
~P435&P29:	I-CLORQ	=>CLOSE-PD
~P435&~P29:	I-CLORQ,L-RECVR	=>CLOSE-PD
14:	[547],	
P435:	[535]	=>W-RECOVER-WT
~P435:	I-CLORQ	=>CLOSE-PD
15: P445&P30:	[537],[538],[522],I-WRTRQ,L-RESEND[523]	=>W-RESTART
P445&~P30:	[537]	=>W-XFER-IDLE
17: P440&P101&P30:	L-SUSPND,I-CANRQ-W[207][507][532],L-HOLD,[547]	=>W-CANCEL-PD
P440&~P101&P30&~P446:	L-ERROR3,[544]	=>samestate
P440&~P101&P30&P446:		=>samestate
18:	L-SUSPND,L-HOLD,[550],[535]	=>W-RECOVER-WT
19:	L-UNHOLD	=>W-PASSIVE
21: P447&~P101:		=>W-PASSIVE
P447&P101:	[535]	=>W-RECOVER-WT
~P447&P101:	[535]	=>W-RECOVER-WT
22;	[550]	=>samestate
25:		=>W-RECOVER-WT

NOTES

1 To avoid recovery procedures that never terminate the action in entry 6 by the condition P101 or ~P101 effectively allowing the local system to determine the number of times a recovery from any one error will be attempted.

2 The condition P101 in entries 12 and 7 includes the test when FQOS is zero but the FERPM is not null.

3. The condition ~P103 in entry 2 implies that this event is not relevant to the FERPM and that normal actions take over.

4. The predicates which model the activity state indicator and wait indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

A.5.11 Responding entity state table - FERPM - (class II and III errors)

Relabel A.5.16:

A.5.11.1 Normal access (non-overlapped) - Responder

Relabel A.5.17:

A.5.11.2 Detailed entries - normal access (class II and III errors)

Add the following sub-clauses.

A.5.11.3 Consecutive access - Responder

STATE / EVENT	Q-XFER	Q-RESTART-PD	Q-RESTART	Q-CANCEL-PD	CLOSE-EX	DSLECT-EX	Q-PASSIVE	XFER-EX	INIT-EX	SEL-EX	OPN-EX	Q-XFER-IDLE
L-ERROR2	1	1	1				14					
I-CANIN	2	2	2									
I-CANCF				3								
I-CLOIN					4							16
I-DESIN						5						
I-RECIN							6					
L-ERROR3	7	7	7	7	7	7	7	7		7	7	7
I-PABIN	17	17	17	17	17	17	17	17		17	17	17
L-GIVEUP							12					
I-INIIN									8			
I-REAIN								13				
I-WRTIN								9				
I-SELIN										10		
I-OPNIN											11	

A.5.11.4 Detailed entries - consecutive access (class II and III errors)

1: P400&~P101:	L-ERROR3	=>samestate
P400&P101&P30:	I-CANRQ[207][435],L-HOLD	=>Q-CANCEL-PD
P400&P101&P29:	I-SUSPND,I-CANRQ[207][436],L-HOLD	=>Q-CANCEL-PD
2: P400&P103&P30:	I-CANRP[435]	=>CLOSE-EX
P400&P103&P29:	L-SUSPND,I-CANRP[436]	=>CLOSE-EX
3:		=>CLOSE-EX
4:	I-CLORP	=>DESELECT-EX
5:	I-DESRP	=>Q-PASSIVE

6.	P400&P101&P39&~P33&(P30|P29):	[209],I-RECRP[521][533],[503],[529]	=>Q-XFER-EX
	P400&P101&P39&P33&~P31&~P32:	I-RECRP[201]	=>SEL-EX
	P400&P101&P39&~P33&~P31&~P32:	I-RECRP[201]	=>samestate
	P400&P101&~P39:	I-RECRP[75][201]	=>samestate
	P400&~P101:	I-RECRP[201]	=>samestate
7:	P400&P29:	L-SUSPND,	
	P400&P101:	L-HOLD,L-PABORT	=>INIT-EX
	P400&~P101:	[55],L-ERRABT	=>Q-PASSIVE
8:	P102:	I-INIRP	=>Q-PASSIVE
	~P102:	I-INIRP[201]	=>Q-PASSIVE
9:	P400&P30&~P33&~P34:		=>Q-RESTART
	P400&~P30|P33|P34:	[55],L-ERRABT	=>Q-PASSIVE
10:		L-SELRP	=>OPN-EX
11:		L-HOLD,I-OPNRP	=>Q-XFER-IDLE
12:		[55],L-ERRABT	=>samestate
13:	P400&P29&~P33&~P34:	L-RESEND[523]	=>Q-RESTART
	P400&~P29|P33|P34:	[55],L-ERRABT	=>Q-PASSIVE
14:	P101&P400:	[210],L-HOLD	=>samestate
	~P101&P400:	[55],L-ERRABT	=>samestate
16:	P103	I-CLORP,L-HOLD	=>DESELECT-EX
17:	P400&P103&P101&P29:	L-SUSPND,L-HOLD	=>INIT-EX
	P400&P103&P101&P30:	L-HOLD	=>INIT-EX
	P400&(~P103|~P101):	[55],F-PABIN	=>Q-PASSIVE

NOTES

1. The condition P101 in entries 7 and 17 includes the test when FQOS is zero but the FERPM is not null.

2. The condition ~P103 in entries 2 and 16 implies that this event is not relevant to the FERPM and that normal actions take over.

3. The predicates which model the activity state indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

A.5.11.5 Concurrent access - Responder

A.5.11.5.1 Concurrent read access - Responder

STATE / EVENT	R-XFER	R-RESTART-PD	R-RESTART	R-CANCEL-PD	CLOSE-EX	DESLECT-EX	REC-EX	XFER-EX	INIT-PD	SEL-PD	OPN-PD	R-XFER-IDLE	R-RECEIVER-WT	R-PASSIVE
I-CANIN-R	2	2	2											
I-CANCF-R				3										
I-CLOIN					4							16	25	
I-DESIN						5							25	
I-RECIN							6						25	
L-ERROR3	7	7	7	7	7	7	7	7		7	7	7	25	7
I-PABIN	17	17	17	17	12	12	12	12		12	12	12	25	12
L-GIVEUP							12							
I-INIIN									8					
I-REAIN								13					26	
I-SELIN										10				
I-OPNIN											11			
L-RECVR												29	28	
L-ERROR2-R	18	18	18									14		
L-PASIVE	23	23	23	23								23	23	

A.5.11.5.2 Detailed entries - concurrent access (class II and III errors) - read

2: P440&P101: I-SUSPND,I-CANRP-R[436],[546],
P435: [536] =>R-RECOVER-WT
~P435&P29: =>CLOSE-EX
~P435&~P29: L-RECVR =>CLOSE-EX

3: [546],
P435: [536] =>R-RECOVER-WT
~P435: =>CLOSE-EX

4: I-CLORP =>DESELECT-EX

5: I-DESRP =>REC-EX

6: P440&P101&P39&~P33&~P34:
[209],I-RECRP[540][533],[538],[548] =>XFER-EX
P440&P101&P39&P33: I-RECRP[201],[548] =>SEL-EX
P440&P101&P39&~P33&P34: I-RECRP[201],[548] =>samestate
P440&P101&~P39: I-RECRP[75][201],[548] =>samestate
P440&~P101: I-RECRP[201],[548] =>samestate

7: P440&P101:	L-SUSPND,L-HOLD,L-PABORT	=>INIT-EX
P440&~P101:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
8: P102:	I-INIRP	=>REC-EX
~P102:	I-INIRP[201],L-PASIVE	=>R-PASSIVE
10:	L-SELRP	=>opN-ex
11:	I-OPNRP	=>XFER-EX
12: P444:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
13: P440&P29&P445:	L-RESEND[524],L-RECVR	=>R-RESTART
P440&P29&~P445:	L-RESEND[524]	=>R-RESTART
P440&~P29&P445:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
14: P447&P101:	[210],L-HOLD	=>samestate
P447&~P101:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
~P447&P101:	[536]	=>R-RECOVER-WT
16: P103:	I-CLORP,L-HOLD	=>DESELECT-EX
17: P440&P103&P101:	L-SUSPND,L-HOLD	=>INIT-EX
P440&(~P103|~P101):	[55],F-PABIN,L-PASIVE	=>R-PASSIVE
18: P440&P101:	L-SUSPND,I-CANRQ-R[207][436],L-HOLD,[546]	=>R-CANCEL-PD
P440&~P101&~P446:	L-ERROR3,[544]	=>samestate
P440&~P101&P446:		=>samestate
25:	[550]	=>samestate
26: P440&P29&P444:	L-RESEND[524],[537]	=>R-RESTART
P440&~P29:	[55],L-ERRABT,L-PASIVE	=>R-PASSIVE
28: ~P29:	[537]	=>R-XFER-IDLE
P29:		=>samestate
29:		=>R-RECOVER-WT

NOTES

1 The condition P101 in entries 7 and 17 includes the test when FQOS is zero but the FERPM is not null.

2 The condition ~P103 in entries 2 and 16 implies that this event is not relevant to the FERPM and that normal actions take over.

3 The predicates which model the activity state indicator and wait indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

A.5.11.5.3 Concurrent write access - Responder

STATE / EVENT	W-XFER	W-RESTART-PD	W-RESTART	W-CANCEL-PD	CLOSE-EX	DESLECT-EX	REC-EX	XFER-EX	INIT-PD	SEL-PD	OPN-PD	W-XFER-IDLE	W-RECEVER-WT	W-PASSIVE
I-CANIN-W	20	20	20											
I-CANCF-W				21										
I-CLOIN					4							24	25	
I-DESIN						5							25	
I-RECIN							6						25	
L-ERROR3	24	24	24	24	7	7	7	7		7	7	24	25	24
I-PABIN	24	24	24	24	12	12	12	12		12	12	24	25	24
L-GIVEUP							12							
I-INIIN									8				25	
I-WRTIN								9					27	
I-SELIN										10				
I-OPNIN											11			
L-RECVR												30	15	
L-ERROR2-W	19	19	19									29		
L-PASIVE	22	22	22	22								22	22	

A.5.11.5.4 Detailed entries - concurrent access (class II and III errors) - write

4:	I-CLORP	=>DESELECT-EX
5:	I-DESRP	=>REC-EX
6: P440&P101&P39&~P33&~P34:	[209],I-RECRP[540][533],[538],[548]	=>XFER-EX
P440&P101&P39&P33:	I-RECRP[201],[548]	=>SEL-EX
P440&P101&P39&~P33&P34:	I-RECRP[201],[548]	=>samestate
P440&P101&~P39:	I-RECRP[75][201],[548]	=>samestate
P440&~P101:	I-RECRP[201],[548]	=>samestate
7: P440&P101:	L-HOLD,L-PABORT	=>INIT-EX
P440&~P101:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
8: P102:	I-INIRP	=>REC-EX
~P102:	I-INIRP[201],L-PASIVE	=>W-PASSIVE
9: P440&P30&P444:	L-RECVR	=>W-RESTART
P440&P30&~P444:		=>W-RESTART
P440&~P30&P444:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE

10:	L-SELRP	=>opN-ex
11:	I-OPNRP	=>XFER-EX
12: P445:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
15: ~P30:	[537]	=>W-XFER-IDLE
P30:		=>samestate
16: P103:	I-CLORP,L-HOLD	=>DESELECT-EX
19: P440&P101:	I-CANRQ-W[207][435],L-HOLD,[547]	=>W-CANCEL-PD
P440&~P101&~P446:	L-ERROR3,[544]	=>samestate
P440&~P101&P446:		=>samestate
20: P440&P103&P30:	I-CANRP-W[435],[547],	
P435:	[535]	=>W-RECOVER-WT
~P435&P29:		=>CLOSE-EX
~P435&~P29:	L-RECVR	=>CLOSE-EX
21:	[546],	
P435:	[535]	=>W-RECOVER-WT
~P435:		=>CLOSE-EX
22:	L-UNHOLD	=>W-PASSIVE
24:	L-HOLD,[550],[535]	=>W-RECOVER-WT
25:	[550]	=>samestate
27: P440&P30&P445:	[537]	=>W-RESTART
P440&~P30:	[55],L-ERRABT,L-PASIVE	=>W-PASSIVE
25: P447&~P101:		=>W-PASSIVE
P447&P101:	[535]	=>W-RECOVER-WT
~P447&P101:	[535]	=>W-RECOVER-WT
30:		=>W-RECOVER-WT

NOTES

1 The condition P101 in entries 7 and 17 includes the test when FQOS is zero but the FERPM is not null.

2 The condition ~P103 in entries 2 and 16 implies that this event is not relevant to the FERPM and that normal actions take over.

3 The predicates which model the activity state indicator and wait indicator are mutually exclusive, so that combinations of them cannot occur. These combinations are not included in the state table entries.

This page intentionally left blank

UDC 681.3 : 621.39

Descriptors: data processing, information interchange, network interconnection, open systems interconnection, communication procedure, files, transfer, access, management, protocols.

Price based on 96 pages