1394 Specification for Response Code Usage

Version 1.0
April 11, 1996

Sponsored by:
Architecture Working Group of the 1394 Trade Association

Approved for Release by:
1394 Trade Association Steering Committee

Abstract: This specification clarifies the response codes to be used by Serial Bus LINK and TRAN layer implementations. Uniform behavior of LINK and TRAN layers is intended to promote the interoperability of IEEE 1394 products manufactured by diverse vendors.

Keywords: 1394, LINK, response codes, TRAN, Specification

1394 Trade Association
3925 W. Braker Lane, Austin, TX 78759 USA
http://www.1394TA.org

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to product, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
3925 W. Braker Lane
Austin, TX 78759
USA

Table of Contents

This page intentionally left almost blank.

1. Overview

IEEE Std 1394–1995, Standard for a High Performance Serial Bus, has been ratified and published by the IEEE. Recent discussions, commencing with a presentation at the Architecture working group meeting of the 1394 Trade Association in Bothell, Washington in July, 1995, have demonstrated a need for more detailed definitions of response code usage in clause 6, "Link layer specification", and clause 7, "Transaction layer specification", with respect to the uniform use of response codes.

Action is underway to commence IEEE work on extensions and clarifications to IEEE Std 1394–1995; this work is expected to include revisions to the transaction layer specification. However, the proposed work is anticipated to take between twelve and eighteen months, commencing in January 1996. In the interim, industry consensus is needed to maintain interoperability between Serial Bus devices manufactured by different vendors.

This specification addresses the clarifications needed in clauses 6 and 7 of the Serial Bus standard. Members of the 1394 Trade Association shall implement the LINK and TRAN layers in their products in accordance with this specification.

Although the 1394 Trade Association is not an accredited standards body, its membership consists of the leading developers of Serial Bus products. This membership, for the most part, is also very active in the IEEE standards development process and intends to propose this specification for inclusion in the IEEE extensions to Serial Bus.

2. References

IEEE Std 1394–1995, Standard for a High Performance Serial Bus

3. Current ambiguities

With respect to the response codes (*rcode*) defined in IEEE Std 1394–1995, the descriptions are sufficiently terse as to leave room for potentially divergent interpretations.

4. Response codes

Clause 6.2.4.10 of IEEE Std 1394–1995, “Response code (rcode)”, defines the meaning of the *rcode* returned in a split transaction request. The table is reproduced below for convenience of reference.

Table 6–11 — Response code encoding

code	name	comment
0	resp_complete	The node has successfully completed the command.
1 — 3	reserved	
4	resp_conflict_error	A resource conflict was detected. The request may be retried.
5	resp_data_error	Hardware error, data is unavailable.
6	resp_type_error	A field in the request packet was set to an unsupported or incorrect value, or an invalid transaction was attempted (*e.g.*, a write to a read-only address).
7	resp_address_error	The destination offset in the request was set to an address not accessible in the destination node.
8 — F_{16}	reserved	

Despite the intended sufficiency of the standard, discussions at 1394 Trade Association meetings, on the various 1394 reflectors and elsewhere have made it clear that the usage of the response code requires clarification. The purpose is to assure equivalent behavior, and hence interoperability, of LINK layer, TRAN layer and application implementations from different vendors.

Response code usage by members of the 1394 Trade Association shall be as described in the clauses that follow.

4.1 resp_complete

Nodes shall respond with *resp_complete* in the circumstances described below (this is not an exhaustive list, just some examples of circumstances for which there might be confusion with other response codes):

A write transaction is received for a writeable address that contains read-only bits or fields. The transaction completes successfully and the write effects on the read-only bits are as specified in IEEE Std 1394–1995 or the document that describes the unit architecture. Generally an address is not considered writeable if all bits are read-only; see the discussion of *resp_type_error* below.

4.2 resp_conflict_error

Nodes shall respond with *resp_conflict_error* in the circumstances described below:

An otherwise valid request packet is received but the resources required to act upon the request are not available. The requester may reasonably expect the same packet to succeed at some point in the future when the resources are available. Note that the distinction between *resp_conflict_error* and *ack_busy_X*, *ack_busy_A* or *ack_busy_B* hinges upon the possibility of deadlock. The busy acknowledgments are appropriate for transient conditions of expected short duration that cannot cause a deadlock. On the other hand, *resp_conflict_error* shall be returned when an end-to-end retry is necessary to avoid the possibility of deadlock. Potential deadlocks typically arise when a request cannot be queued and blocks a node's transaction resources.

4.3 resp_data_error

Nodes shall respond with *resp_data_error* in the circumstances described below:

An otherwise valid request packet is received but there is a data CRC error for the data payload.

For read requests, an otherwise valid packet is received but a hardware error at the node prevents the return of the requested data. For example, an uncorrectable ECC error reported by the underlying medium shall be reported as *resp_data_error*.

For write or lock requests, an otherwise valid packet is received but a hardware error at the node prevents the updates indicated by the data payload from initiation or completion.

4.4 resp_type_error

Nodes shall respond with *resp_type_error* in the circumstances described below:

A request packet is received with a valid *tcode* (transaction code) value but the *extended_tcode* field value is reserved by IEEE Std 1394–1995.

NOTE: If a packet is received with a *tcode* value that is reserved by IEEE Std 1394–1995, the node shall not respond since it is not possible to determine whether the packet is a request, response, isochronous or some other packet.

A request packet is received with valid *tcode* and *extended_tcode* values, but the referenced address does not implement the indicated request. An example of this is a write request to an address that is entirely read-only (note that this is distinct from a write request that references an otherwise writeable location that contains read-only bits or fields). Another example is a transaction whose *tcode* specifies a lock operation but the destination address supports only read and write operations.

A request packet is addressed to a valid *destination_ID*, the *destination_offset* references an address implemented by the node but the alignment of the destination offset does not match the node's alignment requirements. For example, a quadlet register is implemented but cannot respond to a one byte data block request.

4.5 resp_address_error

Nodes shall respond with *resp_address_error* in the circumstances described below:

A request packet is addressed to a valid *destination_ID* but the *destination_offset* references an address that is not implemented by the node.

A block request packet is addressed to a valid *destination_ID* but the combination of the *destination_offset* and the *data_length* reference addresses some of which are not implemented by the node.