

To: T10 Technical Committee
 From: Rob Elliott, HP (elliott@hp.com)
 Date: 26 April 2007
 Subject: 07-089r1 SAS-2 Redundant primitive sequence error handling

Revision history

Revision 0 (12 March 2007) First revision
 Revision 1 (26 April 2007) Incorporated comments from April SAS protocol WG.

Related documents

sas2r08 - Serial Attached SCSI - 2 (SAS-2) revision 8

Overview

In SAS-2, redundant primitive reception was changed from “receive 3 identical primitives in a row” to “receive 3 identical primitives out of any 6 dwords” to tolerate more than a single-bit error. The 6 Gbps Decision Feedback Equalizer (DFE) receiver architecture increases the likelihood that a single-bit error on the wire will cause subsequent bits to also be decoded incorrectly, as incorrect equalization is applied to them (less and less for each subsequent bit).

The rules added are unclear on handling of (unlikely) sequences where two valid redundant primitive sequences are interleaved.

- a) A B A B A B: does this mean just A was detected, both A and B, or just B?
- b) A B C B C B C: when does the receiver decide A is no longer possible? When does it decide that B and/or C are candidates?
- c) A B C D A A: this is the worst case for detecting A. After D, is the receiver keeping track of four possible redundant primitive sequences - A, B, C, and/or D?

A valid transmitter will not send patterns like these, and bit errors are unlikely to result in them. Nevertheless, SAS-2 needs to clearly define how receivers are expected (or not) to handle these patterns. The simplest approach is once a redundant primitive is received, the redundant primitive detector uses the next 5 dwords solely for detecting that redundant primitive and does not start detecting a second redundant primitive until the 6th dword. A more complex approach tracks up to four possible primitive sequences at a time.

Similar issues arise when considering redundant primitives intermixed with triple primitives sequences, like:

- d) A B B B A A where A is redundant and B is triple. Does that detect just A, or both A and B, or just B?

If viewed as a triple primitive sequence detector running concurrently with a redundant primitive sequence detector, “both” is the simplest answer. If there is one primitive sequence detector handling both types, then detecting A only is probably easiest.

Since these patterns are not likely, SAS-2 should specify that the receiver behavior is vendor-specific. The first one must be detected; after that, any others that begin within the next 5 dwords may or may not be detected.

Suggested changes to SAS-2

7.2.4 Primitive sequences

7.2.4.1 Primitive sequences overview

Table 1 summarizes the types of primitive sequences.

Table 1 — Primitive sequences

Primitive sequence type	Number of times the transmitter transmits the primitive to transmit the primitive sequence	Number of times the receiver receives the primitive to detect the primitive sequence	Reference
Single	1	1	7.2.4.2
Repeated	1 or more	1	7.2.4.3
Continued	2 followed by SATA_CONT	1	7.2.4.4
Extended	3	1	7.2.4.5
Triple	3	3	7.2.4.6
Redundant	6	3	7.2.4.7

Any number of deletable primitives may be sent inside primitive sequences without affecting the count or breaking the consecutiveness requirements. Rate matching deletable primitives shall be sent inside primitive sequences inside of connections if rate matching is enabled (see 7.13).

7.2.4.6 Triple primitive sequence

Primitives that form triple primitive sequences (e.g., CLOSE (NORMAL)) shall be sent three times consecutively. Any number of deletable primitives may be sent inside primitive sequences as described in 7.2.4.1.

Receivers shall detect a triple primitive sequence after the identical primitive is received in three consecutive dwords. After ~~receiving~~ detecting a triple primitive sequence, a receiver shall not detect a second instance of the same triple primitive sequence until it has received three consecutive dwords that are not any of the following:

- a) the original primitive; or
- b) a deletable primitive.

Figure 148 shows examples of triple primitive sequences.

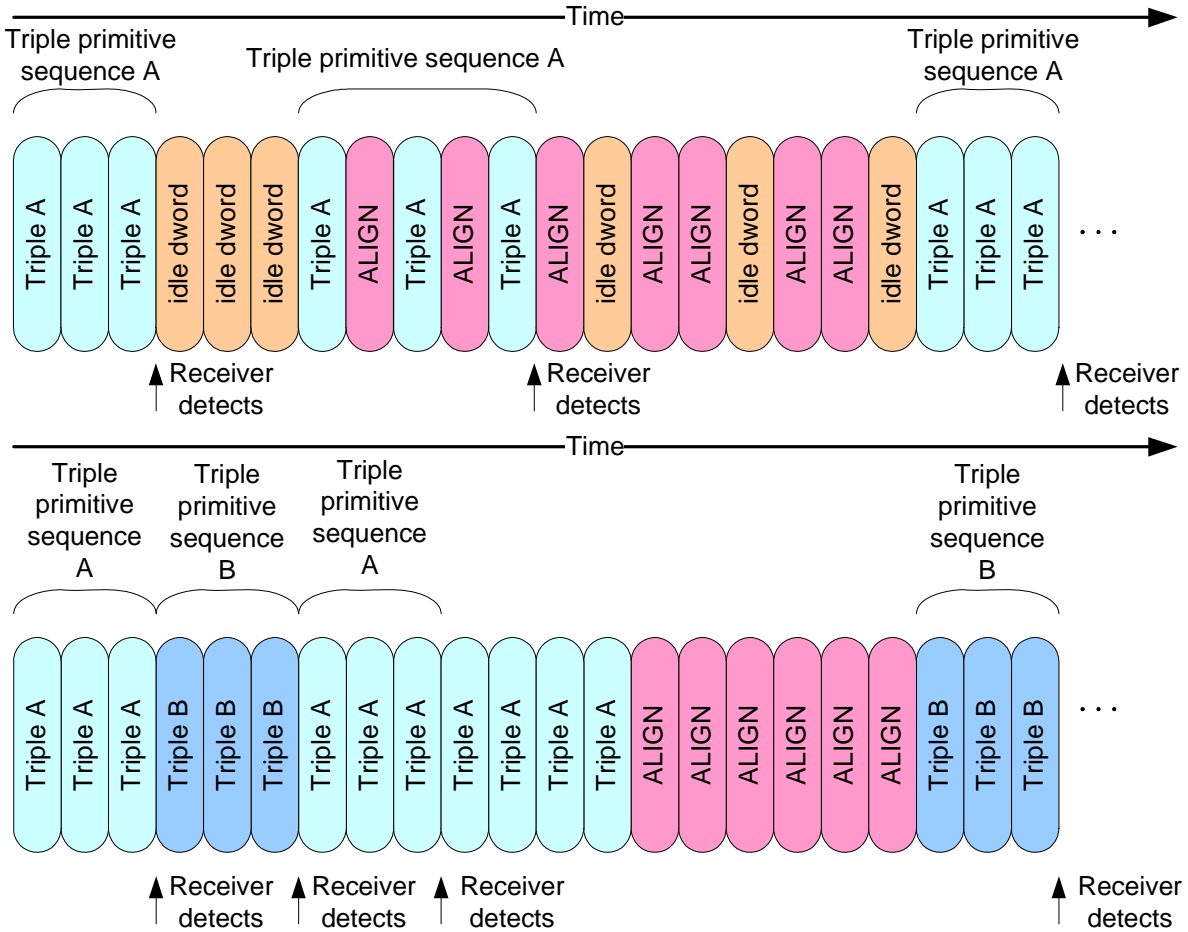


Figure 148 — Triple primitive sequences

[A triple primitive sequence that begins within three dwords after a receiver detects a redundant primitive may or may not be detected \(see 7.2.4.7\).](#)

7.2.4.7 Redundant primitive sequence

Primitives that form redundant primitive sequences (e.g., BROADCAST (CHANGE)) shall be sent six times consecutively. Any number of deletable primitives may be sent inside primitive sequences as described in 7.2.4.1.

A receiver shall detect a redundant primitive sequence after the identical primitive is received in any three of six consecutive dwords. After ~~receiving~~[detecting](#) a redundant primitive sequence, a receiver shall not detect a second instance of the same redundant primitive sequence until it has received six consecutive dwords that are not any of the following:

- a) the original primitive; or
- b) a deletable primitive.

Editor’s Note 1: It would be a change from SAS-2 to change that “six” to “three”, but that might make sense so the transmitter can send redundant A - triple - redundant B without having to add 3 fill dwords. If it does not, the receiver only starts detecting B midway, leaving it sensitive to bit errors.

After detecting a redundant primitive sequence, the receiver may or may not detect another redundant primitive sequence or a triple primitive sequence that occurs within the next three consecutive dwords.

Figure 149 shows examples of redundant and triple primitive sequences.

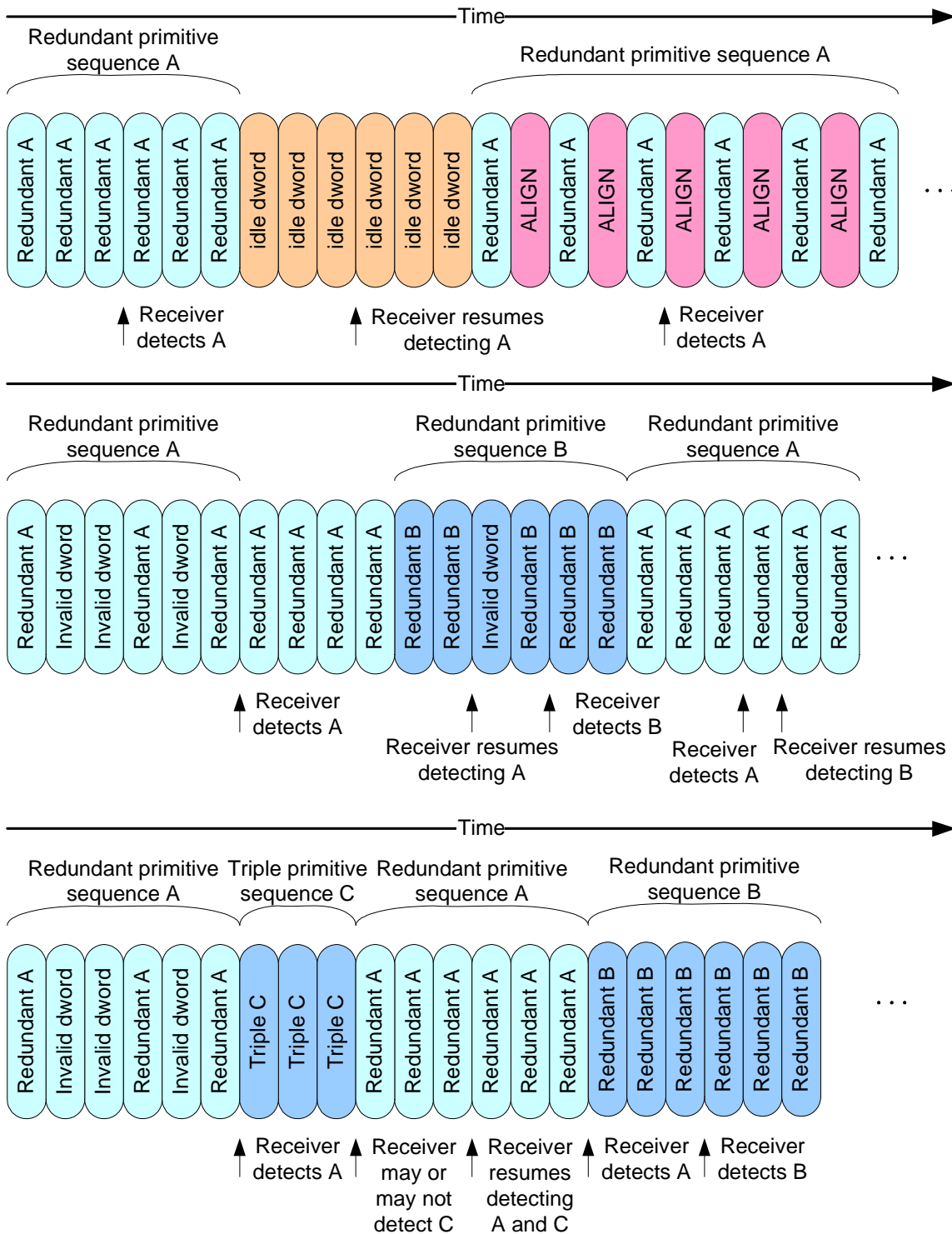


Figure 149 — Redundant and triple primitive sequences [modified]

Editor's Note 2: more invalid dwords added to the picture. 3rd row added showing interaction

[between redundant and triple.](#)
