

To: T10 Membership
From: Lawrence J. Lamers (ljlamers@ieee.org)
Subject: Extended Modes for Margining using READ/WRITE BUFFER
Date: Friday, October 16, 1998

This proposal defines additional modes to allow the READ/WRITE BUFFER commands to perform margin testing. This proposal assumes that the extension of the mode field to 4-bits as defined in 98-184 has been accepted.

Bits (7:4) of byte 1 are defined as a data pattern type field.

A mode value of (1011b) is assigned to margin mode.

The following paragraph is added to the READ BUFFER command:

Read Data margin mode (1011b)

In this mode the target transfers data to the initiator from the margin buffer. The margin buffer shall be filled with the specified data pattern. The Buffer ID and Buffer Offset fields are ignored in this mode.

The initiator compares the received data against the specified data pattern and generates an attention condition if there is a miscompare. The message sent to the target is INITIATOR DETECTED error.

The following paragraph is added to the WRITE BUFFER command:

Write data margin mode(1011b)

In this mode the target transfers data from the initiator and stores it in a margin buffer. A margin buffer is assigned in the same manner by the target as it would for a write operation. The Buffer ID and Buffer Offset fields are ignored in this mode.

The data pattern field defines the data pattern sent by the initiator.

The target compares the received data against the specified data pattern and returns CHECK CONDITION status if there is a miscompare.

READ BUFFER Command

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-------------------------------|---|---|---|------|---|---|---|
| 0 | Operation Code (3Ch) | | | | | | | |
| 1 | Data Pattern | | | | Mode | | | |
| 2 | Buffer ID | | | | | | | |
| 3 | (MSB) Buffer Offset (LSB) | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | (MSB) Allocation Length (LSB) | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | Control | | | | | | | |

WRITE BUFFER Command

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|------------------------------|---|---|---|------|---|---|---|
| 0 | Operation Code (3Bh) | | | | | | | |
| 1 | Data Pattern | | | | Mode | | | |
| 2 | Buffer ID | | | | | | | |
| 3 | (MSB) Buffer Offset (LSB) | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | (MSB) Parameter Length (LSB) | | | | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | Control | | | | | | | |

The data pattern type field is defined in table x. The data pattern repeats until the specified parameter length or allocation length is reached.

Data Pattern Field.

| Value | Description | Type | |
|----------|-----------------|---------------------------------|---------------------------------|
| 0h | Reserved | | |
| 1h | Alternating | 00h/FFh | See table x. |
| 2h | Counting | 0,1,2,3,...3,2,1,0 | See table x. |
| 3h | Oscillating | AAh/55h | See table x. |
| 4h | Walking | Walking ones then walking zeros | See table x. |
| 5h to Eh | Reserved | Reserved | |
| Fh | Vendor Specific | Vendor Specific | See technical reference manual. |

Alternating Data Pattern

| Bit Word | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Value |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000h |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FFFFh |

Counting Data Pattern

| Bit Word | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Value |
|----------|----------|---|---|----------|---|---|----------|---|---|----------|---|---|---|---|---|---|-------|
| 0 | | | | | | | | | | | | | 0 | 0 | 0 | 0 | 0000h |
| 1 | | | | | | | | | | | | | 0 | 0 | 0 | 1 | 0001h |
| 2 | | | | | | | | | | | | | 0 | 0 | 1 | 0 | 0002h |
| 3 | | | | | | | | | | | | | 0 | 0 | 1 | 1 | 0003h |
| 4 | | | | | | | | | | | | | 0 | 1 | 0 | 0 | 0004h |
| 5 | | | | | | | | | | | | | 0 | 1 | 0 | 1 | 0005h |
| 6 | | | | | | | | | | | | | 0 | 1 | 1 | 0 | 0006h |
| 7 | | | | | | | | | | | | | 0 | 1 | 1 | 1 | 0007h |
| 8 | | | | | | | | | | | | | 1 | 0 | 0 | 0 | 0008h |
| 9 | | | | | | | | | | | | | 1 | 0 | 0 | 1 | 0009h |
| 10 | | | | | | | | | | | | | 1 | 0 | 1 | 0 | 000Ah |
| 11 | | | | | | | | | | | | | 1 | 0 | 1 | 1 | 000Bh |
| 12 | | | | | | | | | | | | | 1 | 1 | 0 | 0 | 000Ch |
| 13 | | | | | | | | | | | | | 1 | 1 | 0 | 1 | 000Dh |
| 14 | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 000Eh |
| 15 | | | | | | | | | | | | | 1 | 1 | 1 | 1 | 000Fh |
| repeat | Nibble 3 | | | Nibble 2 | | | Nibble 1 | | | Nibble 0 | | | | | | | |

The counting pattern counts up then down. The total pattern is 131,070 bytes long. Nibble 0 repeats in nibble 1, then nibble 2, then nibble 3, and then decrements.

Oscillating Data Pattern

| Bit Word | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Value |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | AAAAh |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 5555h |

Walking Data Pattern

| Bit Word | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Value |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0001h |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0002h |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0004h |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0008h |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0010h |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0020h |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0040h |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0080h |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0100h |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0200h |
| 10 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0400h |
| 11 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0800h |
| 12 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000h |
| 13 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000h |
| 14 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4000h |
| 15 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8000h |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | FFFEh |
| 17 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | FFFDh |
| 18 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | FFFBh |
| 19 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | FFF7h |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | FFEFh |
| 21 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | FFDFh |
| 22 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | FFBFh |
| 23 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FF7Fh |
| 24 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FEFFh |
| 25 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FDFFh |
| 26 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FBFFh |
| 27 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | F7FFh |
| 28 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | EFFFh |
| 29 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | DFFFh |
| 30 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | BFFFh |
| 31 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7FFFh |