Checking the OpenLCB Memory Configuration Standard

October 15, 2024

1 Introduction

This note documents the procedure for checking an OpenLCB implementation against the Memory Configuration Standard.

The checks are traceable to specific sections of the Standard.

The checking assumes that the Device Being Checked (DBC) is being exercised by other nodes on the message network, e.g. is responding to enquiries from other parts of the message network.

2 Memory Configuration Procedure

A node which does not self-identify in PIP that it supports Memory Configuration will be considered to have passed these checks.

This plan does not check:

1. Stream-based memory configuration operations
2. Write and Write Under Mask to configuration memory. There's no generally-compatible way to do that. You can't even assume that a read / write / read sequence will return you to the original state, because nodes may react when information is written to them. [1]
3. The Reinitialize/Factory Reset Command and the Get Unique ID Command because these permanently change the state of the node being checked.

2.1 Configuration Options checking

This section checks the messages and interaction in Standard sections 4.13 and 4.14.

[1] See the ACDI checks in the CDI check plan for another possibility.

The checker sends a Get Configuration Options Command datagram. It then checks:

1. That the datagram reply is received.
2. That a Get Configurations Options Reply datagram is received,
3. The reply datagram i sat least six bytes long,
4. That the reserved bits in bytes 2, 3 and 4 are zero or one as required.

2.2 Address Space Information checking

This section checks the messages and interaction in Standard sections 4.15 and 4.16.

Using the common[2] 0xFD and 0xFF memory configuration spaces, [3], the checker sends a Get Address Space Information Command datagram. It then checks:

1. That the Datagram Received OK is received,
2. That a Get Address Space Information Reply datagram is received,
3. The reply datagram is at least eight bytes long,
4. That the reply datagram refers to the same memory space as the request,
5. That the reserved bits in byte 7 are correct.
6. That if the low bit of byte 7 is 1, there's a low address present in bytes 8-11.

If the reply comes back indicating that that space is not present, a warning is issued, but that part of the check is assumed to pass.

2.3 Read Operations checking

This section checks the messages and interaction in Standard sections 4.4 and 4.5.

The checker sends Read Commands to space 0xFD, the memory configuration space.

1. A read of 64 bytes from address 0,
2. A read of 10 bytes from address 0,
3. A read of 2 bytes from address 0,

Each of these are done once with the address space in byte 1 and once with the address space in byte 6.

[2]See section 4.2

[3]Note that 0xFE, the all-memory space, is considered optional here. The 0xFF CDI space check is omitted if CDI is not found in PIP.

For each, the checker checks:

1. The Datagram Received OK has the Reply Pending bit set,
2. The reply datagram is long enough to be checked,
3. It's a Read Reply datagram,
4. The error bit is not set,
5. The space matches (in either form),
6. The starting address matches,
7. The right number of data bytes are returned.

2.4 Lock/Reserve checking

This section checks the messages and interaction in Standard section 4.17 and 4.18.

The checker requests that the node be reset/restarted then

1. Sends a Lock/Reserve Command with node id zero and checks that a Lock/Reserve Reply is received with contents zero.
2. Sends a Lock/Reserve Command with node id A and checks that a Lock/Reserve Reply is received with contents A.
3. Sends a Lock/Reserve Command with node id B and checks that a Lock/Reserve Reply is received with contents A.
4. Sends a Lock/Reserve Command with zero and checks that a Lock/Reserve Reply is received with contents zero.
5. Sends a Lock/Reserve Command with node id B and checks that a Lock/Reserve Reply is received with contents B.
6. Sends a Lock/Reserve Command with node id A and checks that a Lock/Reserve Reply is received with contents B.
7. Sends a Lock/Reserve Command with node id A and checks that a Lock/Reserve Reply is received with contents B.
8. Sends a Lock/Reserve Command with zero and checks that a Lock/Reserve Reply is received with contents zero.
9. Sends a Lock/Reserve Command with zero and checks that a Lock/Reserve Reply is received with contents zero.

2.5 Reset/Reboot checking

This section checks the message and interaction in Standard section 4.24 Reset/Reboot Command.

The checker sends a Reset/Reboot Command and then checks that a Node Initialization Complete message is received to indicate a reboot. This may or may not have been preceded by a Datagram Received OK response.

If a Node Initialization Complete message has not been received after a timeout, the checker sends an AME with the device being checked's node ID. Some virtual node systems may need this as a prompt to recreate the virtual node. The checker then waits for an AMR with the device being checked's node ID, followed by a Node Initialization Complete message from that node.