Testing hot-swap on storage devices

Quarch Technology Ltd

Andy Norrie
Mike Dearman

April 2019

Contents

Introduction

Hot-swap capability is crucial in many modern storage and networking systems. Hot-swap refers to the action of replacing a system component without shutting down or disrupting the operation of the system. Hot-plug refers more specifically to the non-disruptive addition of components to a system).

Hot-swap is a complex operation due to the high level of variability introduced by human and mechanical factors, which must be considered during the design and test phases.

Over the last decade storage interfaces have become faster, more complex and more tightly coupled to the host. This has led to a wider and more serious range of failures during hot-plug

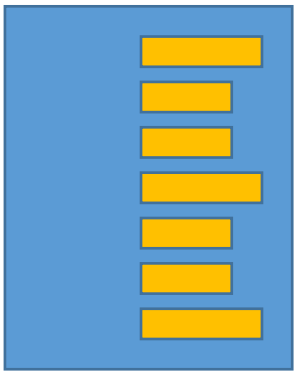
In this paper we will discuss the main causes of failure during hot-swap events, and the best practice means of ensuring that your device is hot-swap compliant in all possible scenarios.

Key Issues

Pin connection sequence

During a hot-plug operation the pins in a connector system do not all mate at the same time; microscopic differences in pin lengths and contact bounce will result in some signals connecting before others. This behavior may lead to undesirable system operation, especially where pins mate earlier or later than expected.

To reduce this problem, many connectors use multiple lengths of pin, allowing a 'planned' mating sequence to be created. This is normally used to mate the more critical pins at specific times:



Edge fingers, with two different pin lengths

- **Pre-charge Vs Power**

  Many connectors have a pre-charge pin on a longer pin than the main power rail. This is used help with inrush current issues.

- **Ground**

  Ground pins are often connected first, to ensure a common system ground as early as possible. This may also help to protect sensitive components if there is a difference in ground levels.

- **Mated/Present**

  Many interfaces have a specific 'present' pin, to indicate that a device is present. If this is placed on a short pin, we can hope that all other signals will be connected by the time the mated signal is seen. Alternatively, the present pin may be designed to mate first, providing a warning to the system that the device is about to attach.

> ***Example: U.2 (SFF-8639)***
>
> This connector has a 3-stage mating process:
>
> 1. GROUND and IF_DETECT (present) mate first, using the long pins
> 2. Pre-charge pins mate next, using the intermediate length pins
> 3. Power, Data and Sideband signals mate last, using short pins

Pin bounce

Electrical pins within a connector do not mate cleanly on contact with one another; typically, a pair of contacts will bounce in and out of connection repeatedly at a microsecond scale before a constant connection is achieved.

This behavior occurs for an insignificant time in terms of human perception, but a digital logic system interfaced to such a signal must deal with many events in a short time.

Pins also bounce out of sync making the connection sequence unique in each case, this is a major reason why 1 in 1000 hot-plugs may fail on a system while the remainder will succeed.

Some pins in an interface will be naturally resilient to pin-bounce (they may be idle until well after the device is detected as present). Others will be much more liable to failure. This might include reset pins, mode select pins and any bus that is used for early communication during a hot-plug.

Pin bounce on pull will likely cause different issues. For instance, it may increase the change of corrupt SM BUS transactions (on PCIe based devices).

Host status

The point in time that a hot-swap occurs can make a major difference. This is especially true for a removal event.

If the system is idle, a removal is likely to be simple. If a large amount of data is moving and a critical transaction is in progress, then there are far more variables.

- The current transaction(s) and errors must be handled
- Pending items in the read/write queue must be dealt with
- Application and/or OS tasks which were accessing the disk may fail
- SM BUS transactions (for PCIe devices, based on I^2C) may be corrupted

Even if a problem is not apparent at the time, a problem hot-swap may have left corrupted memory or invalid settings that will cause failures later.

Host type

Some systems are inherently ‘easier’ to make hot-plug than others. Generally, systems with greater isolation between the host and device are more stable.

On a SAS/SATA system, an unexpected failure during hot-swap will normally end with an error on the SAS controller, which will be noted and handled by the OS. This is due to the controller logic, which isolates the storage device from the host.

With NVMe (over the PCIe bus), there is a much tighter coupling of memory and sideband signals to the host, the storage device can be directly connected with no additional controller logic. This allows for greatly improved performance, but at the cost or more serious failures.

A failed drive can corrupt memory or even cause the host to crash without warning. This is a far more critical failure, as it may result in downtime and loss of critical user data.

Even keeping within the system specification, there are a huge range of potential issues. It is also important to note that you may have to deal with situations that are not in the spec:

> *“Some NVME backplanes do not follow the PCIe spec cold boot sequence to the letter of the law. For example: PERST# is de-asserted too soon with respect to DUT voltage.”*
>
> *(Principal engineer at a major drive vendor)*

While there not be a requirement to work perfectly with an out-of-spec component, it is critical that such situations are detected and handled appropriately.

Testing hot-swap

In order to fully test the hot-swap capability of a system, several things are required:

- Individual, precise and repeatable sequencing control over pin mating times
- The ability to create repeatable pin bounce scenarios
- Driving high/low of critical sideband pins, to simulate different host timings
- Understanding of the interface and most likely points of failure

The human factor

A major reason for inconsistencies between hot-plugs is the human factor, it is practically impossible for a human to insert or remove a device with the same velocity and force in a repeatable manner. Differences in the time between pin's mating has a huge effect on the power up profile of a device, making it impossible to manually test a hot-pluggable device over all its likely operating conditions.

Questions over why two apparently identical system fail at different rates may come down purely to the way the different operators insert drives.

This means than any hot-swap testing involving human intervention will always be limited to basic functionality testing.

Robots are little better

In some cases, robotic systems are used to hot swap devices as a time saver. These systems are still unlikely to produce deterministic results. Mechanical factors, friction and pin-bounce will still make each insertion unique.

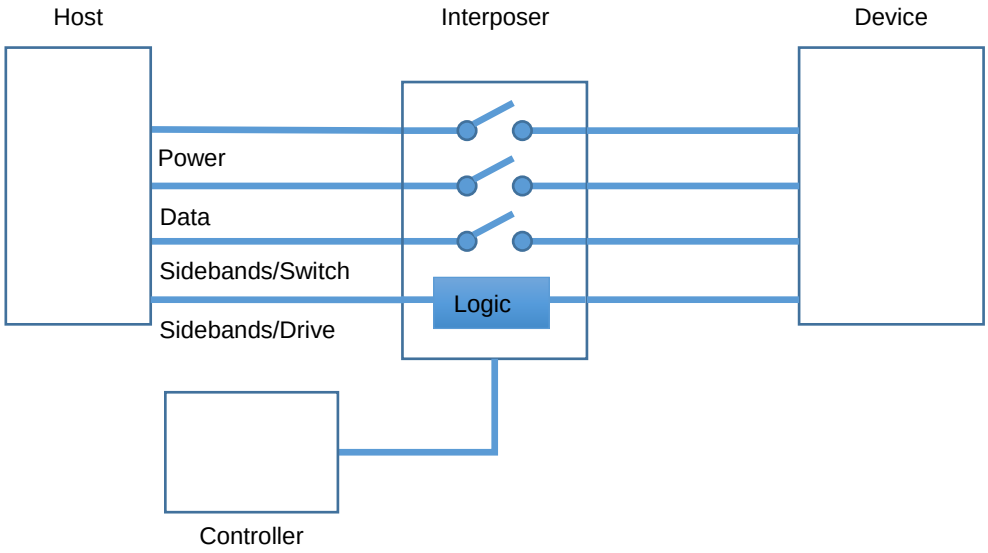
In addition, each system and connector will be subtly different. Connector and manufacturing tolerance will cause minor, but significant changes to the way they mate.

Robotic systems help with automation and will reduce variability in some cases, but still fail to meet the main requirements.

Testing in full

In order to meet all the requirements, we need an automated test which will give us full control over the hot-swap sequence, but not affect the communication between the drive and host.

To do this, we need an interposer in the link, but one which is passive and will not retime or otherwise alter the data.



Interposer layout

In this design, we can see that individual switches are used for all power, data and sideband pins in the connector.

Some sideband signals may need additional driving as well as simple isolation. This will be required in order to simulate different backplane assertion timings and similar.

Interface research

The first step is to confirm the approximate range of hot-swap speeds and confirm if pin bounce does exist in these cases.

An oscilloscope was used to monitor three different pin lengths on a 2.5” SFF drive sled. The pre-charge and power pin were chosen, along with a ground pin which gives access to all 3 lengths of pin.

The system under test was a bare backplane and drive connector without rails or latching mechanism, therefore the following results can be assumed to be ‘worst case’ for a fast plug or pull scenario as the additional insertion force created by drive rails and latching mechanism would make it unlikely to achieve a faster speed than this.

Measured limits of hot-swap speed

Test Number	Delay GND to Charge	Delay Gnd to Power
1	4.44 mS	6.72 mS
2	4.70 mS	6.25 mS
3	2.88 mS	6.30 mS
4	2.67 mS	5.12 mS

Hot-plug events, fastest possible insertion

Test Number	Delay Power to Charge	Delay Power to GND
1	3.28 mS	3.78 mS
2	4.46 mS	3.59 mS
3	3.67 mS	6.87 mS
4	2.00 mS	3.50 mS

Hot-Pull events, fastest possible removal

Note that there is both variation in the timing, and a trend to get faster, as the operator became more used to the action.

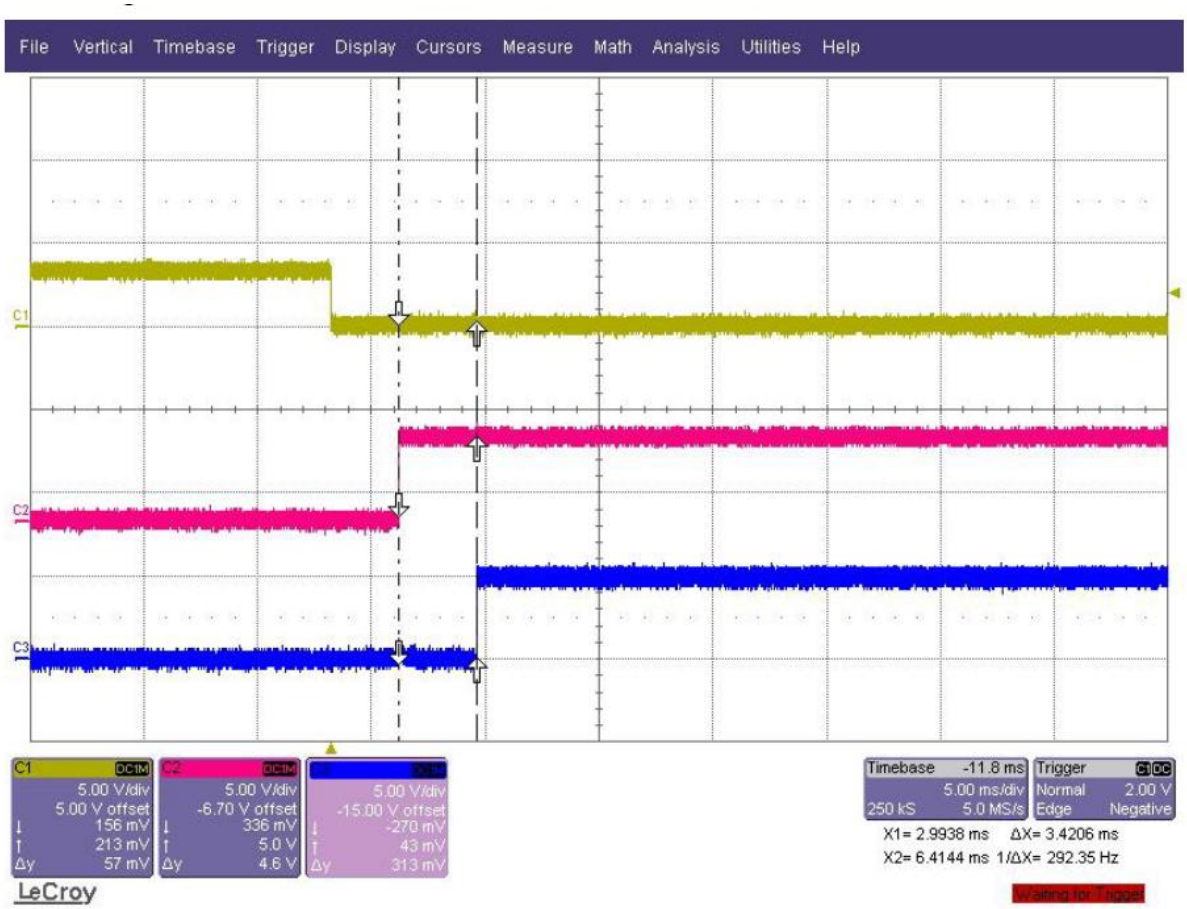
This confirms the unique nature of each event, and also demonstrates how the operators experience and habits can affect the results.

We also see that a pull event tends to be slightly faster than a plug, probably due to the additional alignment of the connector, receptacle and latch that occurs during an insertion.

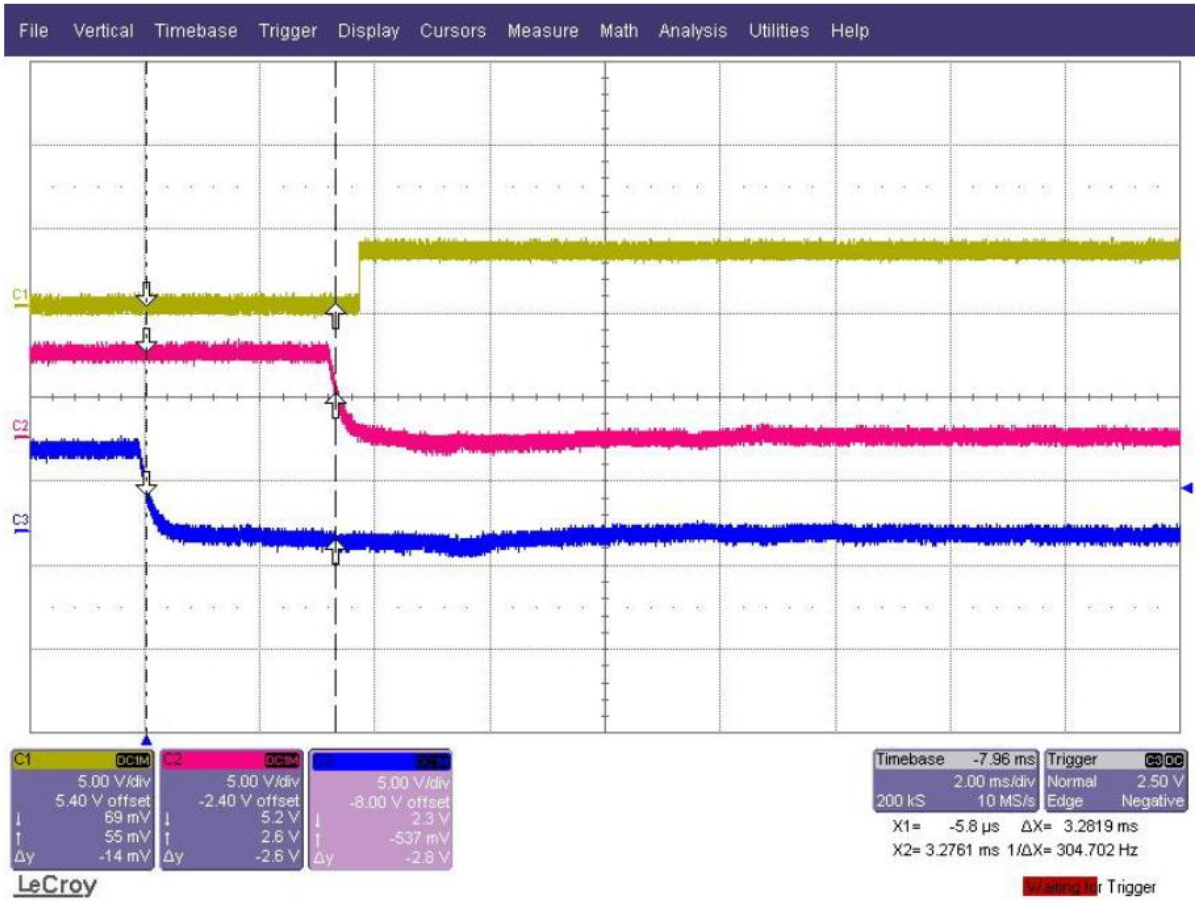
Scope examples

Here we can see a plug and pull event, as captured on the scope.

Gold is Ground
Red is Pre-charge
Blue is Power



Example plug/insertion event



Example pull/removal event

Note that:

- GND pin is pulled down to ground on plug
- Pre-charge and Power pins are pulled up to their rail on plug
- The voltage rails are held up by drive capacitance and fall more slowly on during the drive pull

Slowest possible hot-swap

There is no practical limit on the slowest end of the scale. In the worst case, a drive can be partially inserted and left in that state for an unlimited period before finally being latched into place.

To ensure full testing of all possible scenarios, the slowest plug/pull time should be longer than the time for the device to power up, mount and be accessed by the host. This is likely to be in the order of 5 to 10 seconds.

Standard/default hot-swap speed

As well as defining the corner cases, it is sensible to define a default ‘safe’ hot-swap speed, which is chosen to be both realistic and very unlikely to fail. This gives us a starting point to test outward from.

The average fastest plug pull time (from above) is around ~5 mS. This will be a high-stress scenario, so we must be much slower than this.

Further drive sled testing confirmed that (though with very wide variation) a time of 50 mS for the hot-plug appears to correspond well with a smooth, average hot-plug event.

A 50 mS hot-swap time will give an order of magnitude more time for Pre-charge circuits to work but will not risk appearing ‘slow’ to the devices under test.

This means a 25 mS time between the mating of each length of pin.

Pin bounce measurement

Not all hot-swap events produce pin bounce, repeated plugs and pulls were performed in order to capture realistic pin bounce examples and the results are shown below. The traces show vast variability over the length and severity of the pin bounce.

In practice pin bounce can occur at any time the connectors are moving across each other. It can also stop for a period then start again further through the plug/pull event.

We also need to be aware that the different lengths of pin meet at different times and so will be likely to bounce at different times.

Example of measured pin bounce on drive pull

Example of measured pin bounce on hot-plug

Test plan outline

The goal is to describe an industry standard test plan for all hot-swap capable systems. This can be modified for different interfaces and requirements but will outline the main tests required and the critical points than need covered.

The plan that starts with the more basic tests first, even if it results in more test points in total. This will make it easier to identify the reason for the failure.

It will often be possible to force a failure by creating an extreme scenario. This still provides valuable information, as it allows us to see where the system does breakdown, then evaluate if this is acceptable (and if the system responds to the errors correctly).

What we're looking for

- Consistent, reliable behavior
- Pre-charge/power up system work reliably
- Correct enumeration and operation after each cycle
- If fails occur, that the host/device error in a sensible way and recover after another cycle
- Points in the interface specification where problems are more likely to occur

To do this, we will need many hot-swap cycles across a range of scenarios

1. At a standard speed

   This is a simple first test, ensuring the basic hot-swap function works

2. Across a range of speeds from very fast to very slow

   This checks all speeds of hot-swap work, including corner cases

3. Where each pin/group in turn are given an low level of pin bounce

   This is a simple test of pin bounce resiliency across the interface

4. Where a more significant length/severity of pin bounce at the limit of what is possible

   Now we do corner-case testing of pin bounce on each pin

5. Worst-case pin bounce, where every pin bounces for the maximum extent of the plug time

   This is an extreme test where pin bounce is well beyond its theoretical maximum

6. Sweep the mating time of a pin/group across the remaining pins

   Here we look for specific timing issues, by changing the mating time of a single part of the interface in comparison to the rest.

7. Interface specific tests

   Knowing the electrical/protocol design of the interface, run test cases to exercise the critical areas. This will include specification mandated timing limits.

In cases such as pin-bounce where there are a very large number of tests possible, we may chose to test a limited number of points, representing the just ‘likely’ and ‘extreme’ cases. If these tests all pass, the intermediate ones are very likely to as well.

PCIe/NVMe Specific

Spec Point 2.2.1 – Initial Power-Up (G3 to S0)

"As long as PERST# is active, all PCI Express functions are held in reset"

This spec point indicates that PERST should only de-assert after the host power rails and REFCLK are stable.

Power must be stable for at least 100mS before PERST may de-assert

REFCLK must be stable for at least 100US before PERST may de-assert

TEST: Use the drive feature to de-assert PERST at the minimum time after the interface is stable. The hot-plug should still succeed

TEST: Use the bounce feature to corrupt/break the interface prior to plug. Stop the corruption and then de-assert PERST after the minimum time. This should be done for individual critical signals (Power, REFCLK, DUALPORT_EN and SM_BUS are likely examples). Testing each individual signal then testing all combined at the same time is recommended to cover all likely failures

"Enterprise PCIe SSD may train as a single x4, x2 or x1 link. Support for x2 is optional for both the system and the PCIe SSD"

TEST: Plug the module with lane width limited to x2 and x1 link width (not mandatory to work, but should be tested if supported)

"[In dualport mode] Either port may train only as x2 or x1"

TEST: For dualport systems, plug the module with lane width on each port restricted to x1 in turn.

“When an Enterprise PCIe SSD is not capable of dual port operation… Enterprise PCIe SSD should train on either port A or Port B in x2 or x1”

TEST: For dualport host with single port drive, plug the module with DUALPORT_EN asserted and verify the drive enumerates and is accessible.