Status: PR #277 needs some minor merge conflicts that need updating, but should be otherwise ready to go.

This PR is a major update to the trap handler and standard macro header file. It:

* splits out test specific macros (e.g. TEST\_FPRR\_OP) into a separate file to reduce merge conflicts (such as the ones preventing it to be merged now…), and
* adds support for traps into different privileged modes and CLIC requirements.
* fixes (too) numerous (to mention) bugs
* harden code for error conditions (e.g. trap signature overflow, unconfigurable xTVEC))
* many small code optimizations, including reduced reliance on the LA() macro

**Discussion: Misaligned store handling.**

Misaligned stores 4 different legal behaviors: trap before writing anything, trapper after writing higher address, trap after writing lower address , or storing both halves without trapping. There may or may not be deterministic rules that can be followed to select which of these behaviors are implemented, but that is not required. It is unrealistic to define all possible cases for Sail to emulate.

The proposal is a “partially” self checking test which will effectively force the DUT result to look like the SAIL result if the DUT result is the expected value.

**PMA restrictions**

PMAs We need to put In restrictions on what PMAs on the regions that test execute in can restrict.

e.g. writes to code regions must be allowed, otherwise we can’t even load the tests.

So, a requirement for a DUT is that that test memory must have RWX PMA permissions.

PMPs or PTEs can further restrict these, of course

This may require that secure implementations have boot code that does not restrict permissions in either PMAs or PMPs (e.g. don’t allow entries to be locked before handing over control to the tests – which is another battle for custom core-only test benches when an SOC might have on-chip bootcode that does this).

Another battle is whether the PMA is considered part of the core, or part of the SOC.

In general, though, implementations have model-specific linker scripts that can put test code into regions that have those permissions – though this make come with size restrictions.

**PMP Test setup**

!0xe presented a proposal for how PMP tests will be structured.

It requires 4 PMP regions with 6 PM entries to cover all cases.

(see slides)

Issues: how do we deal with PMP WARL configurations?

Granularity can probably be dealt with, but other WARL configurations

(e.g. restrictions on address ranges that can be written, Read-only bits, values of mode field, etc)

One view:

* we will only properly test PMP whose entries that are contiguously number starting at zero
* we only properly test PMP entries where all bits that can be written must be RW

(e.g. PMPADDRx is RW from bits PA\_SIZE-1..G (where G is log2(granularity)-2. )

Zfinx: sign extension not working in ctg

(or perhaps, CTG is setup to look for NaN boxing, but not sign extension)

- this is being taken offline for discussion.

Zfinx will be stalled for 4 weeks for travel.