Section 1

Microarchitecture Side-Channel Resistant Instructions Spans (scrispans)

Examples semantics

- `spansec.create [policies]`: create a new scrispan with given optional set of security policies (flags).
- `spansec.save rd`: save the current scrispan configuration (ID + security policy) in a register.
- `spansec.restore rs1`: restore a scrispan configuration from a register holding ID and security policy.

We assume that a change of scrispan ID implies microarchitectural isolation.

- `spansec.alter [policies]`: in addition we could add an instruction that keeps the ID unchanged but modify the security policies.

Proposed encoding 1

Using a custom opcode for now. The scrispan state is split between an ID (e.g. 24 bits) and policy flags (e.g. 8 bits).

- `spansec.restore rs1, imm`: load ID and policy flags from `rs1`, then update policies from `imm`.
- `spansec.save rd`: store ID and policies in `rd`.

Pseudo instructions:

- `spansec.create imm` = `spansec.restore x0, imm`.
- `spansec.alter imm` = `spansec.save xX`; `spansec.load xX, imm`.

Proposed encoding 2

Using CSRs to store ID and policy flags.

- scrispan ID CSR = 0xXXX
- scrispan policies CSR = 0xYYY

Pseudo instructions:

- `spansec.create imm` = `CSRRW x0, x0, 0xXXX`; `CSRRI x0, imm, 0xYYY`
- `spansec.save rd` = `CSRRW rd, ?, 0xXXX`; + same for flags

Allow altering policies with `CSRRS` and `CSRRC`. Atomic behaviour of `CSRRW` could be a good thing ?