Virtual Tapeout Concept

- Real tapeout is not very agile
  - Requires multiple iterations, usually at the end when design is finished
  - First iteration is most painful, successful iterations less so
  - Tapeout usually only done ONCE per design cycle!?

- Virtual tapeout - still very real!
  - But does not (always) go to foundry
  - Tapeout early, tapeout often
  - Start with simple base chip, iterate as design progresses
  - Tapeout / mini-tapeout on every check-in (a la Travis) !!?

Virtual Tapeout (Tape-In)

- Start small: need base flow and a base chip to work from.
    - Pad frame only, then pad frame plus one tile, then...

- Small/fast per-tool test cases for the flow?
    - One for each problem found in the main flow for fast debug
    - Each new tool, technology, uses these tests to boot up rapidly

- Flow will need info from design to create chip
    - Info might include floor planning, power domains, technology details…
    - All this extra info needs to be created by the generator

- Real chip continually created every (week? commit?)
    - To ensure it doesn’t get stale
    - Prev-week design is always available when/if crunch time comes

Proprietary or Open-Source?

- Tapeout must run on machines with proprietary library info
  - Flow has to seamlessly handle proprietary IP
  - How do we do that?

- BUT can also use these flows to create a design in an open pdk

- This gives a path for the larger design community to help
  - Tune the design flow
  - Debug the flows to make then technology agnostic