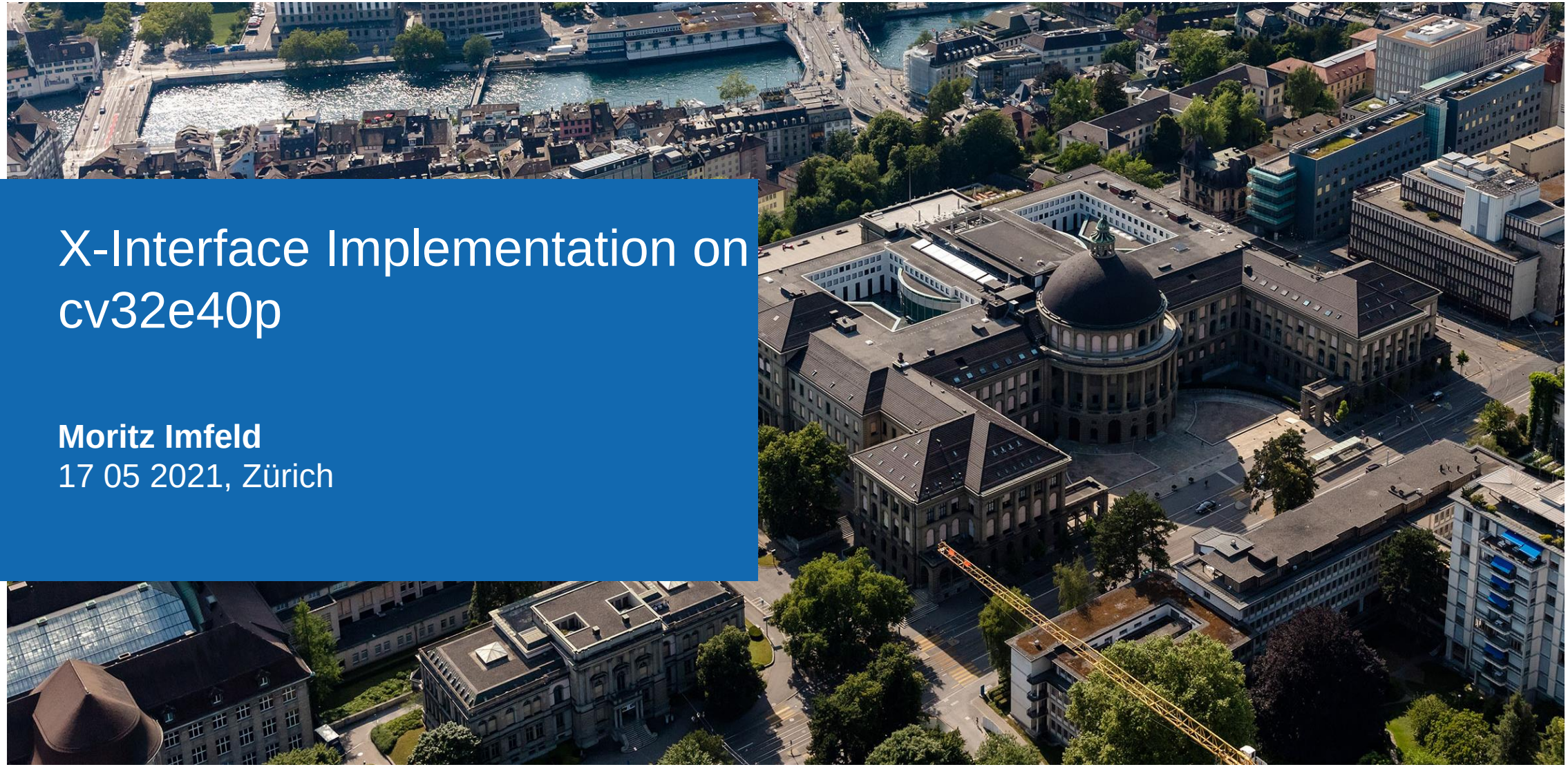
ETH zürich



X-Interface Implementation on cv32e40p

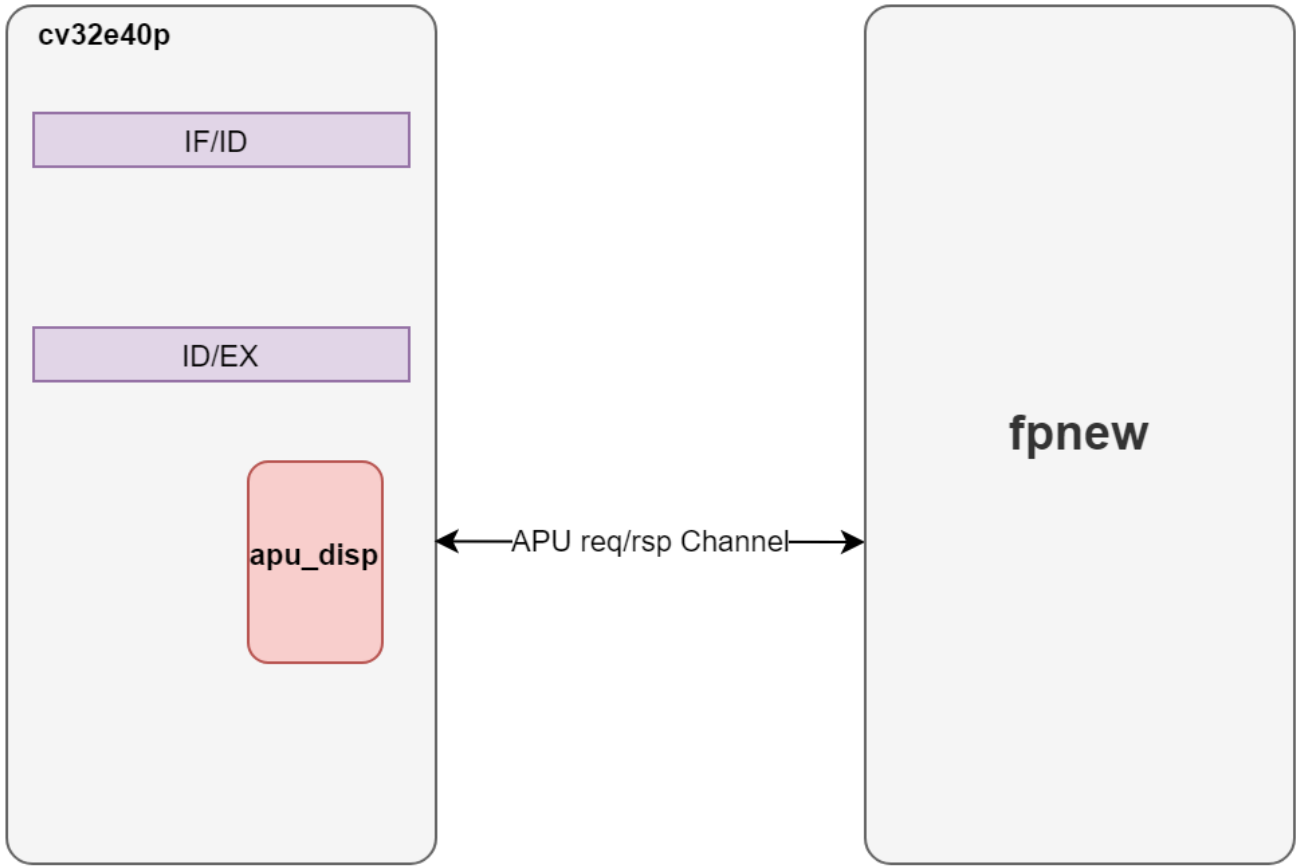
Moritz Imfeld

17 05 2021, Zürich

Table of contents

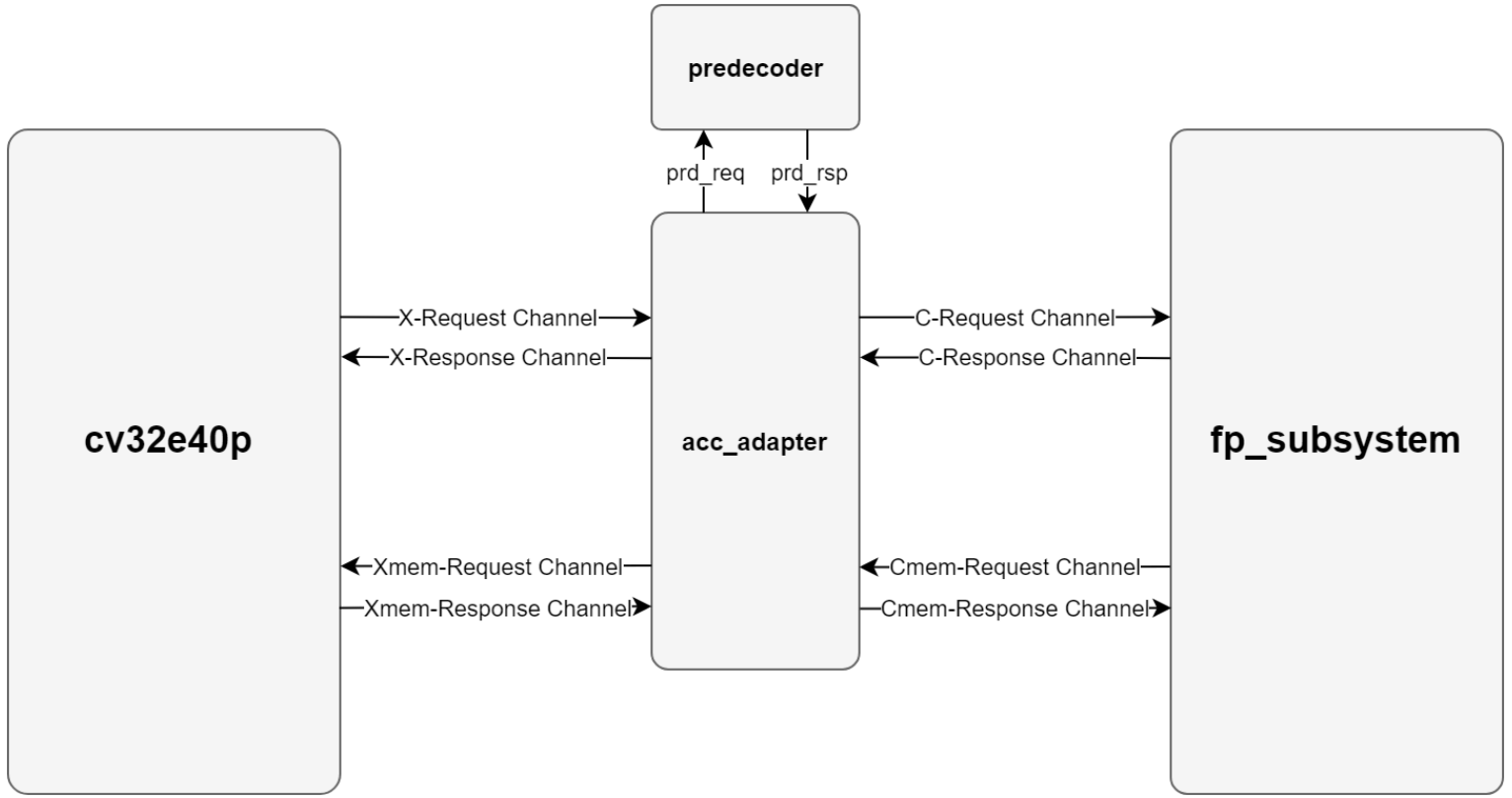
Before X-Interface

- Offloading in EX stage
- Core can continue after successful offload (e.g. instruction is in fpnew or in a buffer)
- fp register file in core
- Memory instructions stay in core
- fp CSR in core



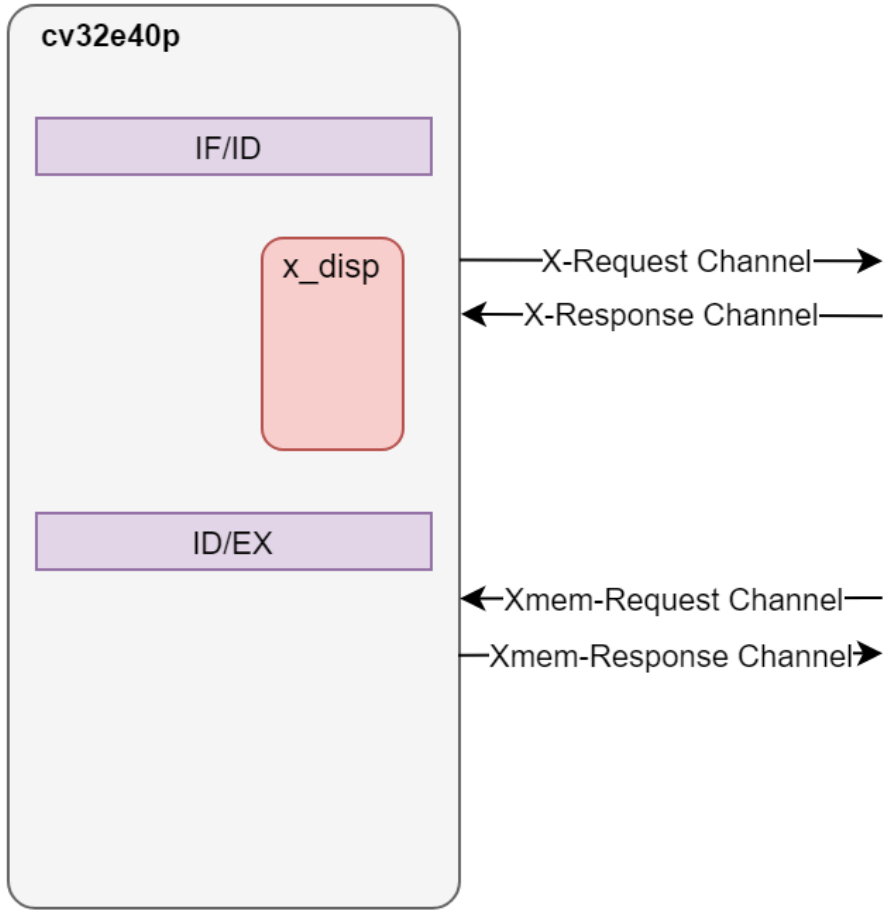
High Level Overview of X-Interface with FPU

- REIv0.1_32TM
- Extensions
  - “F”
  - (“xfvec”)



Core Side Implementation

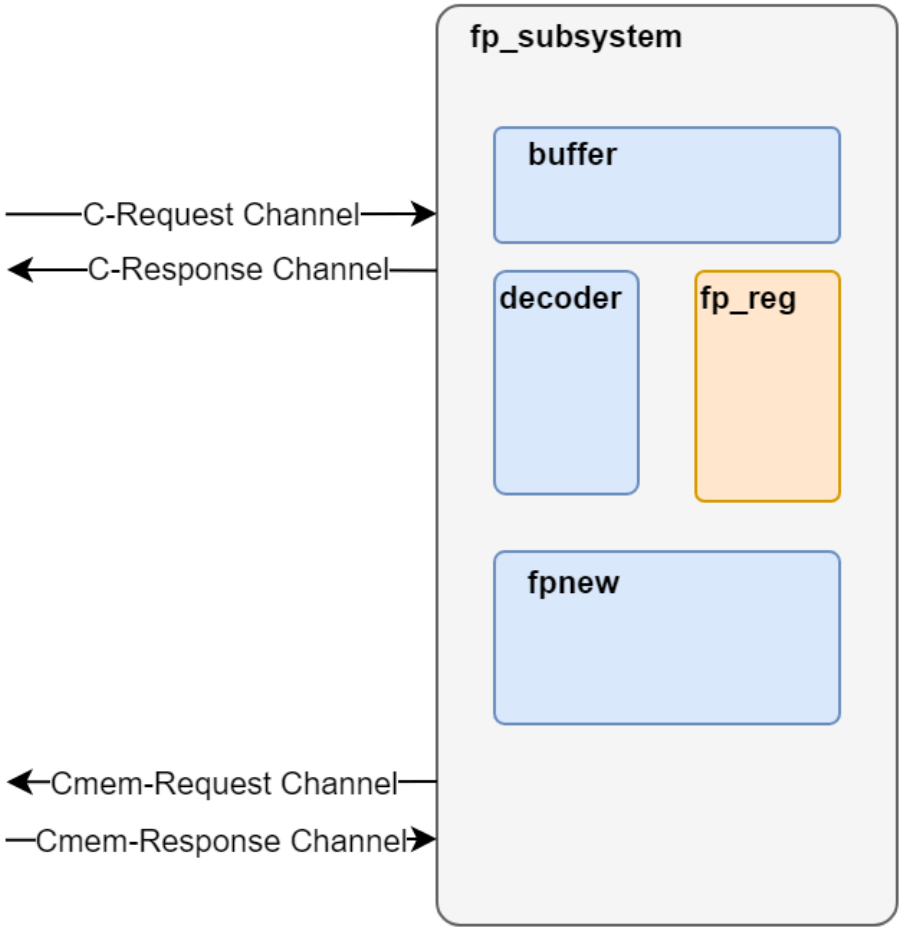
- Offloading in ID stage
- x_disp:
  - Scoreboard, to track outstanding writebacks
  - Dependency check for known and unknown instructions
  - Exception check before offload
- Memory instruction execution in EX Stage (not implemented yet)
- Core can continue after a successful offload
  - Except for memory instructions
- Keep logical equivalence w.r.t. cv32e40p_v1.0.0 when parameter FPU = 0
  - LEC with Cadence Conformal



fp_subsystem Implementation

- Modules
  - Buffer (with parametrizable depth)
  - Decoder
  - FP Register File
  - CSR Register (not implemented yet)
  - Controller for handshakes and Memory instructions
  - fpnew
- In order dispatch, in order execution (at the moment)

- Further development
  - In order dispatch, out of order execution(possibly during my thesis)
  - Out of order dispatch, out of order execution (not during my thesis)



Findings/Questions/Comments

- “C”-Extension not yet supported by cv-x-if
  - Possible solution (that I might implement):
    - Recognize illegal “c”-instruction
    - Expose registers w.r.t. “c” encoding