Update on Ara

26/05/2021

Matteo Perotti
Matheus Cavalcante
Nils Wistoff

Professor Luca Benini
Integrated Systems Laboratory
ETH Zürich

Ongoing work

- **Benchmarks + Verification**
  - Jacobi2d now works
  - Fixed:
    - Misaligned stores
    - Misc bugs
  - Added:
    - Strided memory operations
  - New riscv-tests
    - Misaligned mem ops
    - Strided mem ops
    - vsetivli (config)

- **TO**

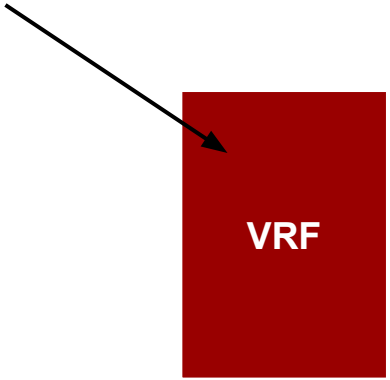
Aligned stores

Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

Support for aligned stores

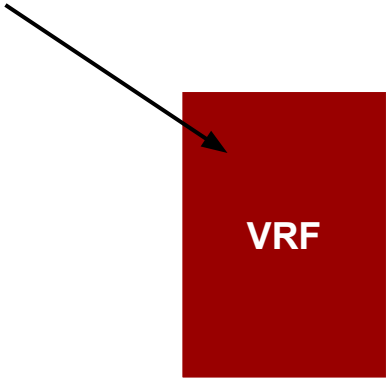
- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

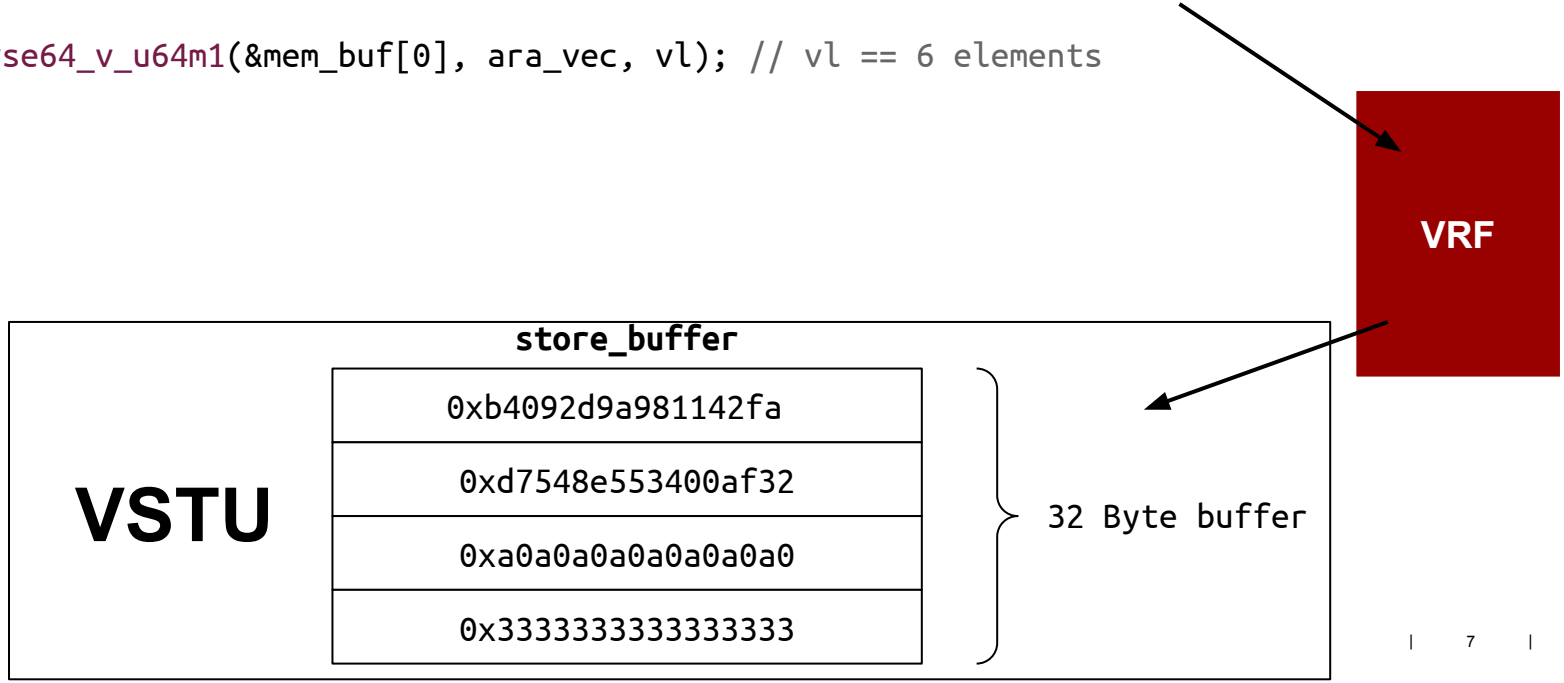
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

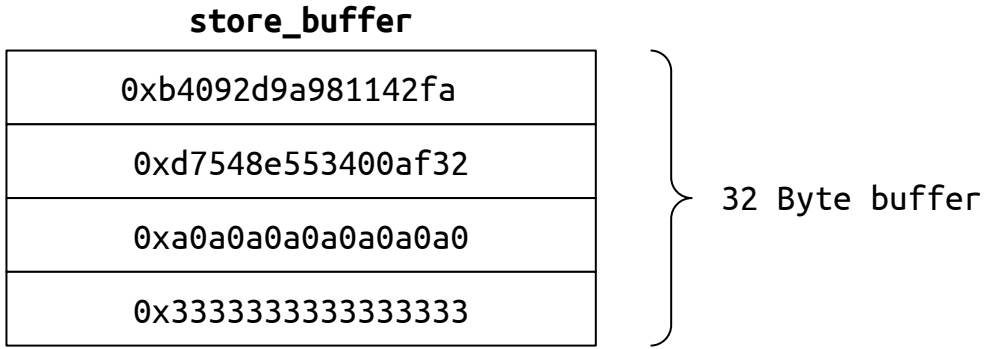
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

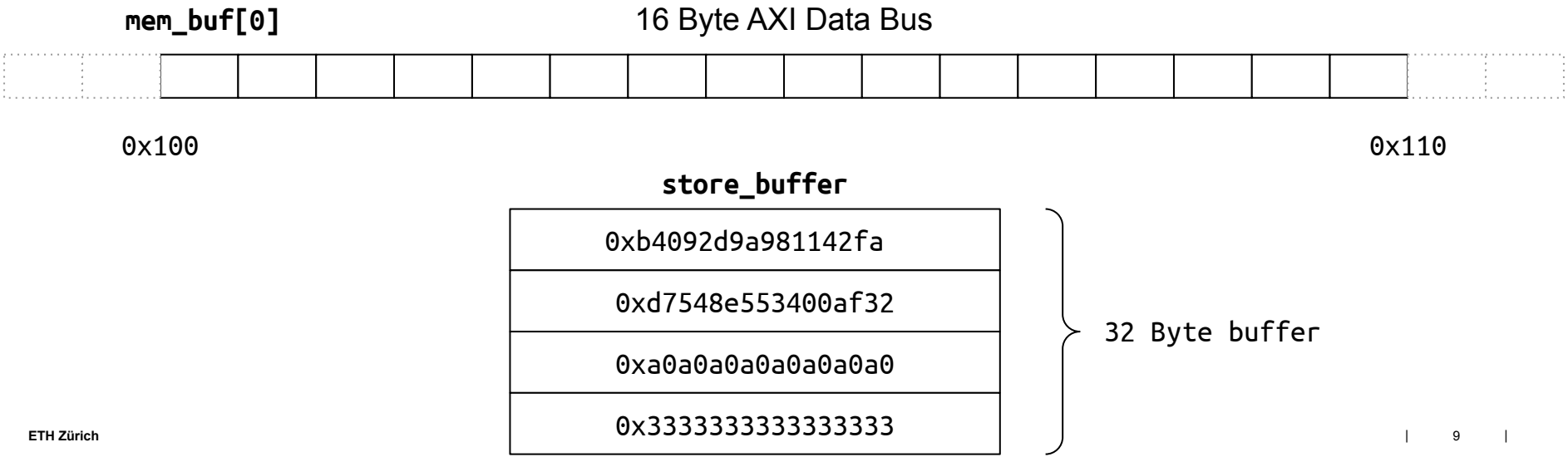
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

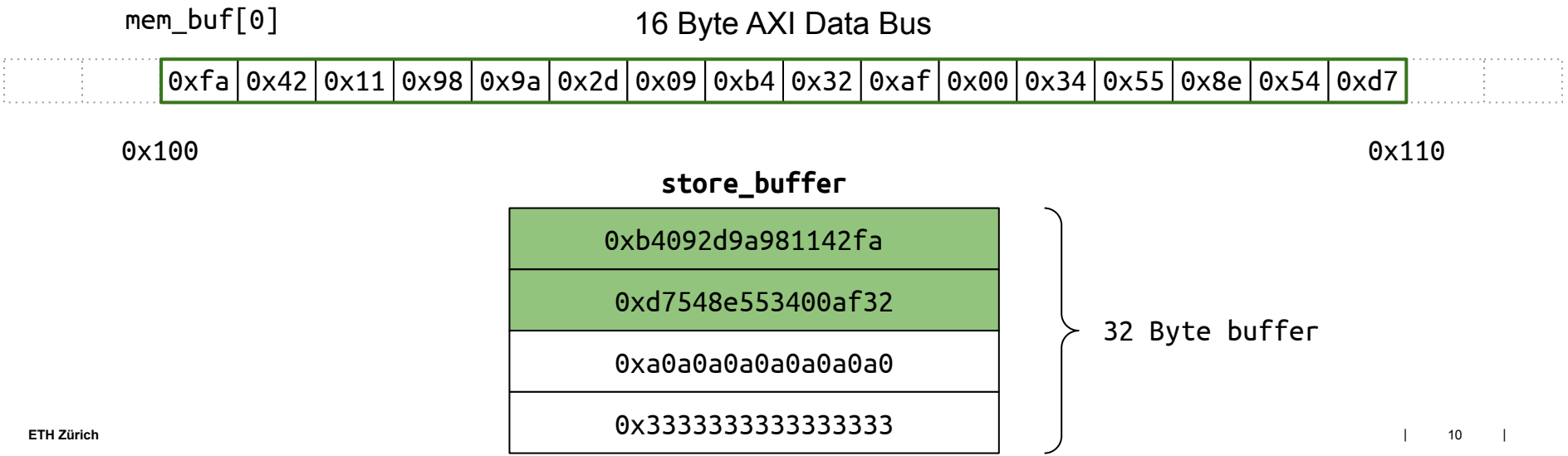
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

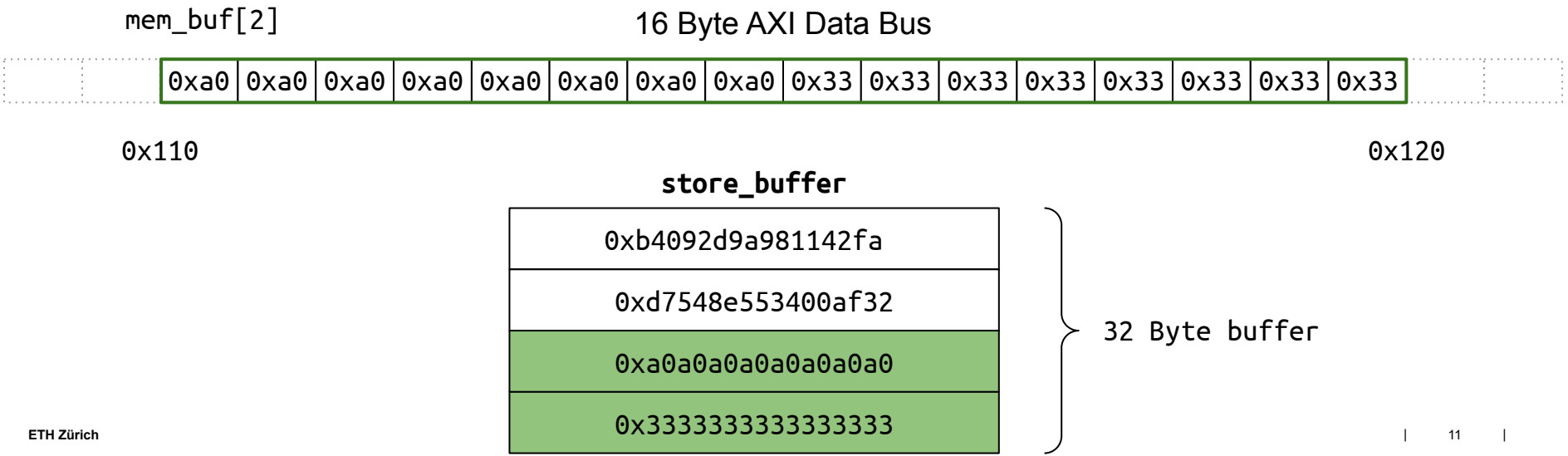
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

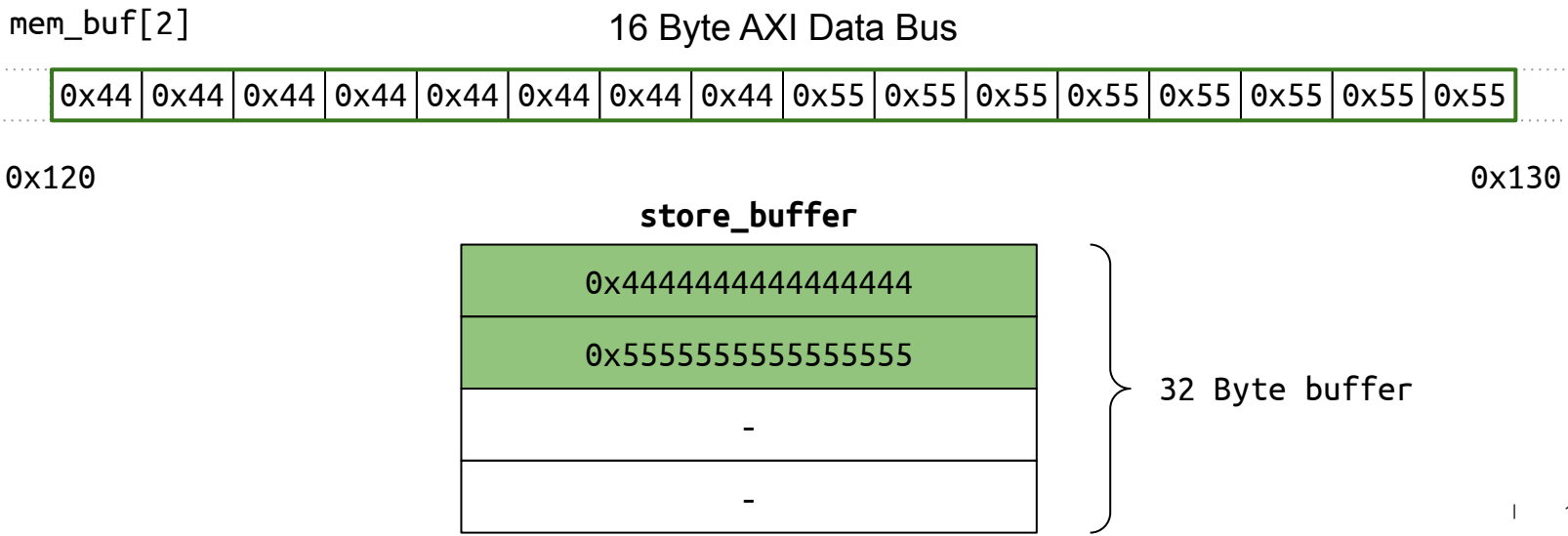
- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 6 elements`



Support for aligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

- `vse64_v_u64m1(&mem_buf[0], ara_vec, vl); // vl == 4 elements`



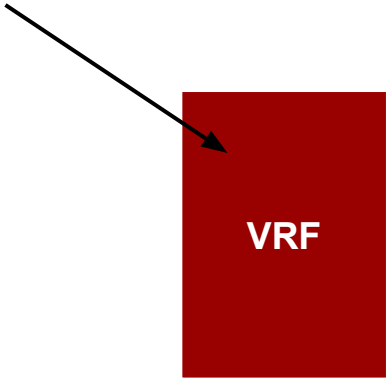
Misaligned stores

Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

Support for misaligned stores

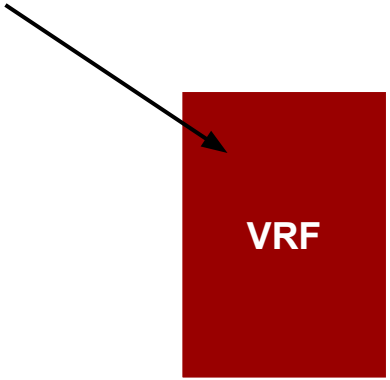
- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`



Support for misaligned stores

- ```
  vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,
                         0xa0a0a0a0a0a0a0a0, 0x3333333333333333,
                         0x4444444444444444, 0x5555555555555555};
  ```

- ```
  vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements
  ```



Support for misaligned stores

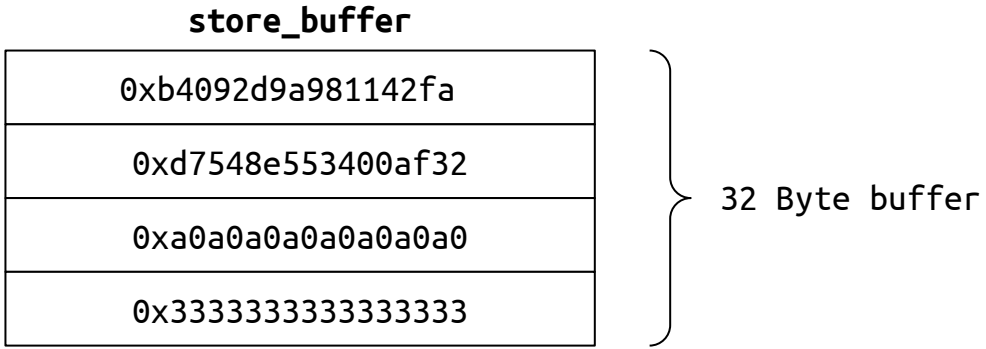
- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`

Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

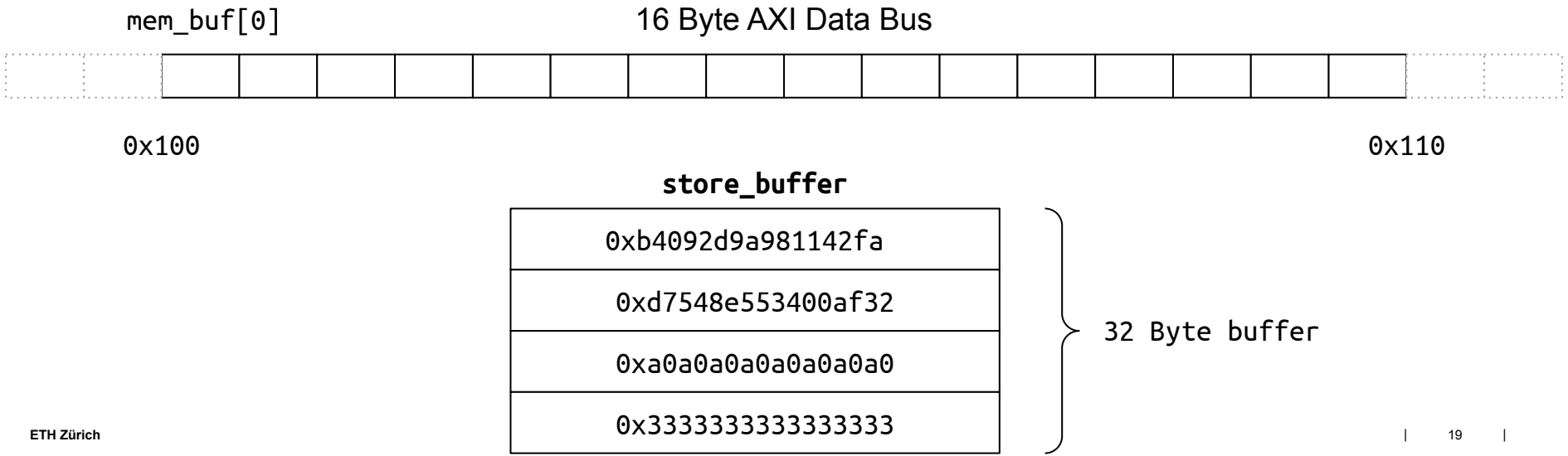
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

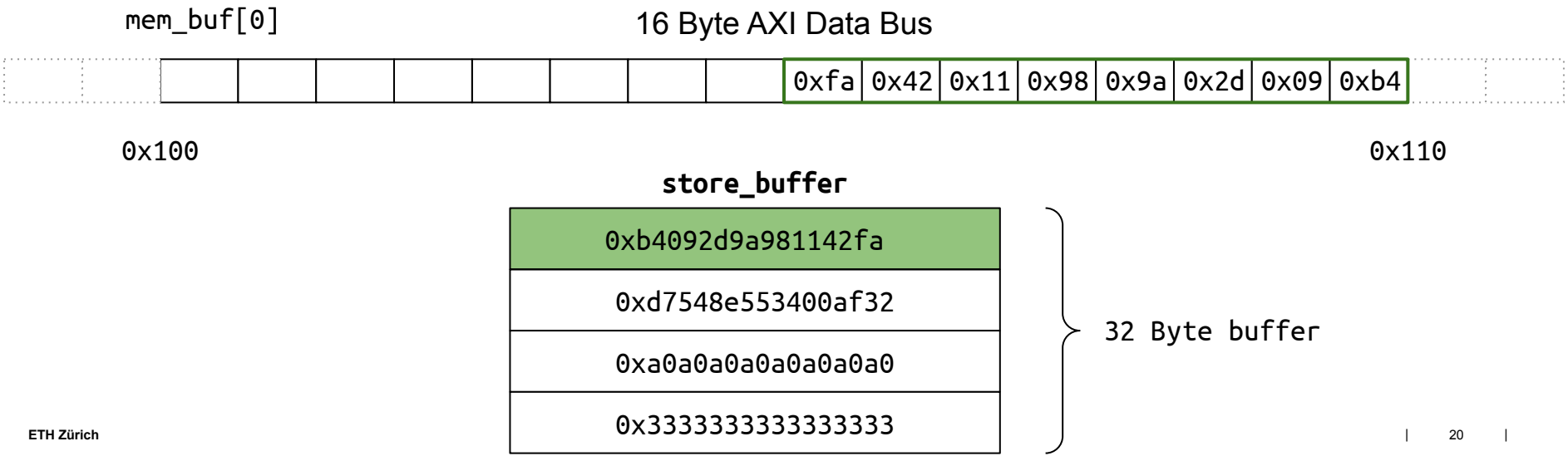
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

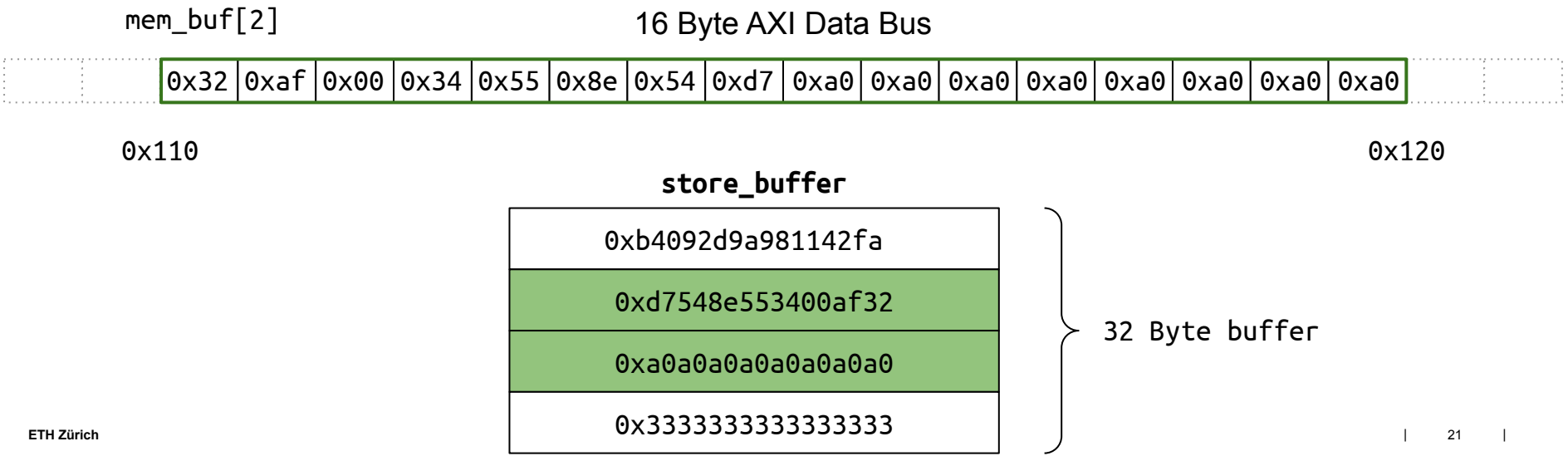
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

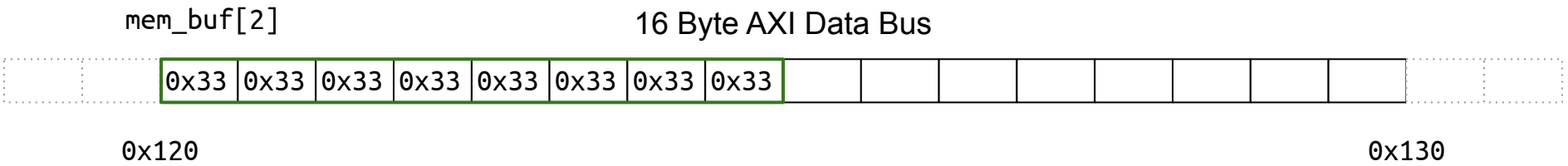
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



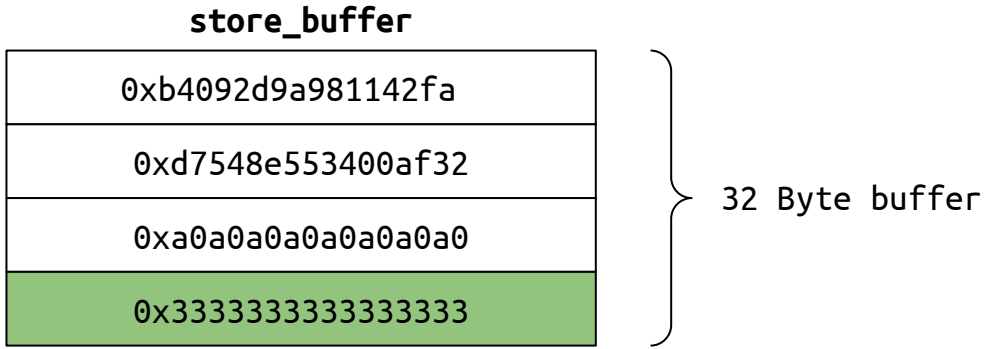
Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



We need the next element to complete the AXI beat. But if we get the next element, we lose the previous one.



Support for misaligned stores

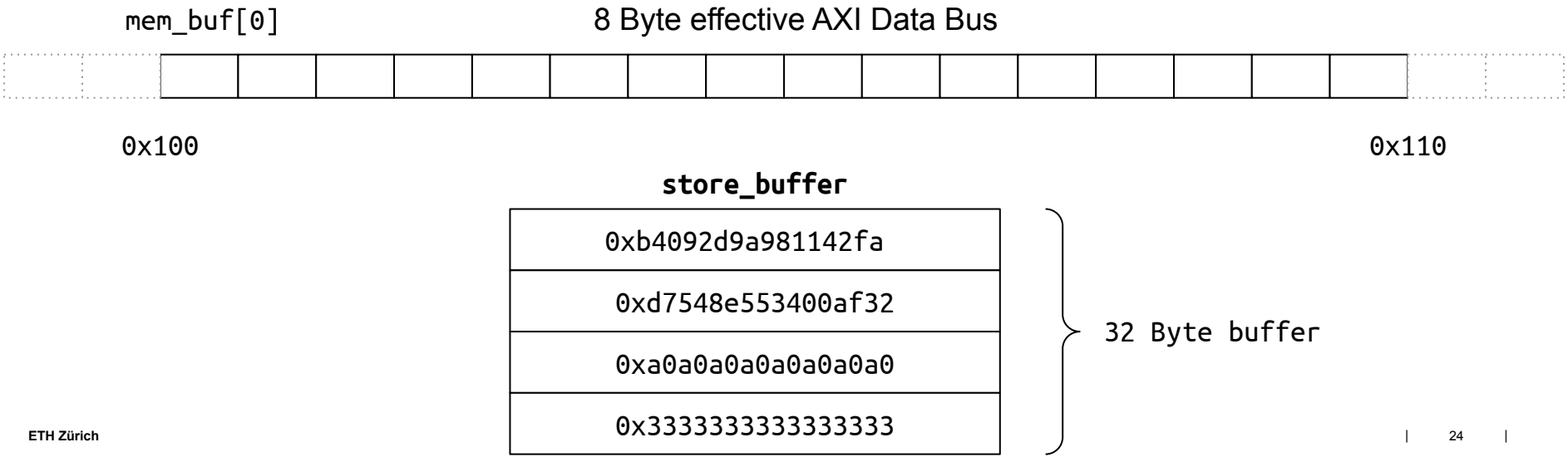
Multiple solutions:

- **Double buffer**
  - We always keep also the next elements
- **Independent ready-signals to the lanes**
  - Each buffer entry can be updated independently
- **Reduce the effective AXI width in case of misaligned store**
  - Less intrusive solution
  - Faster to implement
  - Misaligned stores are not common

Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

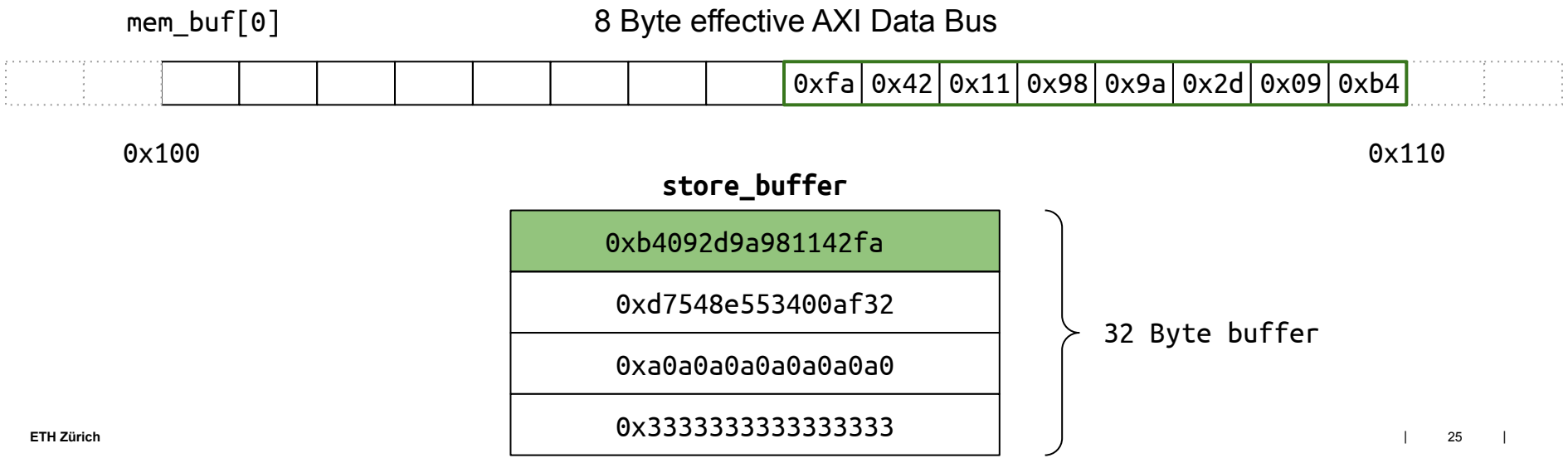
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

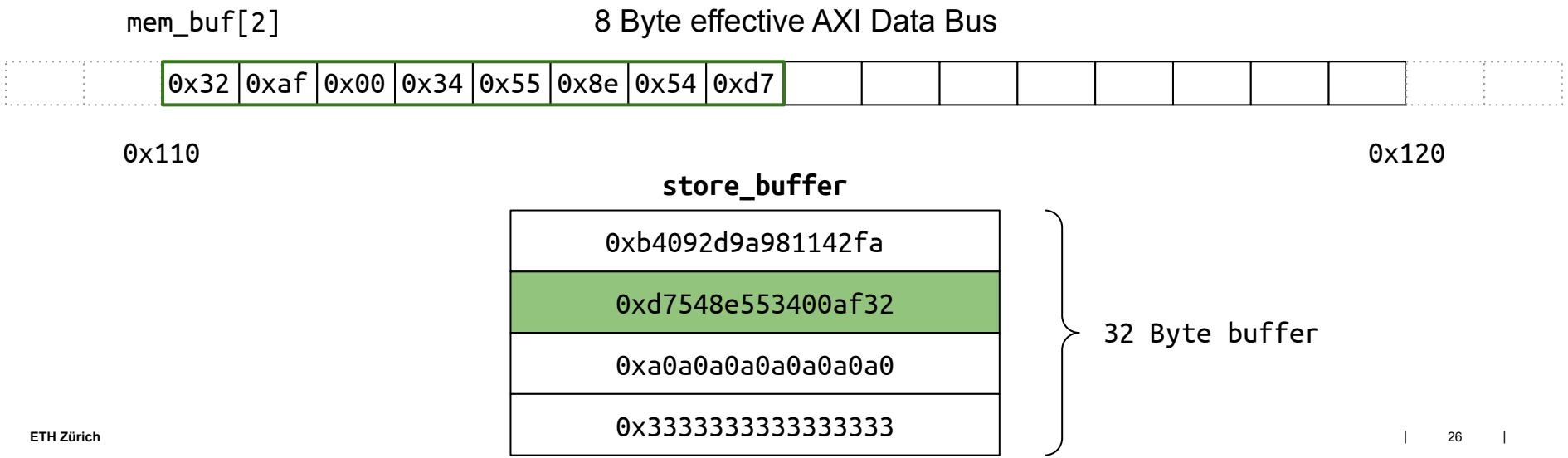
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

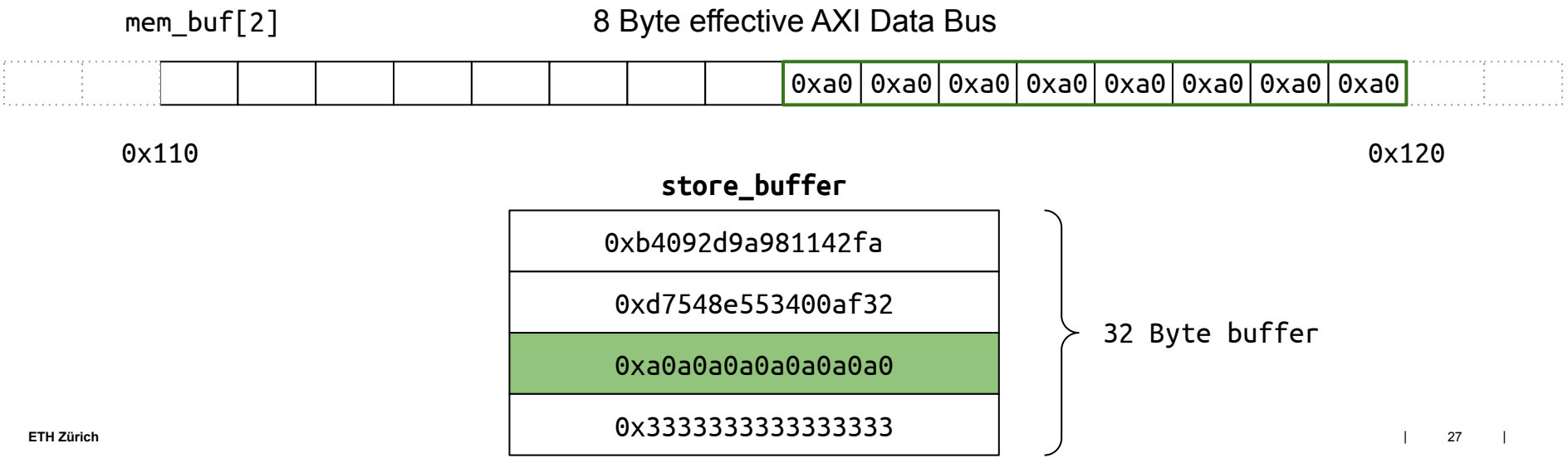
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

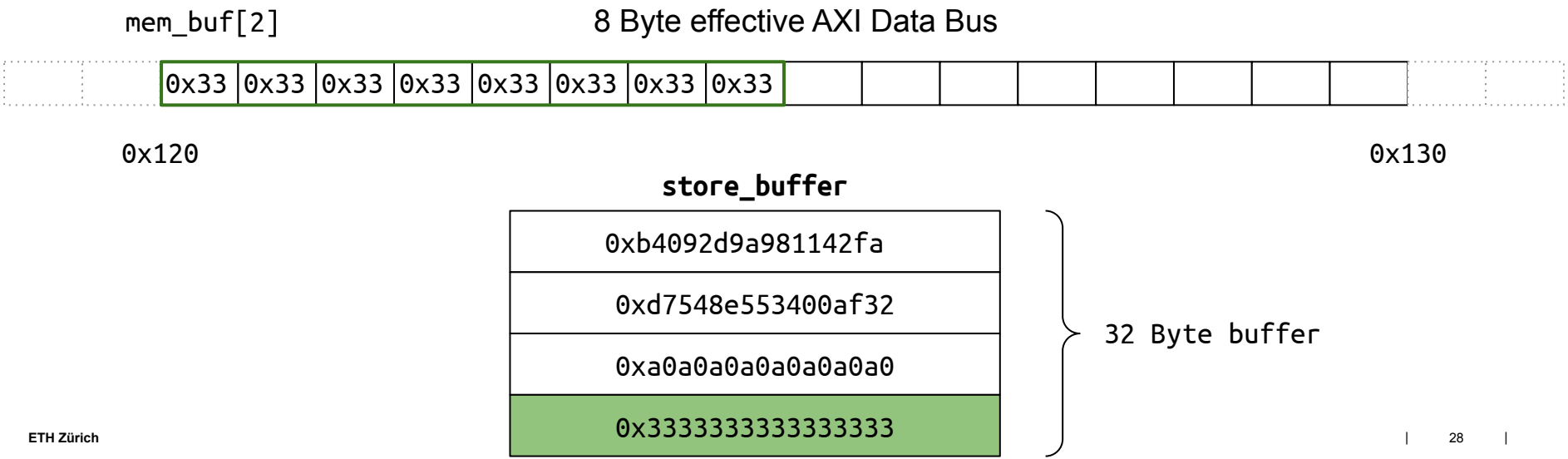
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

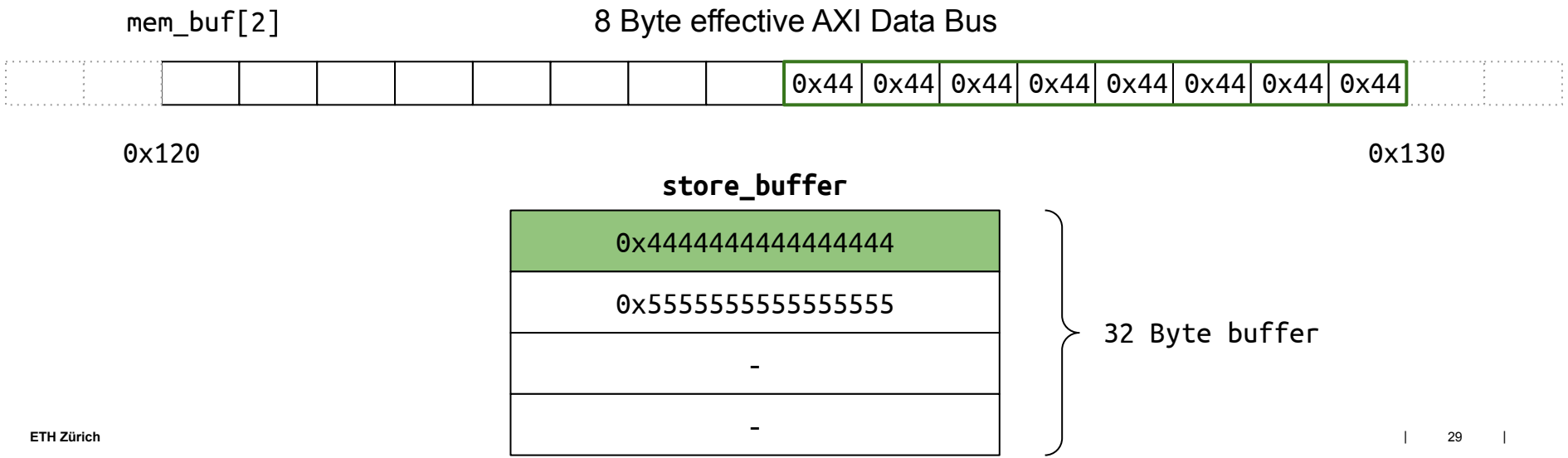
- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`



Support for misaligned stores

- `vuint64m1_t ara_vec = {0xb4092d9a981142fa, 0xd7548e553400af32,`
  `0xa0a0a0a0a0a0a0a0, 0x3333333333333333,`
  `0x4444444444444444, 0x5555555555555555};`

- `vse64_v_u64m1(&mem_buf[1], ara_vec, vl); // vl == 6 elements`