- **Background: readout system in proton collider**
  - There are multiple "events"; each "event" has multiple "particle tracks"
  - Each "track" belongs to a "cluster"
  - Each "cluster" is a collection of fired "pixels"



On the left: Photo of an ALPIDE chip. On the right: General architecture of the ALPIDE chip
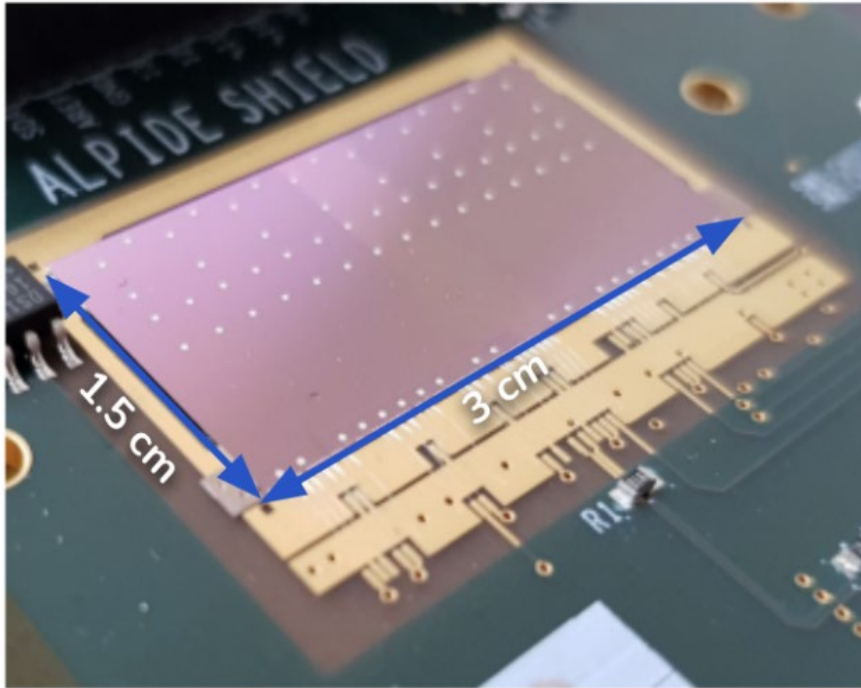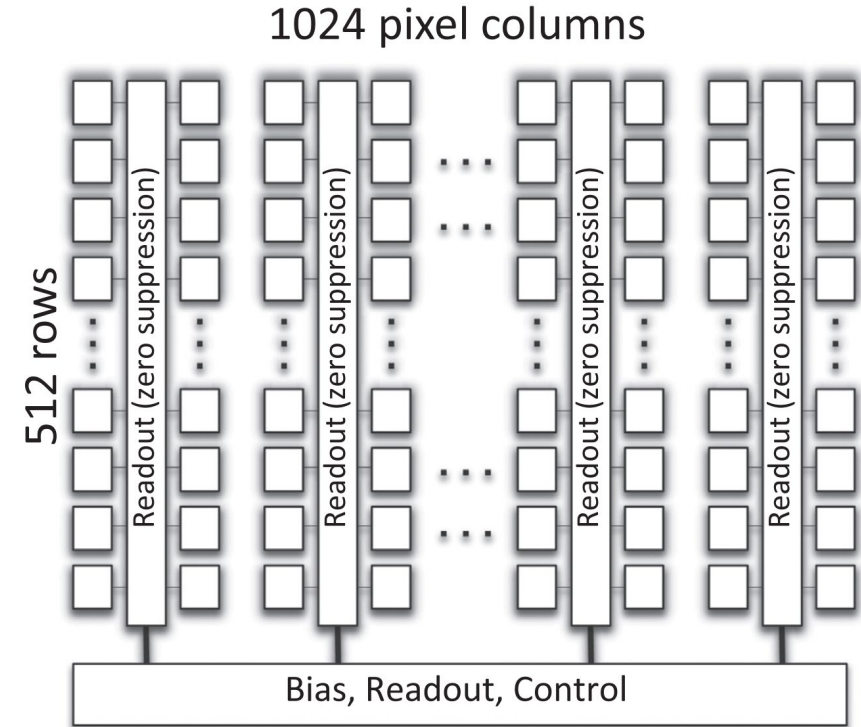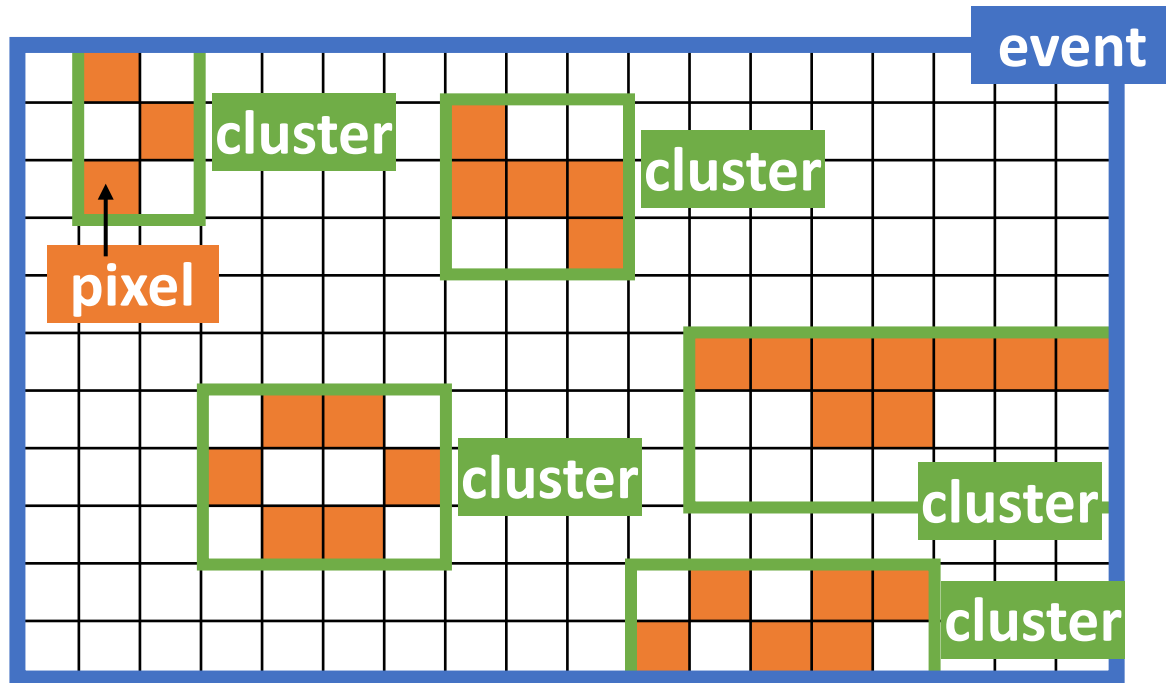
- **Background: readout system in proton collider**
  - There are multiple "**events**"; each "**event**" has multiple "**particle tracks**"
  - Each "**track**" belongs to a "**cluster**"
  - Each "**cluster**" is a collection of fired "**pixels**"

- **How Readout is done:**
  - Readout is zero suppressed, so only fired pixels will be transmitted, no zeros
  - In a zig-zag order: pixels are fed into FPGA one by one, each pixel one clock cycle
  - As long as two pixels are adjacent (horizontal, vertically, diagonal), they belong to the same hit

**Challenge: one cluster can be cut into multiple readouts**



1024 pixel columns

512 rows

Readout (zero suppression)

Bias, Readout, Control
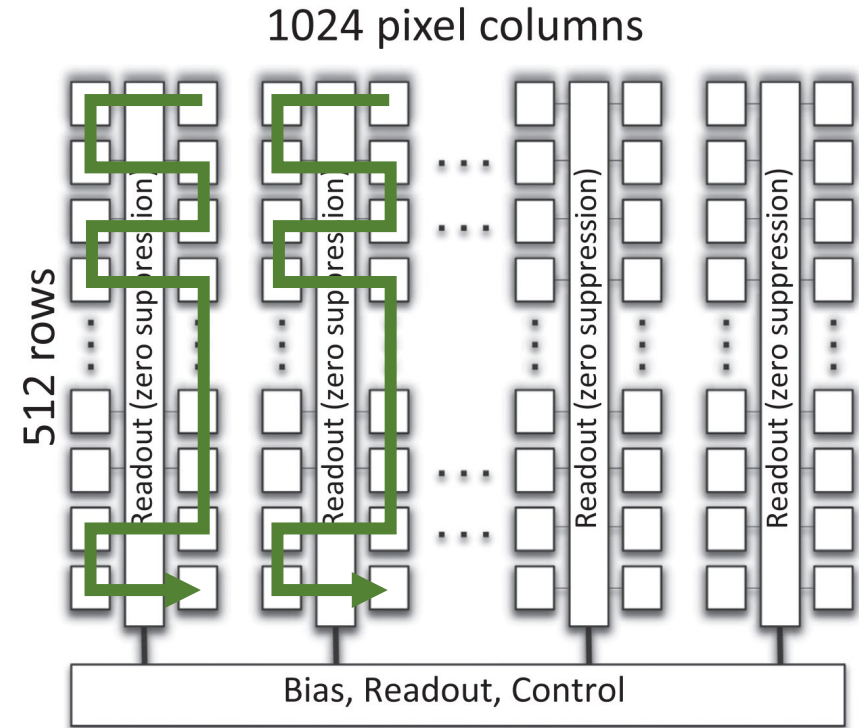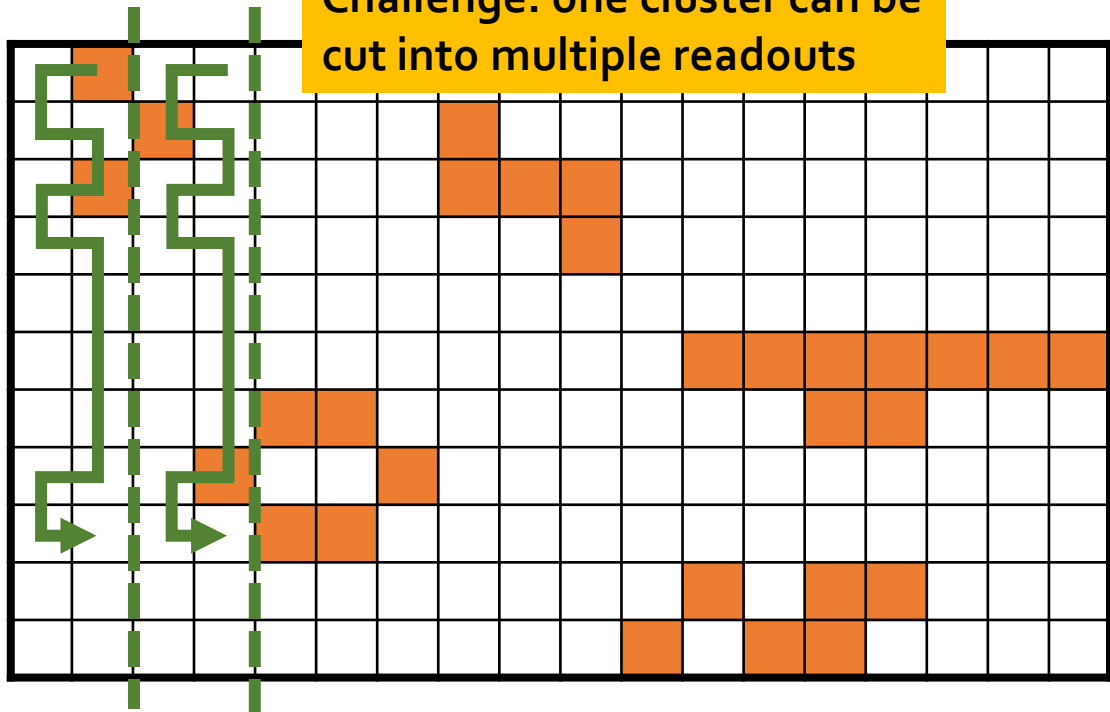
- **Requirements**
  - **Cluster size.** Cluster size (fired pixels) can be any value between **1** up to **128** pixels
    - In reality, we can limit to **16**; but ideally, set it as a parameter to be adaptive to future algorithms
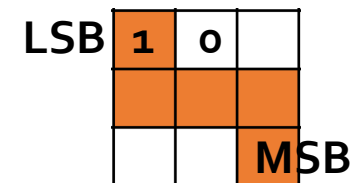    - Note: the bounding box of a cluster can be larger than 16

  - **Outputs.** Output the **center of mass** for each cluster and its **shape**
    - **Center of mass: <x, y>**
    - **Shape: <# of pixels, column, row, key>**

  - **Time constraint:** Less than **10 us per event**
    - End-to-end, including input/output
    - The faster, the better
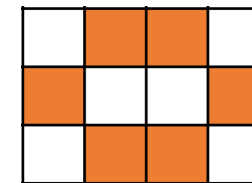    - Note: some events can have many clusters

  - Can change the key encoding if you have a better way

**LSB** | 1 | 0 |

**MSB**

<# of pixels, column, row, key>
**<5, 3, 3, 313>**
Key 313: 100111001

**<6, 4, 3, 1686>**
Key 1686: 011010010110
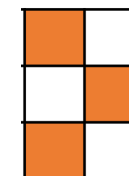
**<3, 2, 3, 25>**
Key 25: 011001

- **Finish all "events" as fast as possible but using as few resources as possible**
  - In real collider, minimum requirement is 10 us
  - The shorter the better – this is real problem!!

- **Input file format: .txt files**
  - You may change the file format (e.g., to binary) but not how the data is provided

- **Output format: not specified**

- **Also design your algorithm smartly – a true co-design problem**

Event no hits
```
1 68C0B
1 68C68
1 68CC5
1 68D21
1 68D7E
```

Hits: cluster them and output each cluster's information
```
0 308/0B6
0 309/0B6
0 30D/0B5
0 30C/0B5
0 30C/0B6
0 30D/0B6
1 68DDB
```

Hits: cluster them and output each cluster's information
```
0 309/0B5
0 308/0B6
0 309/0B6
0 30D/0B5
0 30C/0B5
0 30C/0B6
0 30D/0B6
```

Event no hits
```
1 68E38
1 68E94
1 68EF1
1 68F4E
```