Discussion primer on data transfer protocol

Lars Eklund

The ideas presented here are intended as a starting point to discuss and define the data transfer protocol.

By necessity, it has to make assumptions about on-chip processing algorithms and H/W design choices.

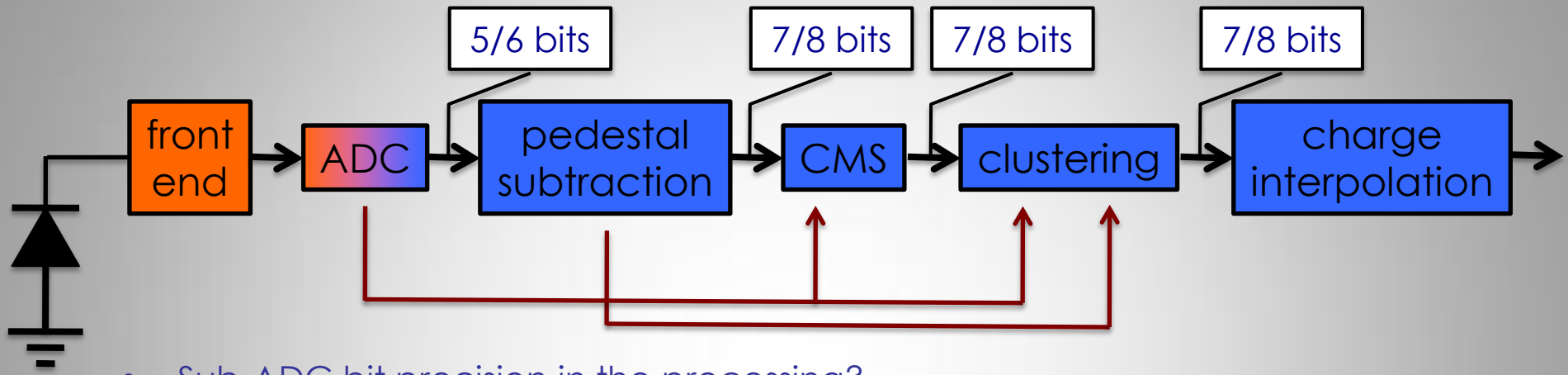
Hence this has to evolve with and feed in to the design specification of the chip

University of Glasgow

Some initial comments

- This presents a few different variants of the protocol
  - Obviously, only one or two of them (ZS+NZS) would be implemented
    - List all options and ideas and then discuss pros and cons
- Closely linked to the on-chip processing
  - Next slide shows an outline of the assumed processing
- Originally assumed 3-4 ADC bits (2008)
  - That would be sufficient for the pure charge amplitude
  - Assume 5-6 bit ADC here
    - Perhaps a couple of more bits used in the on-chip processing
    - Perhaps a few less bits transmitted
- The details of the formatting is a first attempt
  - We have to decide what information is needed offline
  - Exact format to be optimised as we firm things up
- A few options left out in the discussion here
  - No error coded included, probably useful e.g. as an extension of the 'tag'
  - No information about internal processing states
    - Probably better to convey that information via ECS

Assumed on-chip processing



- Sub-ADC bit precision in the processing?
- Possibility of bypassing processing steps?
  - Skip pedestal & CM subtraction
    - Keep number of ADC bits same in the data format
  - Skip clustering and/or charge interpolation
    - This gives non-zero suppressed data – separate data format
- Charge interpolation step optional
  - Would give sub-strip position (10 bits for 128 channel chip?)
- Read-out of ADC values optional
  - If read out perhaps use 4-5 bits?
- Include a high charge sum threshold ?

Data formats - overview

- Some basic assumptions
  - One serial link per FE chip (128 or 256 channels)
  - Only look at the 'pay-load'
- Three different variants of ZS format
  - Read-out of ADC values for strips above threshold
    - Does not require 'charge interpolation'
    - This is the variant used for the current data rate estimates
  - Read out cluster centre
    - Include charge sum or charge sum bit?
  - Read out cluster centre plus ADC values
    - Allows offline refinement of the interpolation
- A conceptual non-zero suppressed format

Transfer only ADC values (I)

- Do charge the interpolation offline
- Information to be conveyed
  - A bunch x-ing number per event (12 bits, reset every orbit)
  - Address within the area read out by the link (128/256 channels)
  - The ADC values of the hit strips
  - Tags that signals
    - New/end of/next event
    - Isolated hit
    - Consecutive hits (cluster)
    - End of buffer, no more events

Transfer only ADC values (II)

The following sequence reads out two events

- First one with one single strip cluster and one two strip cluster
- Second with one three strip cluster

<tag=N><BCnt><Addr><ADC><tag=I><Addr><ADC><tag=C><ADC><tag=N><BCnt><Addr>
<ADC><tag=C><ADC><tag=C><ADC><tag=E>

Legend:

- BCnt: 12 bit number, reset every orbit
- Addr: 7 bits, strip address of first strip in cluster (assumes 128 ch/chip)
- ADC: ADC value (4, 5 bits, truncated from processed values ?)
- Tag (2 bits), controls the type of data to follow
  - New/Next event : <N>=<11> (start of the event)
  - Consecutive hits: <C>=<10> (ADC values for adjacent strips to follow)
  - Isolated hit: <I>=<01> (new cluster to follow)
  - End of buffer: <E>=<00> (no more events in buffer)

Transfer only the cluster centre

- Charge interpolation on chip
- Information to be conveyed
  - A bunch x-ing number per event (12 bits, reset every orbit)
  - Cluster centre with sub-channel precision
    - E.g. 10 (11) bits for a 128 (256) channel chip
  - Add an additional bit on total ADC sum
    - Set it if the cluster sum is above some threshold
  - Tags that signals
    - New/end of/next event
    - Next cluster in event
    - End of buffer, no more events

Transfer only the cluster centre & ADC values

- Allows re-calculation of cluster centres off-line
  - Most costly version in terms of design and data volume
- Information to be conveyed
  - A bunch x-ing number per event (12 bits, reset every orbit)
  - Cluster centre with sub-channel precision
    - E.g. 10 (11) bits for a 128 (256) channel chip
  - ADC value per hit in the cluster
    - Similar to the format of only ADC transmission
    - Addresses of the clusters can be inferred from cluster centre and ADC values
  - Tags that signals
    - New/end of/next event
    - Isolated hit
    - Consecutive hits
    - End of buffer, no more events

Non-zero suppressed format

- Very useful tool for debugging and characterising the system
  - Requires a completely separate data format
    - Data output before clustering (full 6 or 8 bits)
    - Ideally with or without the CM and pedestal subtraction performed
  - Only possible at low trigger rate
- Information to be conveyed
  - Bunch x-ing number 12 bits
  - Array of 128 (256) ADC values, 6 or 8 bits each
  - Conceptually the simplest format

Summary

- Some early ideas on the data format presented
- We should decide on the information needed off-detector
  - To decode and build the events
    - And to recover from errors !
  - To determine the hit position
- Several different options presented
  - Put all the ideas on the table
  - Then reduce it to a design spec
- Comments and suggestions are welcome