

# MCH CRU Firmware and Data Format

**C. Flouzat**  
CEA/DRF/IRFU

- **Data Format**
- **MCH specific developments for CRU**
- **Work Ongoing**

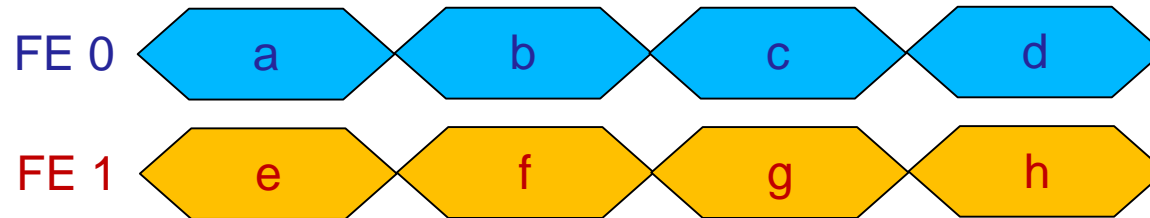
# Data Format

# Raw Data out of the GBTx

MCH uses the GBTx data format with only 80 bits (i.e. no wide bus mode)

→ Each frontend board uses 2 bits inside the 80 bits

79	78	...	...	...	...	3	2	1	0
						e	f	a	b
						g	h	c	d



*Data are sent LSB first*

## SAMPA Packet Format

### SYNC Packets

Identified 50 bit (to find the start of packets within the serial stream)

### HB Packets

50 bits with a given packet type (3 bit) and the BX counter value sampled for pulse

### Data Packets

50 bits followed by up to 1000 10 bit packets → variable length  
many types encoded with packet ID (3 bits)

→ Format doesn't allow to lose synchronization (or we lose the start of the packets)

## Trigger Mode

SYNC packets received when Idle : easy to synchronize, even offline

watch for a 50 bit identified packet

## Continuous Mode

SYNC packets received only if no relevant data seen by Front End Chip

→ need to have a machine to keep synchronized onboard FPGA (GRORC/CRU)

Drop the SYNC packets to save bandwidth

## Synchronization between SAMPA chips and ALICE

With the HB packet: a counter value is chopped when receiving a HeartBeat pulse

This value is inserted in the data stream

A chip ID + Bunch Crossing Counter is encoded inside all data packets

No Physics Trigger information

Work needed to decide the reaction system if BX value doesn't correspond bw FEC's

## Chip identification

A chip ID for each ribbon is encoded

CRU has to add the elink number and the fiber number

# CRU Firmware: MCH specific design

## Startup or Reset

Find the 50 bits of the SYNC packets

## Along Acquisition

```

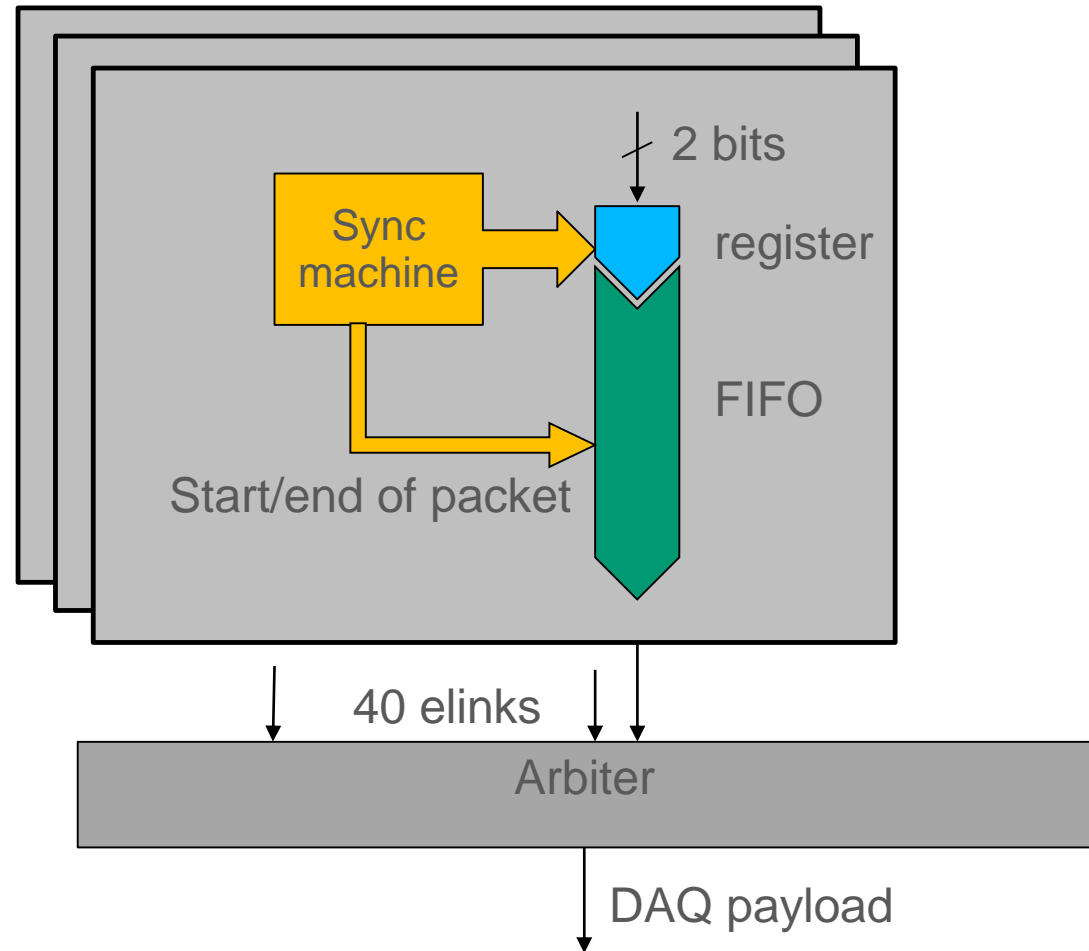
if packet type is SYNC
  → drop it
else if packet type is DATAHDR
  read nb_words

  push StartOfPacket

  loop
    push data
  until all nb_words were pushed

  push EndOfPacket
else
  push StartOfPacket
  push packet
  push EndOfPacket

```





**Consolidate this proposal with O2 team**

**Implement this module on current GRORC platform**

Discussed with Filippo Costa.

Seems feasible before summer 2017

**Choose the right memory size**

Either huge FIFO for each elink

Or huge FIFO after Arbiter

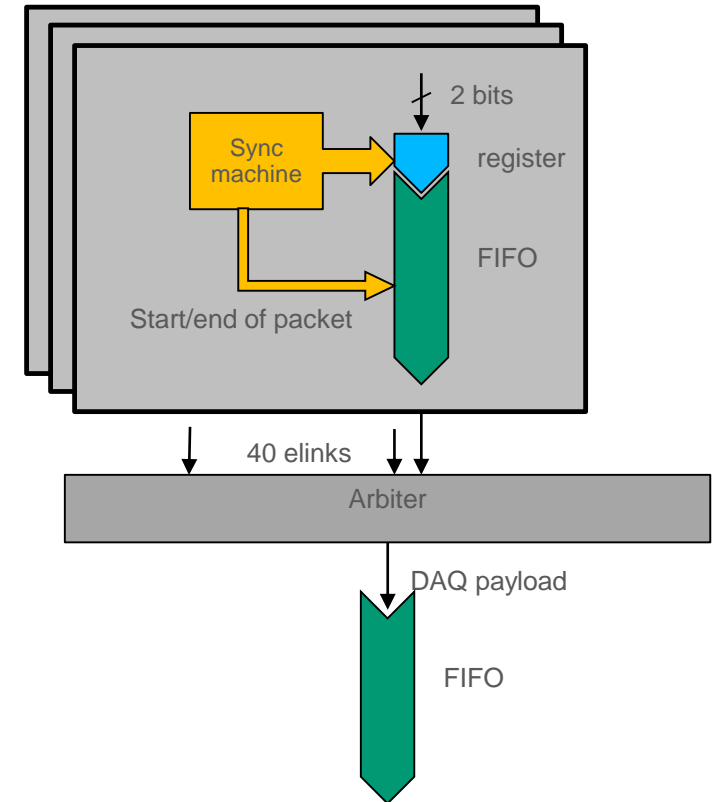
Simpler system (10 bit packets)

Tradeoff to be found with data overhead involved

**Remaining room on CRU's FPGA?**

**Order the necessary computer and software**

To be ready as soon as we get a CRU prototype



Thank you for your attention.

Questions?