Progress on software implementation of RD53B data encoding/decoding

Maurice Garcia-Sciveres, Timon Heim, Hongtao Yang

Lawrence Berkeley National Lab and University of California

Instrumentation weekly meeting

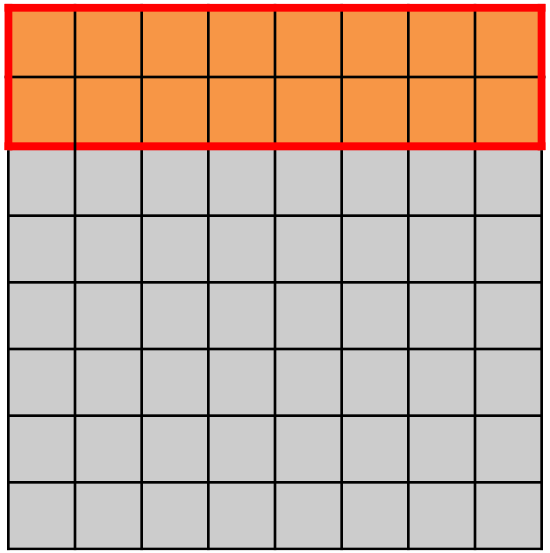
Mar 27, 2020

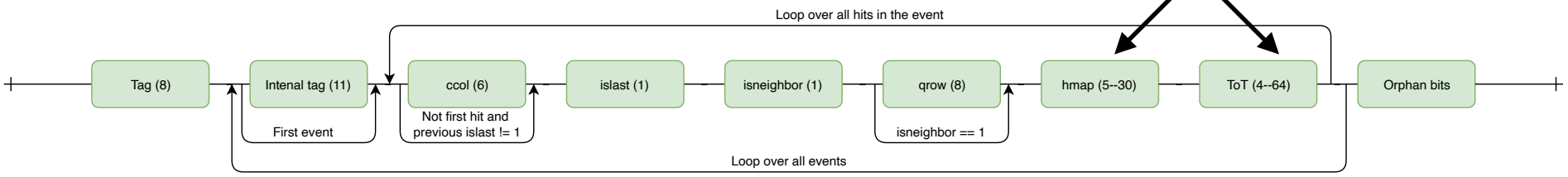
Introduction

- RD53B data stream has variable instead of fixed size
  - Due to stream compression (limited bandwidth)
  - Add great complexity to data decoding
- Goal: implement RD53B data encoding/decoding in SW
  - Encoding part will be used for development and validation of decoding, and eventually become part of SW emulator
    - Performance not a big concern
  - Decoding part will eventually deal with real data from RD53B chips, and be compared with FW approach
    - Performance is crucial
- Today we will discuss the flowcharts for the encoding/ decoding algorithms (as preparation for implementation)

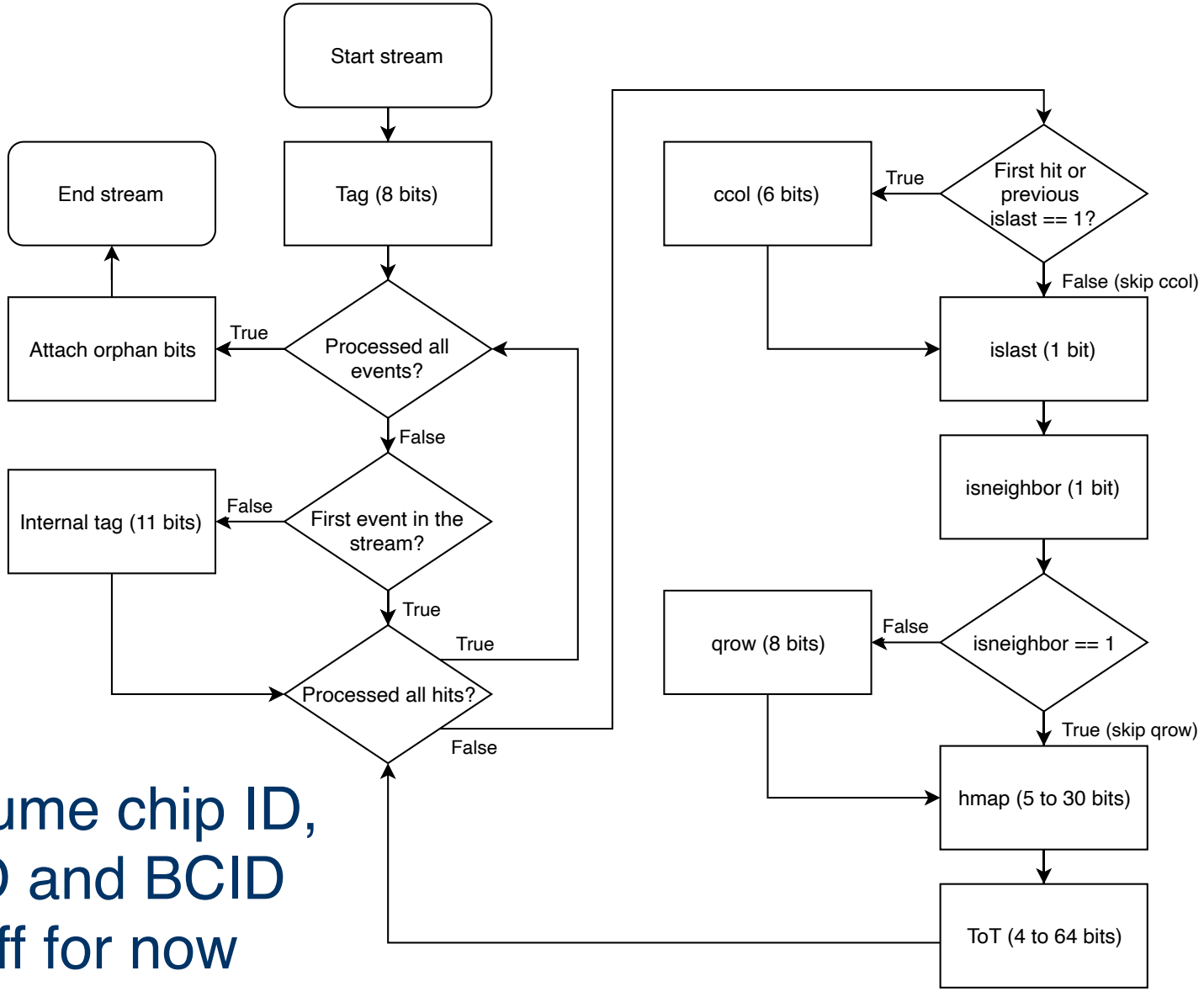
Basics

- A stream could contain one or more events. Each event:
  - Is identified by a tag, and possibly also L1ID and BCID
  - May contain 0 or more pixel hits
- With core column (ccol) and quarter row (qrow) indices, one can uniquely locate a qrow, which is the basic unit for saving hit maps (hmap) and time-over-threshold (ToT)
  - A “core” in RD53B chip is a 8×8 pixel matrix.
  - A core can be further divided into four (qrow), each consists of 2×8 pixels





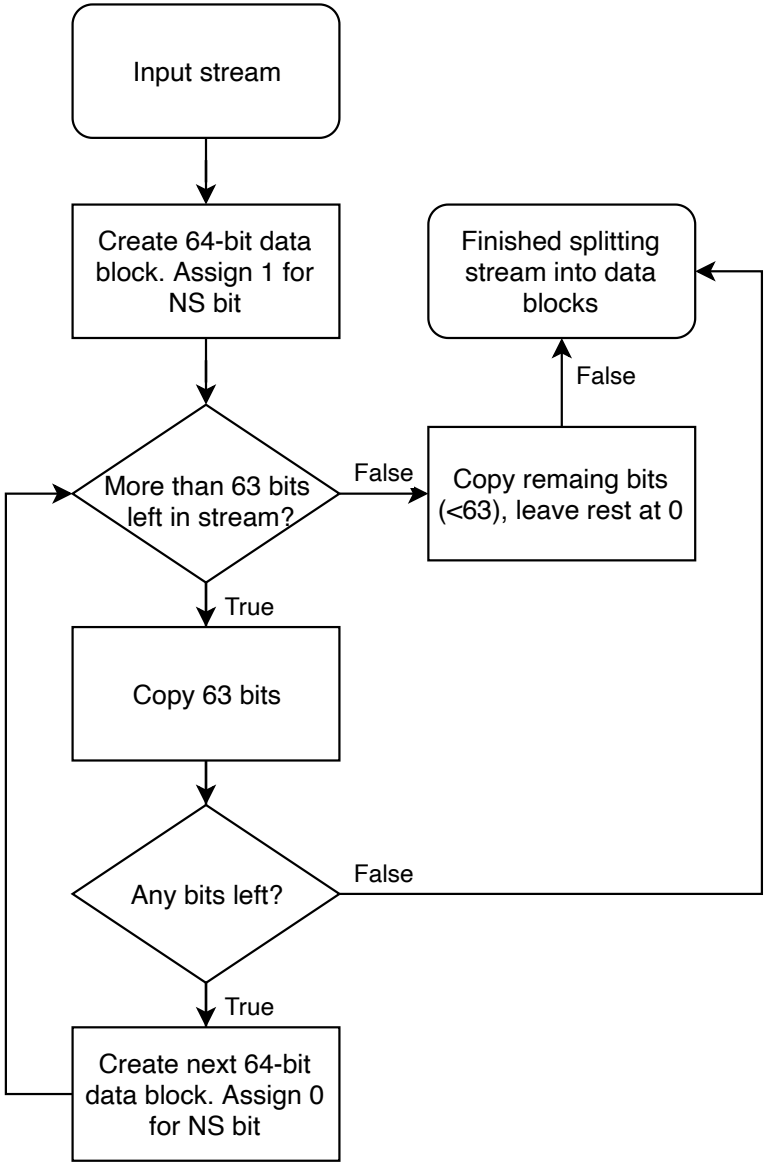
Encoding: build one stream



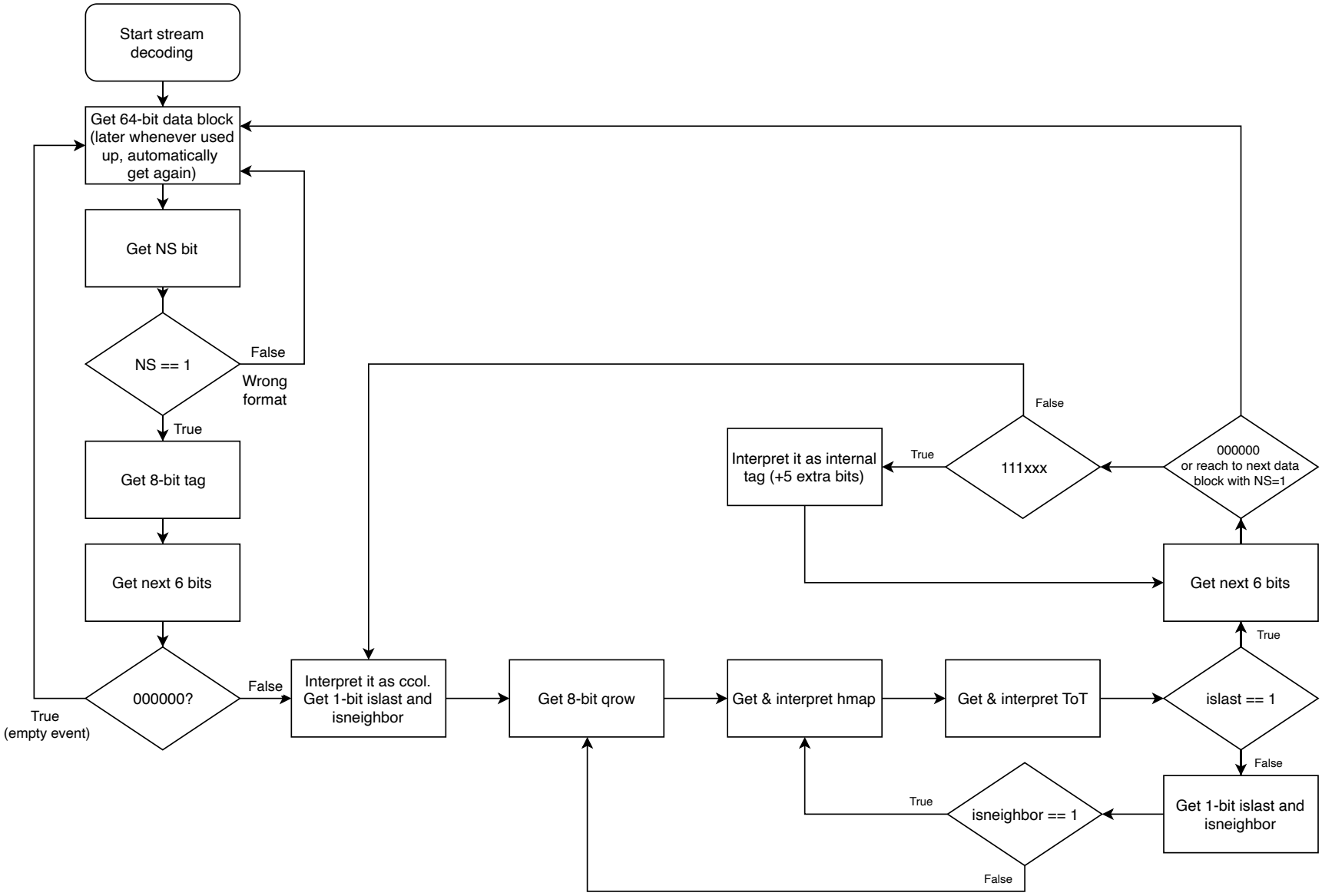
- Assume chip ID, L1ID and BCID all off for now

Encoding: split stream into 64-bit data blocks

- In SW the first bit of a 64-bit block will always be New Stream (NS) bit
  - The first block of the stream has NS=1
  - The trailing blocks, if any, have NS=0
  - Chip ID, if exist, will follow NS bit
  - L1ID and BCID will follow the event tag



Decoding


Start stream decoding
Get 64-bit data block (later whenever used up, automatically get again)
Get NS bit
NS == 1
False
Wrong format
True
Get 8-bit tag
Get next 6 bits
000000?
True (empty event)
False
Interpret it as ccol. Get 1-bit islast and isneighbor
Get 8-bit qrow
Get & interpret hmap
Get & interpret ToT
islast == 1
True
False
Get 1-bit islast and isneighbor
isneighbor == 1
True
False
Get next 6 bits
000000 or reach to next data block with NS=1
111xxx
True
Interpret it as internal tag (+5 extra bits)
False

Decoding: discussions

- In decoding process, the software is receiving a continuous flow of data blocks, which could span over multiple streams
  - Use next NS == 1 OR orphan bits (000000) to decide the end of current stream
- The data processing code for RD53A in YARR is a good starting point
- Minimal sanity check during decoding to ensure performance

Next steps

- Implement encoding program, which will provide a stream for decoding
  - Largely recycling Noemi's code for now. Improvements can be made later
- Implement data structure for RD53B data block
  - Will need to retrieve next data block once the current one is gone through
    - Could happen in the middle of decoding! A simple "for" loop over fixed-size blocks will thus no longer work
    - Basic idea: move the index of last unprocessed bit along the stream