

# RD53C Readout

## For Region of Interest (ROI) Triggering

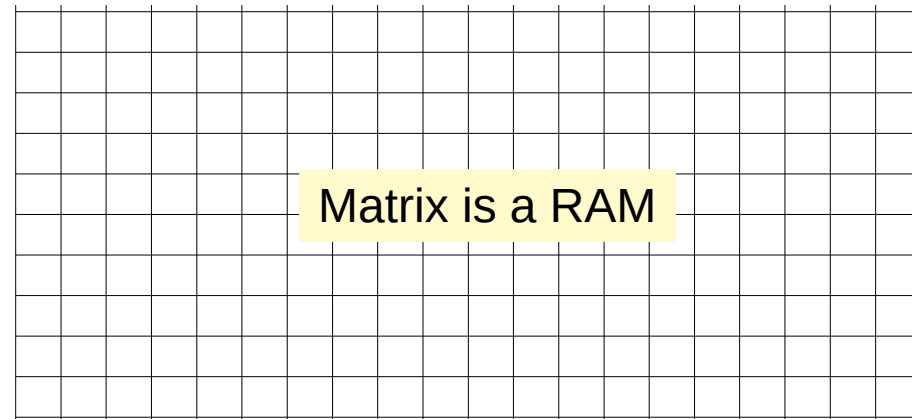
M. Garcia-Sciveres, T. Hemperek  
4 Sept. 2020

# Reminder of how RD53B works

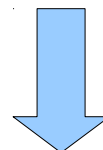
Events in matrix time-stamped with BCID

Triggers select which BCID's to hold-on to

Trigger processing tells core columns when to read them



Data formatting is pipelined in a single server queue



High speed serial out

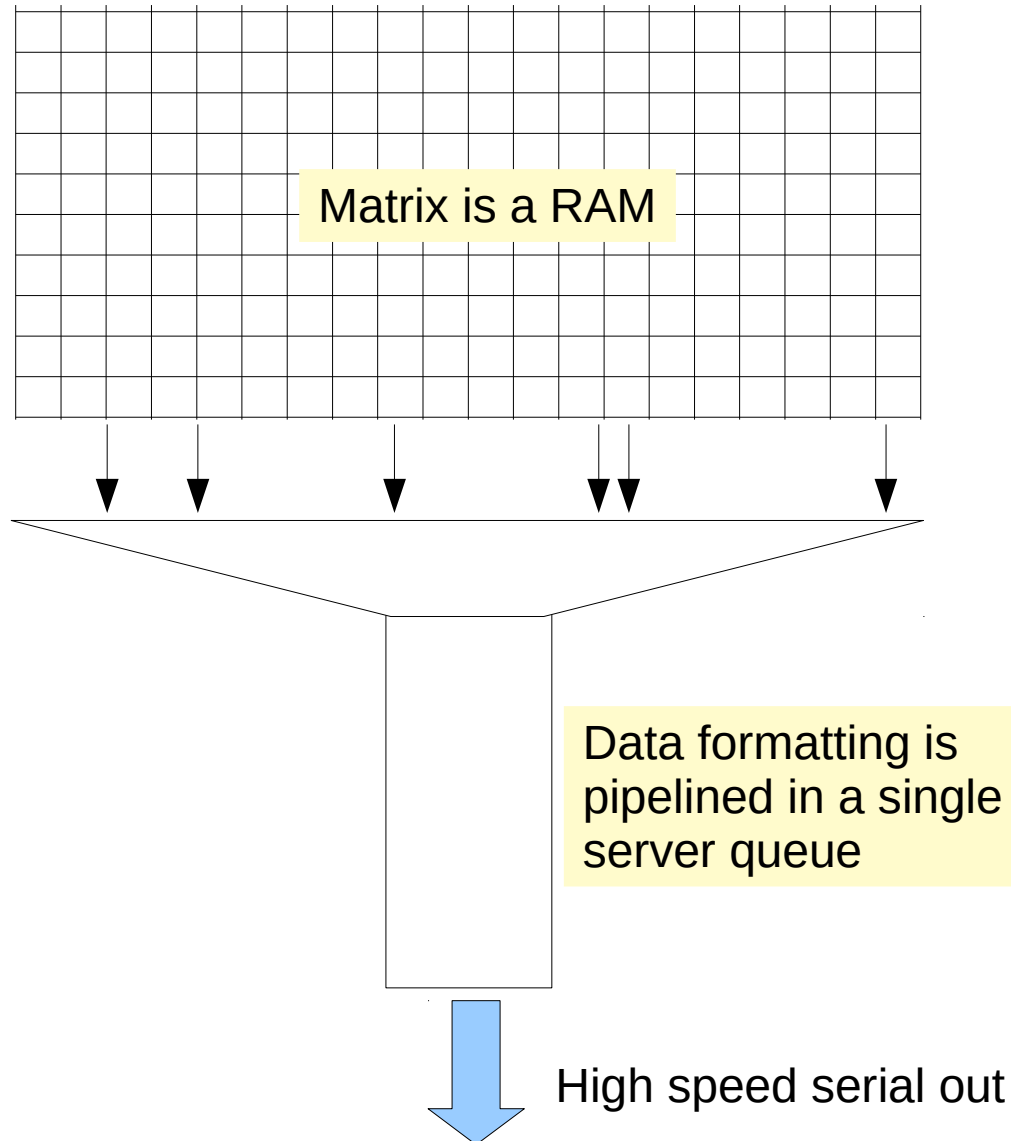
# Proposal for RD53C

Events in matrix time-stamped with BCID

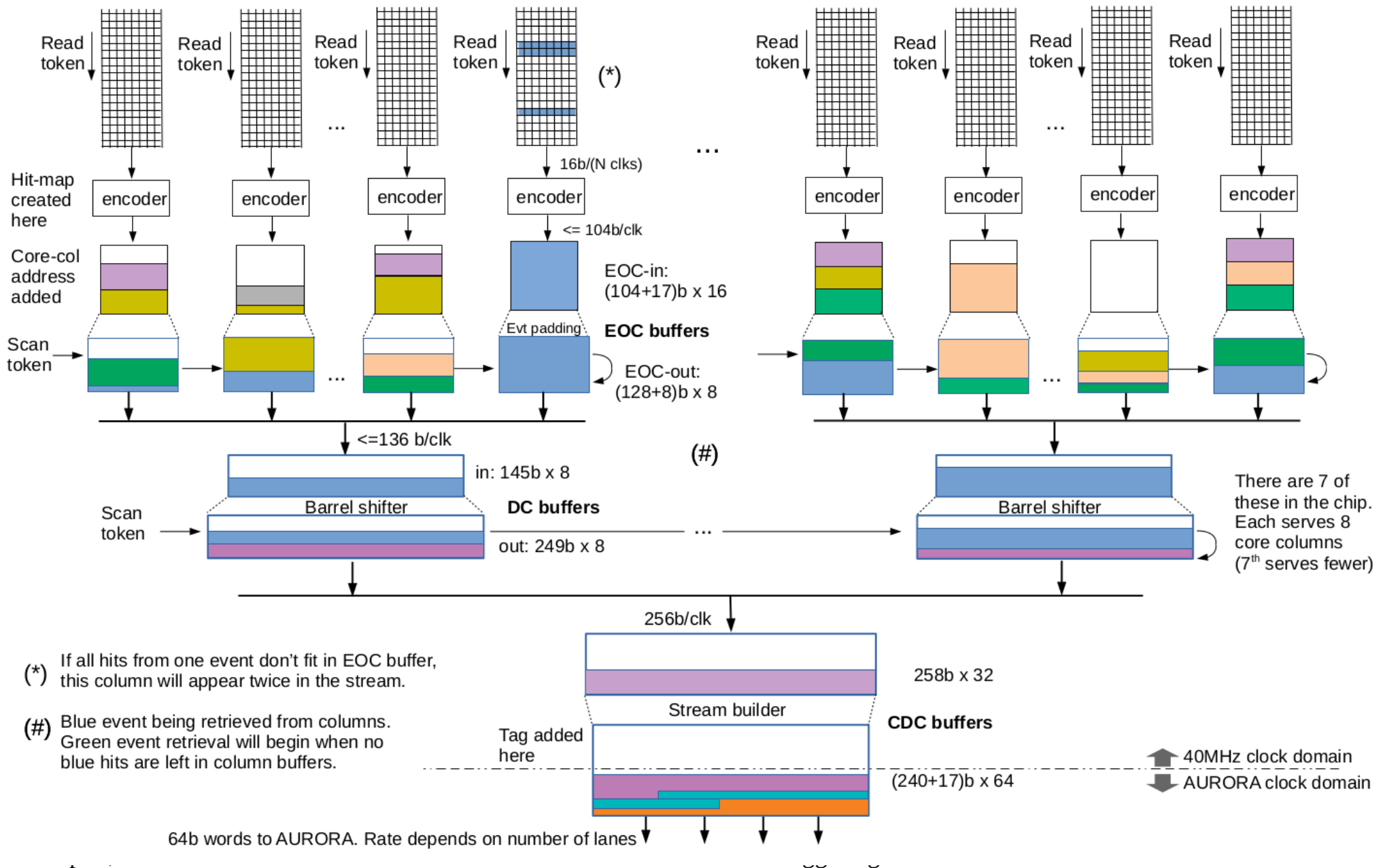
Triggers select which BCID's to hold-on to

Trigger processing tells core columns when to read them

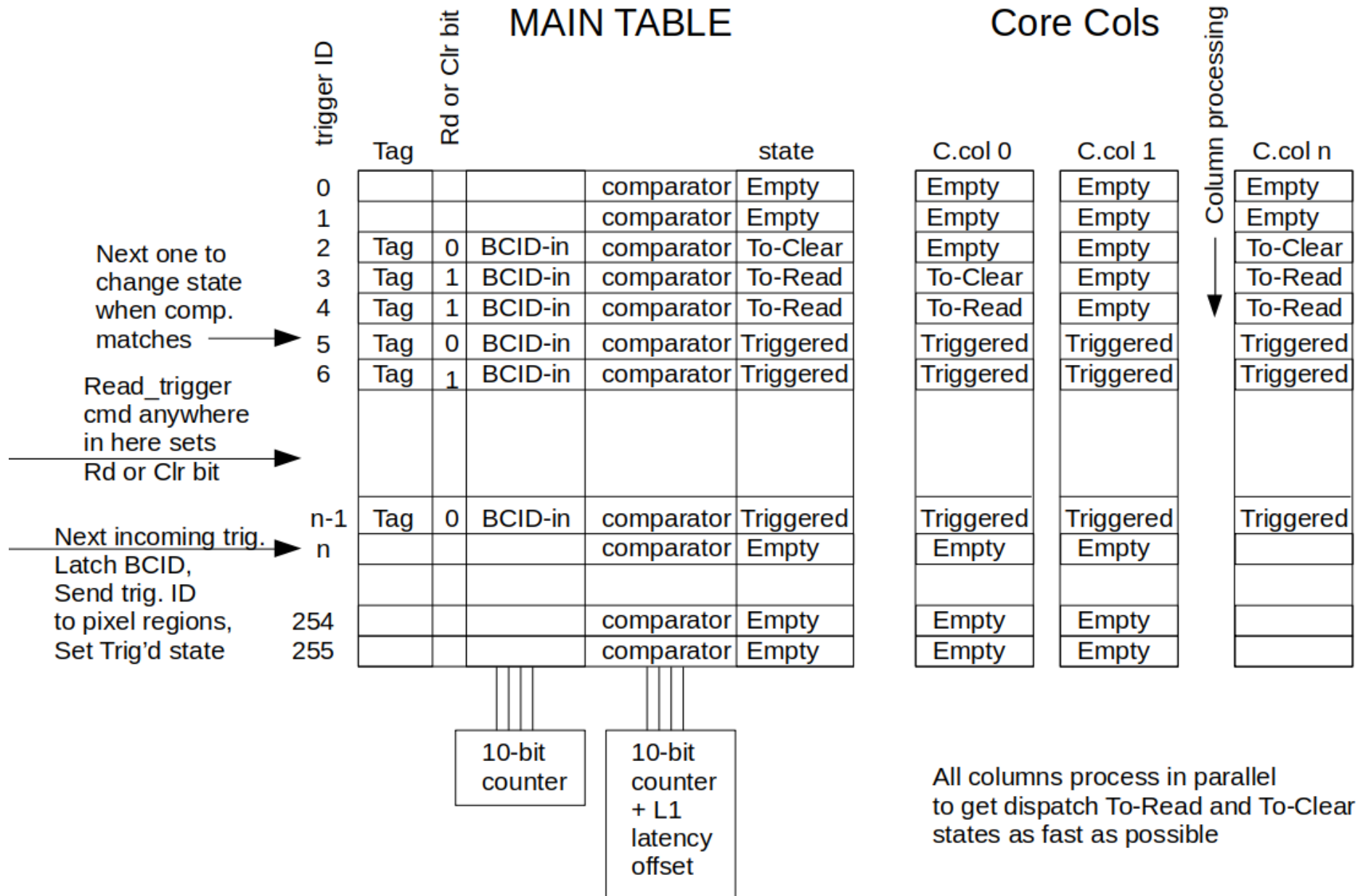
**Change only this**



# Don't touch any of this (& don't touch the core)



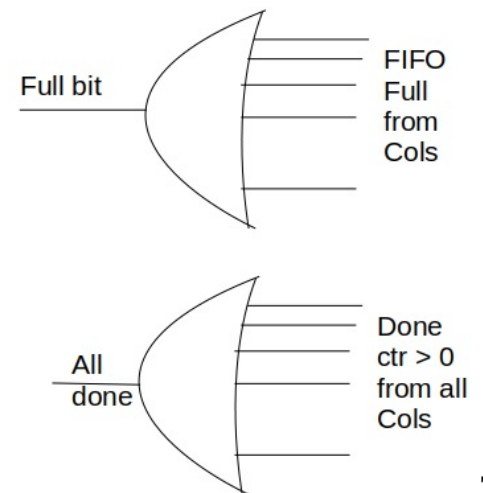
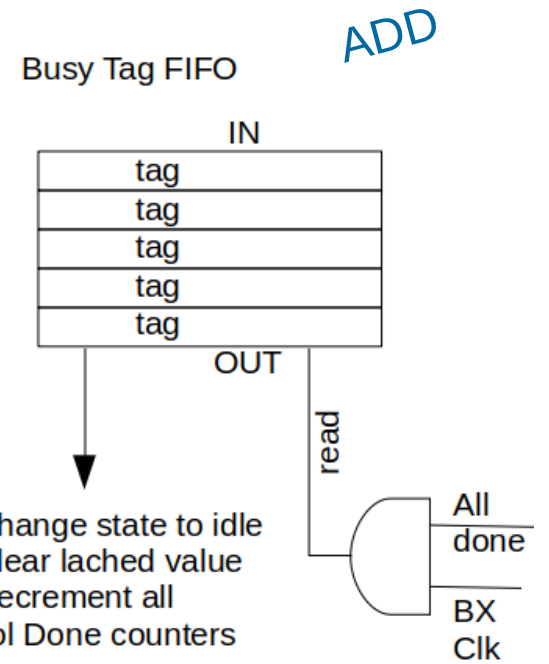
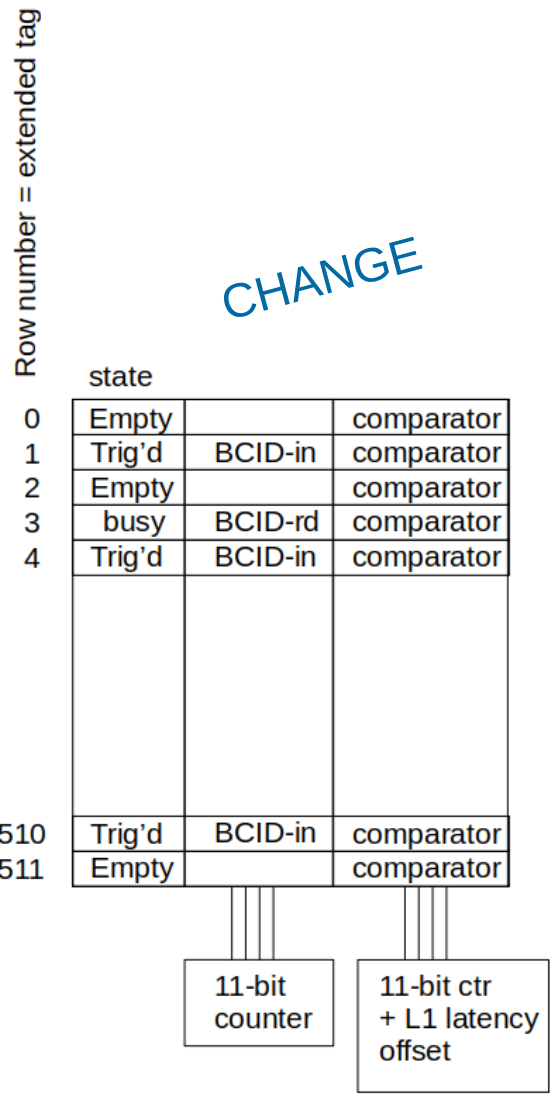
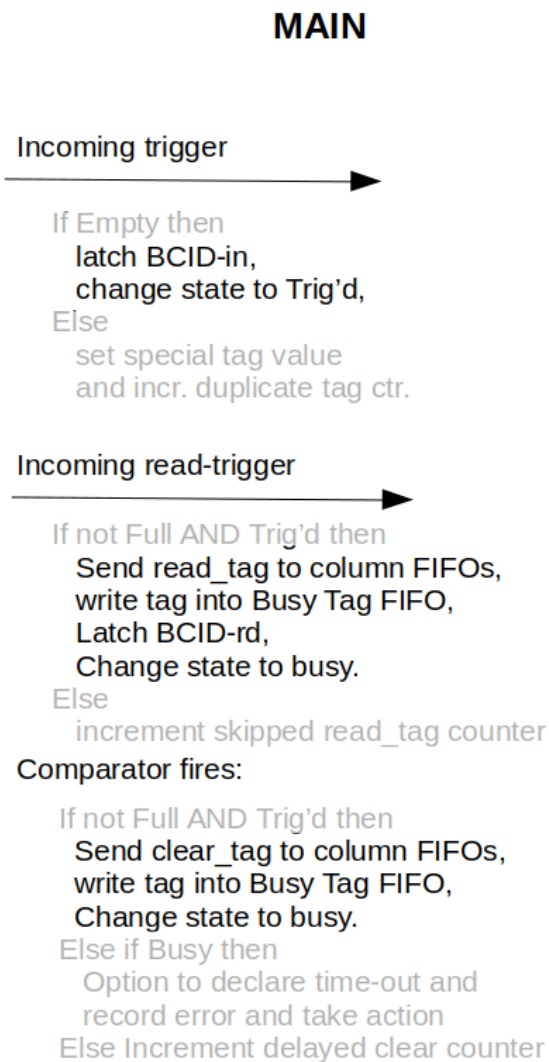
# Review of RD53B trigger processing



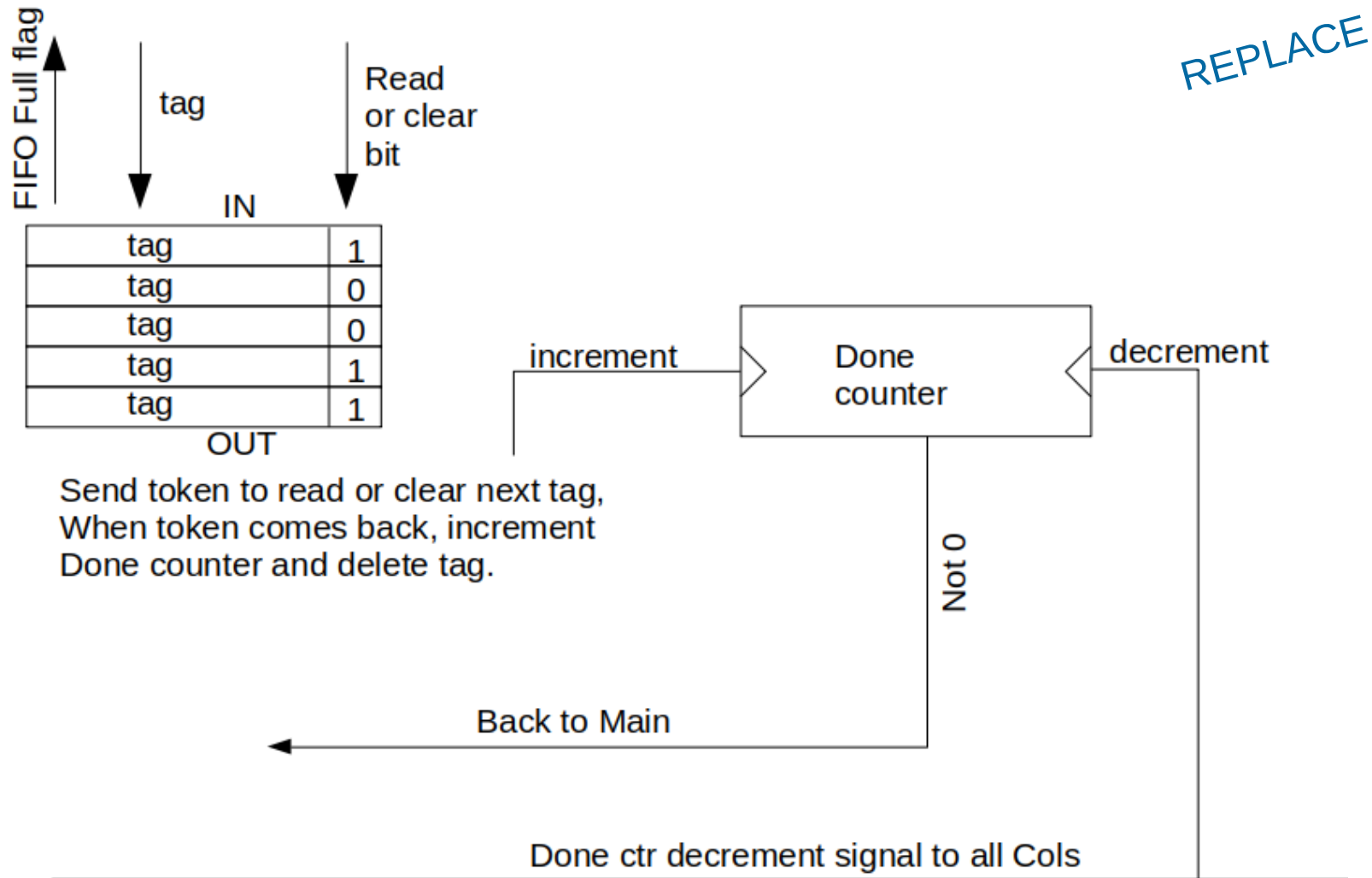
# Functional change

- RD53B processing order is set by trigger arrival.
- Doesn't matter if a trigger will be read out or cleared: the order is predetermined.
- This keeps things simple.
- Read\_trigger command just sets a flag bit-has no read action.
  
- For region of interest triggering, need to read out selected triggers before others
- Processing order for triggers to be read out must follow the order of read\_trigger commands
  - For triggers to be cleared the order will still be that of trigger arrival (there is no clear\_trigger command)
- Read\_trigger command now has read action

# New trigger book-keeping for RD53C



# New trigger book-keeping: each Core Col now has a FIFO and a counter





# List of design tasks/changes

1. Add a 9th bit to the extended tag. Bit value determined by the trigger command used. The bit value is 0 if the command is for reading a single BC, or 1 if the command is for reading multiple BC's. The new bit should be the MSb so that the 8 LSB's are exactly the RD53B tag, for backwards compatibility.
2. Change data label in regions from internal trigger counter value to 9-bit extended tag. The internal trigger counter value will no longer be used in RD53C.
3. Change the tag storage and timeout mechanism. The proposed implementation is a Main scoreboard with 512 rows plus a Main FIFO the same depth as the column FIFOs (see 4). This is shown in Fig. 2.6. The 9-bit extended tag is the row number of the scoreboard. The scoreboard keeps track of the state of each possible tag (each row). There are 3 possible states: *Idle*, *Active* or *Busy*. The scoreboard also keeps track of how long each tag has been Active or Busy.
  - 3.1 When a L0 trigger arrives, the corresponding tag (scoreboard row) is set to Active, the BCID-in timestamp of arrival is latched, and the tag and trigger are sent to all the pixel regions, where they will protect any hits that occurred L0 latency ago.
  - 3.2 Later, if a read\_trigger arrives, the referenced tag (=row) is set to Busy and the tag is sent to all core end of columns (EOC) in read-then-clear mode, as well as to the Main FIFO. The stored timestamp is re-latched to BCID-rd (the time of arrival of the read\_trigger command)
  - 3.3 At the point when the stored timestamp matches the L1 latency programmed in the chip (L0+L1 in TDAQ naming), if the state is still active (which means no read\_trigger command ever arrived) the state is changed to Busy and the tag is sent to all EOCs in clear-only mode as well as to the Main FIFO. The stored timestamp is reset. The reset value should be such that the programmed latency value never matches it (for example all 1's and counters wrap around at 1111111110).
  - 3.4 The sole job of the main FIFO is to decide which tags are finished processing in the matrix and EOC's and, therefore, can be recycled. Tags are pulled from the output of the Main FIFO (up to once per BX) whenever all EOC's show that they have finished processing at least one new tag. If there is even one column showing 0 new tags processed, then it waits. Note the order of the tags is the same in the Main FIFO as in all the column FIFOs. When a tag is pulled from the FIFO, its state is set to Idle in the scoreboard, and a decrement signal is sent to all EOC's. The stored timestamp is reset as in (c).
4. Change EOC from RD53B circular buffer storing flag bits (currently 512 bits total) to a FIFO storing tags plus an up-down *Done* counter showing number of new tags finished processing. This is shown in Fig. 2.7. Propose to store 64 tags. This is more than the present 512 bits, but is expected to fit. To be double-checked that 64 is deep enough to never be reached in normal operation. Each FIFO word is 10 bits (9-bit tag plus one flag bit).
  - 4.1 The EOC FIFO stores tags plus one bit specifying a read-then-clear or clear-only action for that tag. The EOC processing pulls tags from the FIFO output as fast as it can process them- it does not wait for anything. If the FIFO fills up, it asserts a FIFO-full flag that the main scoreboard process monitors. This prevents more tags from being sent to all EOCs until the FIFO is no longer full (see point 4).
  - 4.2 Each time a tag is finished processing, the Done counter is incremented. The Done counter is decremented by the global decrement signal coming from the main scoreboard control. The status of the Done counter (0 or not 0) is monitored by the main control. The counter stops at 0 (decrementing a value of 0 results in 0 again) and at counter max (incrementing counter max results in counter max). Counter max should be also 64.
5. Exceptions are to be handled in fairly simple ways and counted. A duplicate tag counter already exists in RD53B, but now need to add three more error counters: skipped read\_trigger, delayed\_clear, and read\_timeout.
  - 5.1 When a trigger arrives and the corresponding tag is already Active or Busy (not Idle), then a special tag number is allocated (not possible to produce with one of the 54 tag bases), as in RD53B, and the duplicate tag counter is incremented. The corresponding error flag is asserted. If the DAQ is paying attention to this error it could in principle send a read\_trigger with this special tag number and retrieve the data.
  - 5.2 When a read\_trigger arrives and the corresponding tag is not Active, the read\_trigger is ignored and the new skipped read\_trigger counter is incremented. The corresponding error flag is asserted.
  - 5.3 When a read\_trigger arrives and at least one of the EOC's FIFO full flag is asserted, the read\_trigger is ignored and the new skipped read\_trigger counter is incremented. The corresponding error flag is asserted.
  - 5.4 When a scoreboard Active tag reaches the L1 latency (point 2c above), but at least one of the EOC's FIFO full flag is asserted, none of the 2c actions will be taken. Instead, the new delayed\_clear counter is incremented. Note that the tag will remain in the pixel regions longer than normal (past the L1 latency). But eventually the latency counter will come back around and again trigger the 2c action as long as no FIFO full flag is asserted.

# List of design tasks/changes

1. Add a 9th bit to the extended tag. Bit value determined by the trigger command used. The bit value is 0 if the command is for reading a single BC, or 1 if the command is for reading multiple BC's. The new bit should be the MSb so that the 8 LSB's are exactly the RD53B tag, for backwards compatibility.
2. Change data label in regions from internal trigger counter value to 9-bit extended tag. The internal trigger counter value will no longer be used in RD53C.
3. Change the tag storage and timeout mechanism. The proposed implementation is a Main scoreboard with 512 rows plus a Main FIFO the same depth as the column FIFOs (see 4). This is shown in Fig. 2.6. The 9-bit extended tag is the row number of the scoreboard. The scoreboard keeps track of the state of each possible tag (each row). There are 3 possible states: *Idle*, *Active* or *Busy*. The scoreboard also keeps track of how long each tag has been Active or Busy.
  - 3.1 When a L0 trigger arrives, the corresponding tag (scoreboard row) is set to Active, the BCID-in timestamp of arrival is latched, and the tag and trigger are sent to all the pixel regions, where they will protect any hits that occurred L0 latency ago.
  - 3.2 Later, if a read\_trigger arrives, the referenced tag (=row) is set to Busy and the tag is sent to all core end of columns (EOC) in read-then-clear mode, as well as to the Main FIFO. The stored timestamp is re-latched to BCID-rd (the time of arrival of the read\_trigger command)
  - 3.3 At the point when the stored timestamp matches the L1 latency programmed in the chip (L0+L1 in TDAQ naming), if the state is still active (which means no read\_trigger command ever arrived) the state is changed to Busy and the tag is sent to all EOCs in clear-only mode as well as to the Main FIFO. The stored timestamp is reset. The reset value should be such that the programmed latency value never matches it (for example all 1's and counters wrap around at 1111111110).
  - 3.4 The sole job of the main FIFO is to decide which tags are finished processing in the matrix and EOC's and, therefore, can be recycled. Tags are pulled from the output of the Main FIFO (up to once per BX) whenever all EOC's show that they have finished processing at least one new tag. If there is even one column showing 0 new tags processed, then it waits. Note the order of the tags is the same in the Main FIFO as in all the column FIFOs. When a tag is pulled from the FIFO, its state is set to Idle in the scoreboard, and a decrement signal is sent to all EOC's. The stored timestamp is reset as in (c).
4. Change EOC from RD53B circular buffer storing flag bits (currently 512 bits total) to a FIFO storing tags plus an up-down *Done* counter showing number of new tags finished processing. This is shown in Fig. 2.7. Propose to store 64 tags. This is more than the present 512 bits, but is expected to fit. To be double-checked that 64 is deep enough to never be reached in normal operation. Each FIFO word is 10 bits (9-bit tag plus one flag bit).
  - 4.1 The EOC FIFO stores tags plus one bit specifying a read-then-clear or clear-only action for that tag. The EOC processing pulls tags from the FIFO output as fast as it can process them- it does not wait for anything. If the FIFO fills up, it asserts a FIFO-full flag that the main scoreboard process monitors. This prevents more tags from being sent to all EOCs until the FIFO is no longer full (see point 4).
  - 4.2 Each time a tag is finished processing, the Done counter is incremented. The Done counter is decremented by the global decrement signal coming from the main scoreboard control. The status of the Done counter (0 or not 0) is monitored by the main control. The counter stops at 0 (decrementing a value of 0 results in 0 again) and at counter max (incrementing counter max results in counter max). Counter max should be also 64.
5. Exceptions are to be handled in fairly simple ways and counted. A duplicate tag counter already exists in RD53B, but now need to add three more error counters: skipped read\_trigger, delayed\_clear, and read\_timeout.
  - 5.1 When a trigger arrives and the corresponding tag is already Active or Busy (not Idle), then a special tag number is allocated (not possible to produce with one of the 54 tag bases), as in RD53B, and the duplicate tag counter is incremented. The corresponding error flag is asserted. If the DAQ is paying attention to this error it could in principle send a read\_trigger with this special tag number and retrieve the data.
  - 5.2 When a read\_trigger arrives and the corresponding tag is not Active, the read\_trigger is ignored and the new skipped read\_trigger counter is incremented. The corresponding error flag is asserted.
  - 5.3 When a read\_trigger arrives and at least one of the EOC's FIFO full flag is asserted, the read\_trigger is ignored and the new skipped read\_trigger counter is incremented. The corresponding error flag is asserted.
  - 5.4 When a scoreboard Active tag reaches the L1 latency (point 2c above), but at least one of the EOC's FIFO full flag is asserted, none of the 2c actions will be taken. Instead, the new delayed\_clear counter is incremented. Note that the tag will remain in the pixel regions longer than normal (past the L1 latency). But eventually the latency counter will come back around and again trigger the 2c action as long as no FIFO full flag is asserted.