A possible streaming data collection protocol for MPGD

And may be for some other sub-detectors

Irakli Mandjavidze

Irfu, CEA Saclay
Gif-sur-Yvette, 91191
France

Internal
21/Dec/2022

Outlook

- Time framing and data association
- Summary
- Backup

- Few introductory words
  - added after much wider presentation from Jeff on Streaming model

→ Minimize data processing in frontends – radiation, power consumption

  - DAMs have "infinitely" more resources than FEBs
  - Computing PCs have "infinitely" more resources than DAMs

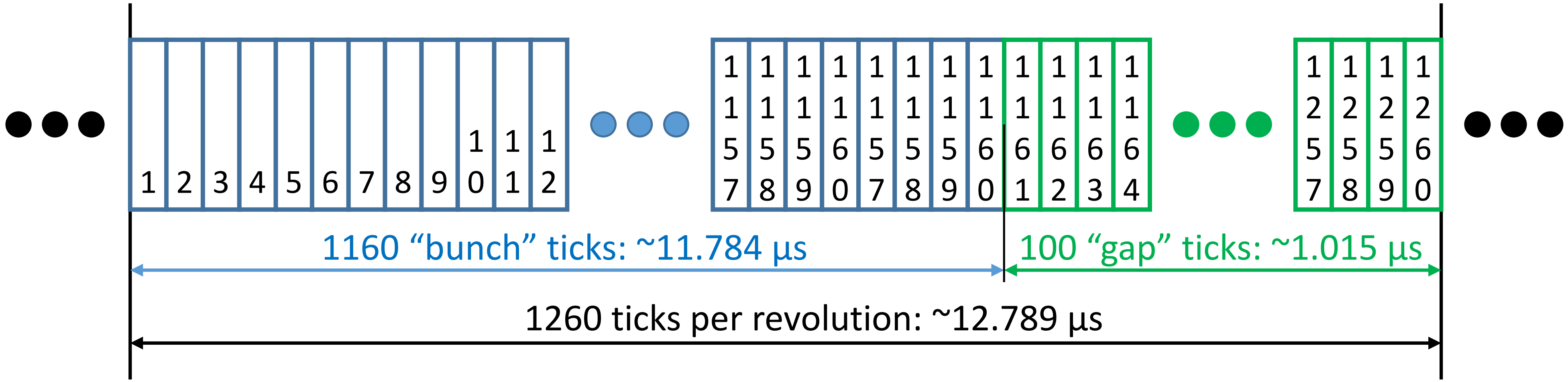
→ Move data out of detector as quickly as possible – radiation

  - Bring data to DAM
  - Dataflow from DAMs to computing PCs is not considered
    - Separate subject
  - Final data formats to be used in on-line or off-line data processing not considered
    - Data reshuffling in computing PCs or in DAMs if absolutely needed
  - Very frontend designer point of view
    - Use as many handles for early error detection (and correction) as possible

Beam structure and bunch crossing timing

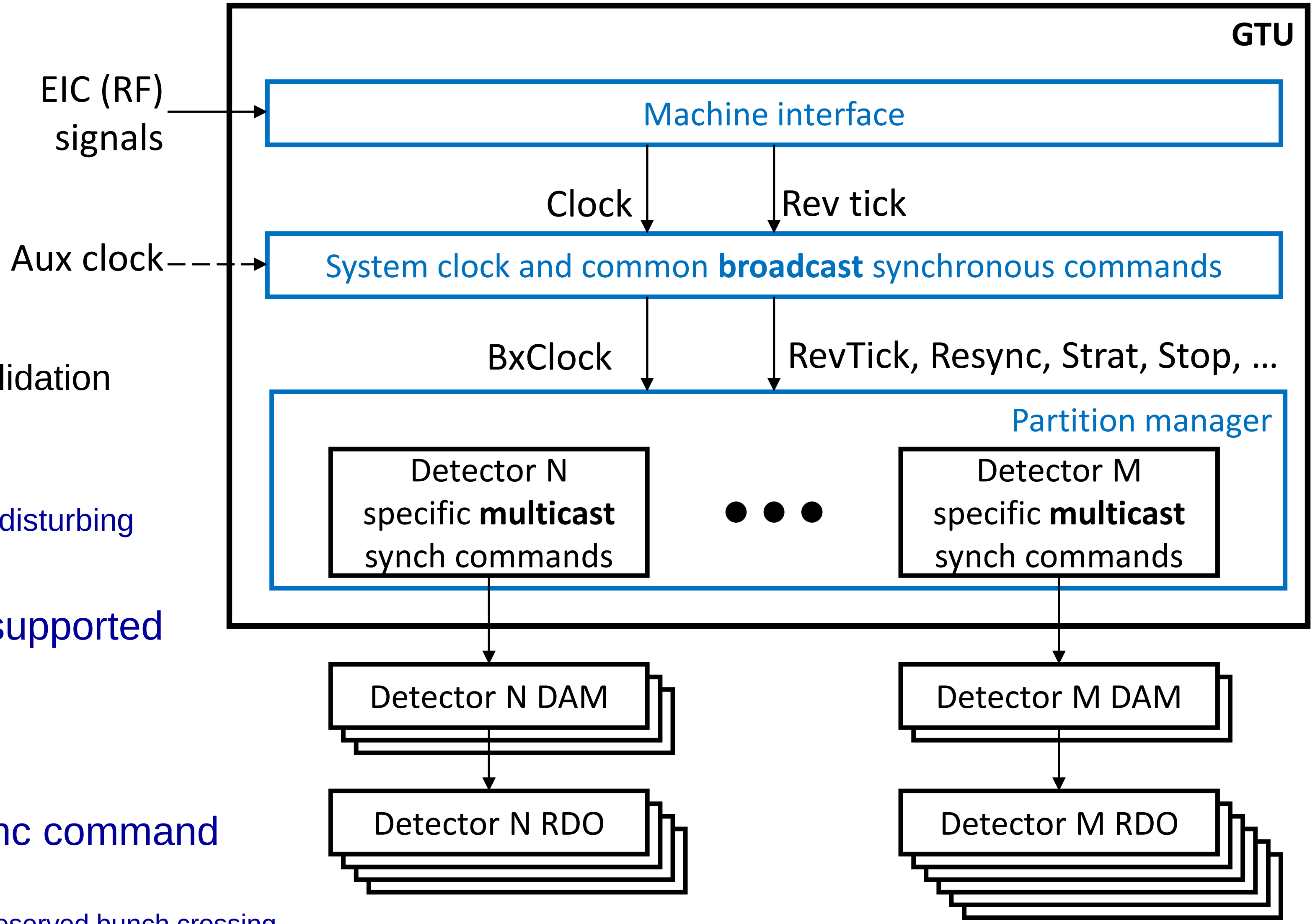
- Beam structure repeats every ~12.7886 µs
  - → Revolution frequency: ~78.195 kHz
- There are 1260 clock ticks in each revolution
  - → Clock period: ~10.14968 ns
    - ■ Frequency: ~98.52525 MHz
  - → 1160 “bunch” ticks *approximately* marking *potentially* active intervals with particles
    - ■ *Approximately*: electron beam structure is much more stable compared to hadron beam structure
    - ■ *Potentially*: there can be 290 or 1160 active bunches with particles
  - → 100 “gap” ticks without particles

Sep 8, 22



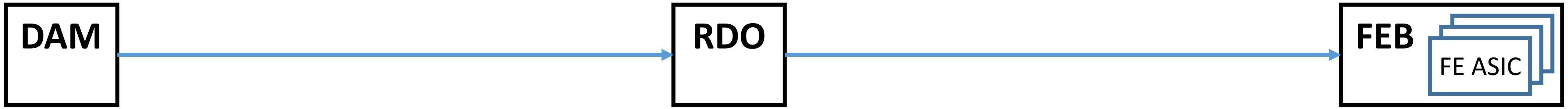
- Assume a stable clock can be derived from Machine with this frequency despite of bunch time variations
  - → This clock or its (sub)multiples is distributed to frontend electronics as “System clock”
    - ■ With bunch-phase recovery mechanism
    - ■ Used for bunch level synchronization, coarse timestamp bookkeeping, serial communication and possibly timing measurements.

Assumption on BX clock and synchronous command distribution

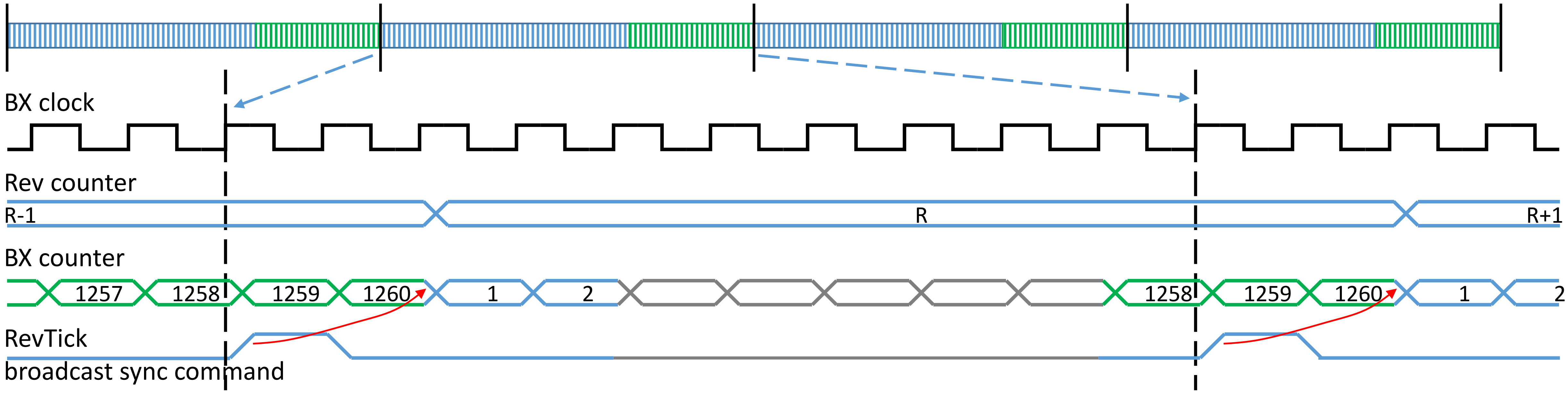


- Partitioning supported
  - → Independent data taking for testing, validation
    - *e.g.* cosmic runs
  - → Spying while the rest is taking data
    - *e.g.* performance, data quality without disturbing
- Detector specific multicast commands supported
  - → Calibration
    - *e.g.* take non-ZS data
- Identify and define protocol for each sync command
  - → Few examples in backup
    - *e.g.* Expected action @ frontend, particular reserved bunch crossing

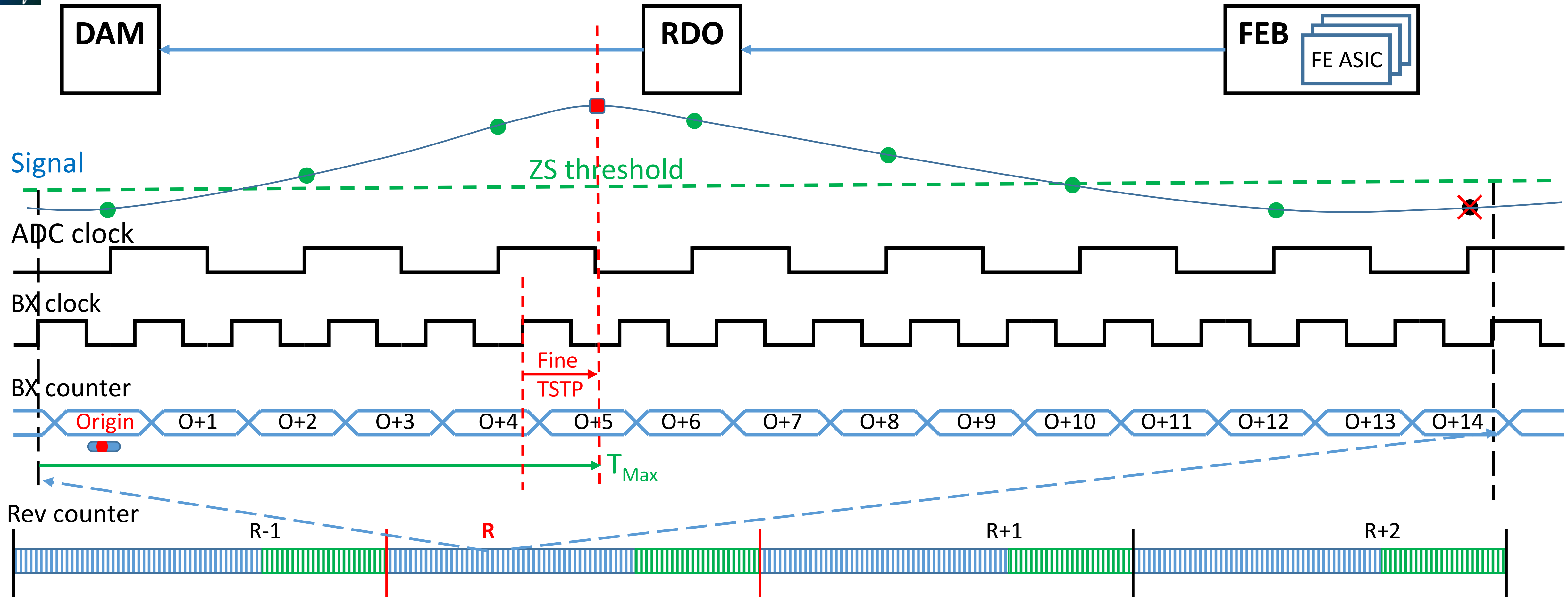
Synchro: GTU → DAM → RDO → FEB → ASIC



- At each revolution GTU generates RevTick broadcast **synchronous** command
  - → Distributed all the way down to FEBs and ASICs
    - ■ Used for synchronization and bookkeeping by DAMs, RDOs and possibly FEBs and ASICs
  - → Can be used as time frame delimiter for data fragment building and association
    - ■ Data fragment is marked by revolution numbers - in RDO or FEB
    - ■ Time association is performed based on bunch crossing counter / fine-time stamp within the revolution data - in DAM and/or upper level
  - → Can be considered as "Heartbeat"
    - ■ The "Heartbeat" response (if any) is **asynchronous** assisting in data fragment building and time association
      - Heartbeat acknowledgment produced after all data belonging to previous revolution have been sent out

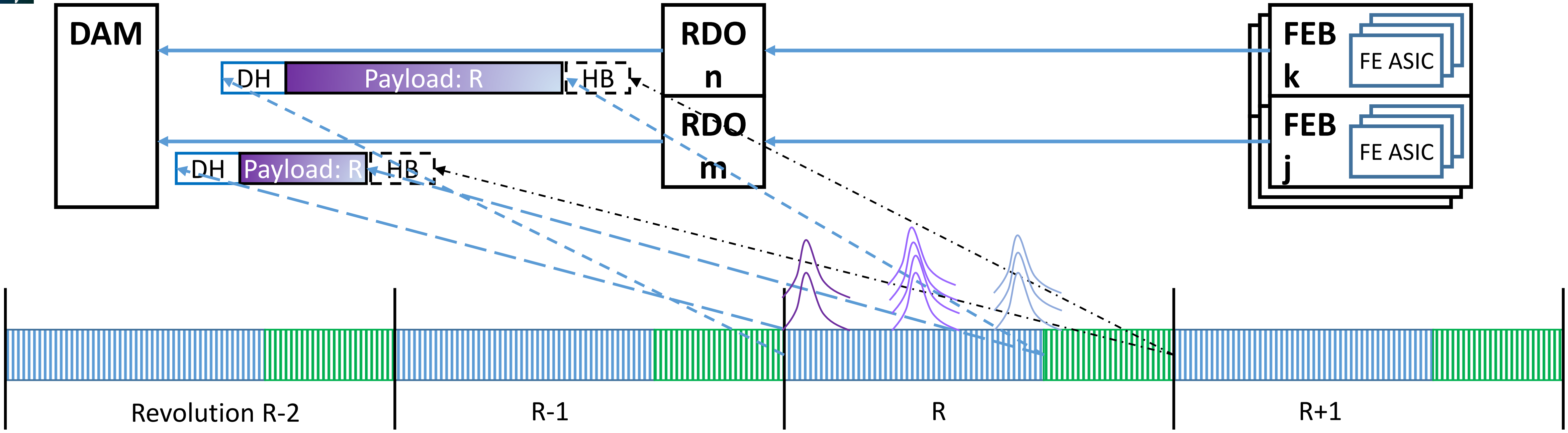


MPGD data: signal shape sampling with ADC



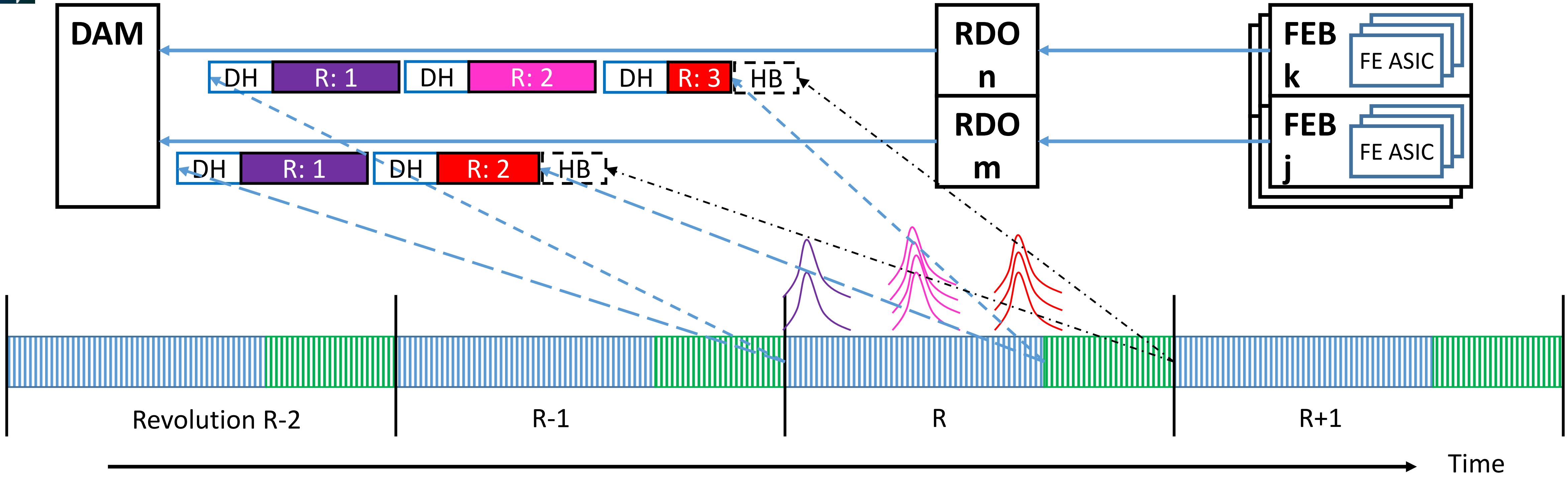
- ADC clock is derived from bunch crossing clock and is its (sub)multiple (e.g. ~50 MHz) or (M / N) fractional
  → Known frequency and phase relationship exists between the two clocks
- Signal association with originating bunch crossing possible
  → Max (as on example) or time of arrival (fitting samples on rising edge)
  → The set of Rev, BX and fine timestamp uniquely identifies the signal and makes association possible with other data

Physics data: RDO → DAM



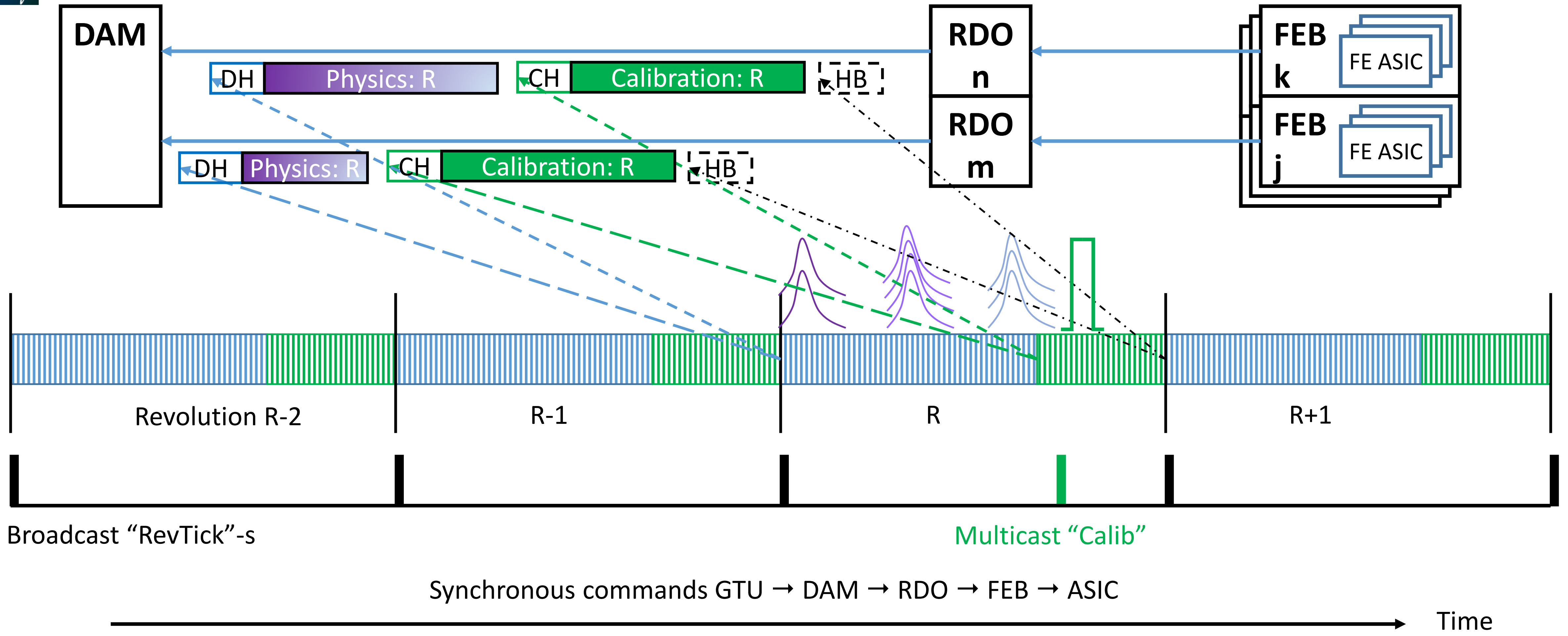
- RDO combines FEB data belonging to the same revolution into a data fragment and sends to DAM
  - → If needed, heartbeat (RevTick) acknowledgement is sent after the data belonging to the same revolution
  - → Revolution-level granularity might be handy for fragment building in RDO
    - A compromise between memory requirement, aggregation, latency
    - If too long, data fragments can be sent in a succession of packets
- DAM performs revolution record building based on revolution numbers embedded in RDO data
  - → Heartbeat packet can be used as indication that no more data is expected from RDO for this revolution
- In other words, consider the RevTick as a 78.195 kHz constant rate trigger
  - → Do classical event building in RDO and DAM with a readout window of ~12.7886 µs

Physics data: RDO → DAM with chopping of data fragments



- RDO can send revolution data fragment in a succession of packets, if fragments are too long
  - → Data packets are of variable length with max size
    - Whenever accumulated data reach max packet size limit, the packet is sent out
    - Last packet can be smaller
    - This guarantees a good use of the available link bandwidth
- DAM performs revolution record building based on revolution numbers embedded in RDO data
  - → Heartbeat packet can be used as indication that no more data is expected from RDO
  - → Or RDO may mark last packet as an end of fragment
  - → Or both, for robustness against packet loss and improved error detection

Physics and calibration data: RDO → DAM



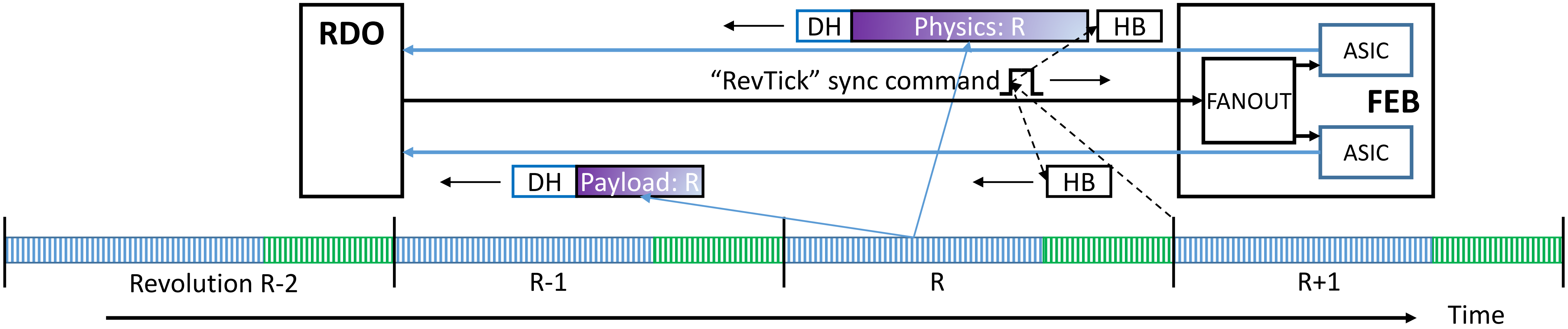
- Use multicast synchronous commands to perform special data acquisition
  - → A special “trigger” type, e.g. calibration
  - → Can be issued during the GAP ticks
  - → Same revolution record building mechanism
    - Dedicated header bits to mark special data and distinguish them from physics data

MpgdCalib

How to detect FEB revolution data closure in RDO

- Depends on on-FEB intelligence

- Case 1: no extra intelligence above the one embedded in ASICs
  - → The RevTick can be considered or associated with a special trigger forcing ASIC to respond with empty data frame
  - → This frame indicates to RDO that no more data is expected from the ASIC on the FEB for a given revolution



- Case 2: companion ASIC / FPGA doing FEB fragment building
  - → Similar protocol as for RDO-DAM communication
    - Either: RevTick acknowledgment frame is sent after all data belonging to a revolution is sent out from FEB
    - Or: last FEB data frame has a special end-of-revolution marker
    - Or both above for robustness against packet loss and improved error detection

RDO: try to estimate memory needs for MPGD fragment building

- Assuming an RDO for a 512-channel FEB
  → 8 64-channel ASICs (*e.g.* Salsa)
- Case 1: Sampling readout
  → 500 ns readout window when signal is above threshold
    - 12-bits per sample, 50 MSPS, 25 samples

Channel rate kHz	64-chanel ASIC Mbit/s	512-chanel RDO Gbit/s	RDO revolution (12 µs) data kbyte
2 (physics)	53	0.42	0.63
10 (safety)	**264**	**2.1**	**3.15**
50 (Clas12)	1 318	10.5	15.75

Based on MpgdFeOpt

- Case 2: Peak-finding readout
  → 12-bit time of arrival, 8-bit ToT and 12-bit amplitude for signal is above threshold

Channel rate kHz	64-chanel ASIC Mbit/s	512-chanel RDO Gbit/s	RDO revolution (12 µs) data kbyte
2 (physics)	8.2	0.066	0.1
10 (safety)	**41**	**0.33**	**0.5**
50 (Clas12)	205	1.64	2.5

Based on MpgdData

- Not much, even if one needs to account for some more required for calibration data

DAM data: try to estimate memory needs for MPGD fragment building

- Assuming ~100k MPGD channels
  - → 192 512-channel RDOs
  - → 8 DAMs aggregating 24 RDOs each

Added after

Streaming model

- Case 1: Sampling readout

		12 us revolution data			1 ms time frame data		
Channel rate kHz	**512-ch. RDO Gbit/s**	**RDO data kbyte**	**DAM data kbyte**	**PC data Mbyte**	**RDO data kbyte**	**DAM data Mbyte**	**PC data Mbyte**
10 (safety)	**2.1**	**3.15**	**76**	**0.6**	**262.5**	**6.3**	**50**
50 (Clas12)	10.5	15.75	378	3	1 312.5	31.5	250

- Case 2: Peak-finding readout

		12 us revolution data			1 ms time frame data		
Channel rate kHz	**512-ch. RDO Gbit/s**	**RDO data kbyte**	**DAM data kbyte**	**PC data Mbyte**	**RDO data kbyte**	**DAM data Mbyte**	**PC data Mbyte**
10 (safety)	**0.33**	**0.5**	**12**	0.1	**41.25**	**1**	**8**
50 (Clas12)	1.64	2.5	60	0.5	205	5	40

- Memory resources should be enough to perform data aggregation in larger time frames than a revolution
  - → DAM might be more appropriate though – no radiation
- Out of scope: attention should be paid for pathological many-to-one traffic pattern from DAMs to Computing PCs
  - → Potentially creating congestion in the data collection network
  - → In particular for non-ZS calibration data

Summary

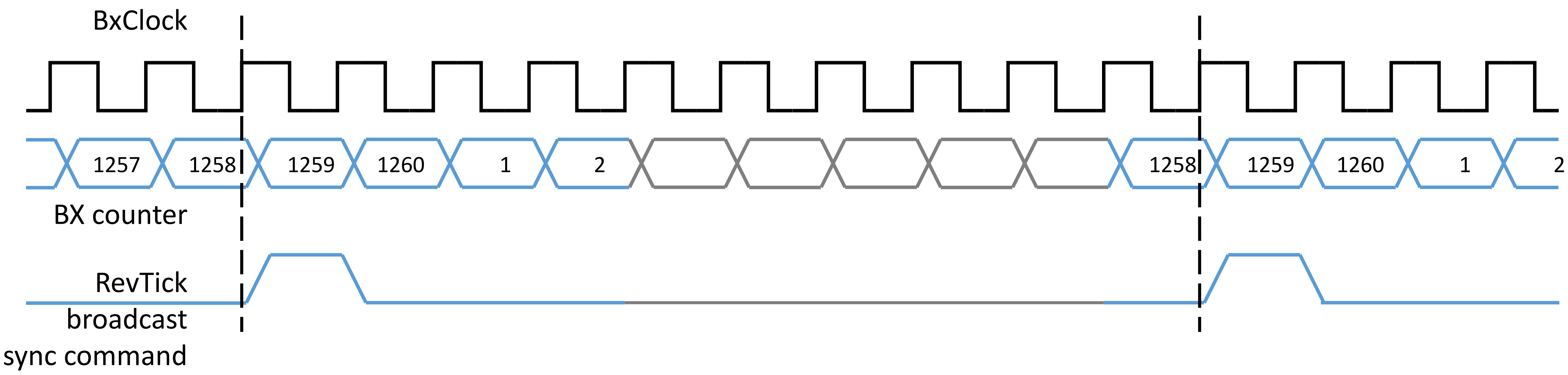
- RevTick synchronous command can be considered as heartbeat
  - → Response may not necessarily be synchronous
  - → Can be issued after the revolution data has been sent out
  - → Simplifies data collection in DAM and RDO and possibly in FEB
  - → Simplifies error detection in DAM, RDO and possibly in FEB and ASICs

- Can RevTick be considered as a 78.195 kHz constant rate trigger?
  - → Perform RDO fragment building and DAM data collection based on well-known event building protocols
    - Should DAM perform fine-time association of data received from RDOs prior to convey them to higher levels?
  - → Would this allow seamless integration of streaming and triggering schemes?
    - *Example*
      - *DAM buffers revolution data*
      - *Upon reception of the trigger, extracts relevant data*
      - *Forms the fragment and send it to upper layers*
  - → May this approach be used to uniform data collection from different sub-detectors till the DAM level?
    - What would be RDO and DAM memory requirements in sub-detectors?

Backup

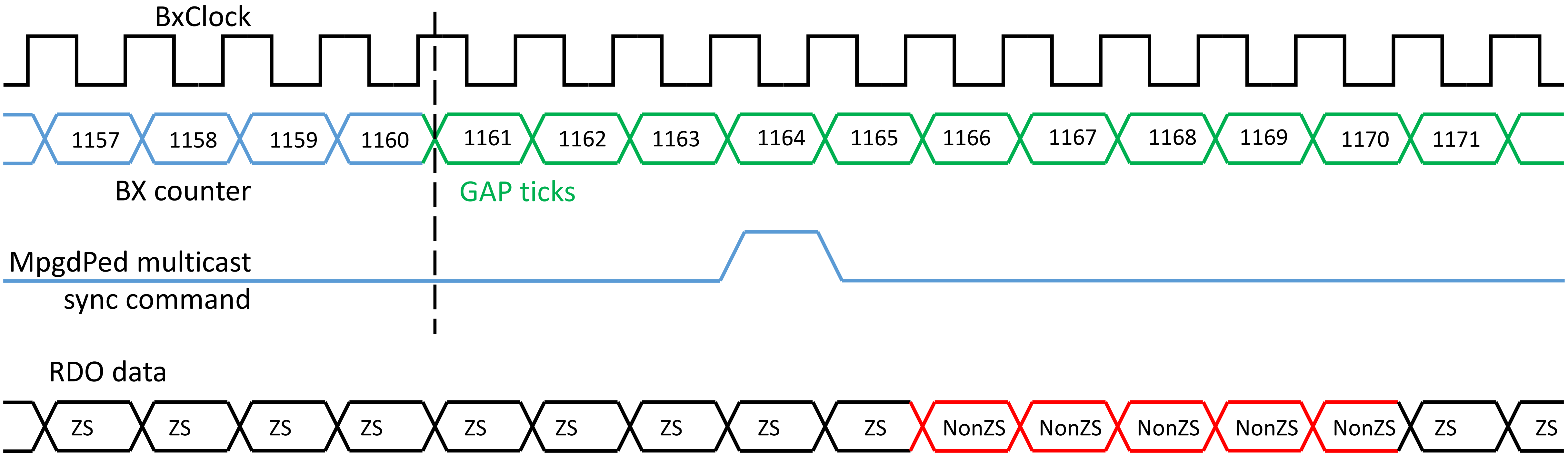
Common broadcast sync command example: "RevTick"

- At each revolution DAQ generates RevTick broadcast sync command
  - → Exactly every 1260 bunch crossings
- RDO maintains N-bit timestamp counter incremented with system clock
  - → 11 bits are enough to code 1260 bunches in revolution
    - Counter can be longer if it is a part of timestamp
- RDO checks synchronization at every RevTick reception
  - → There should be exactly 1260 bunch crossings between RevTiks
  - → Alert (request re-synchronization) in case of mismatch



Specific multicast sync command example: “MpgdPed”

- On-demand, an Mpgd partition sends pedestal reading “MpgdPed” multicast sync command
- “MpgdPed” is sent within the 1 µs GAP period
- Upon its reception concerned RDOs send non-ZS data during programmable time
  - → For More details: https://indico.bnl.gov/event/16040/contributions/64090/attachments/41290/69185/220520_MpgdTrack_CalibRates_IM.pdf

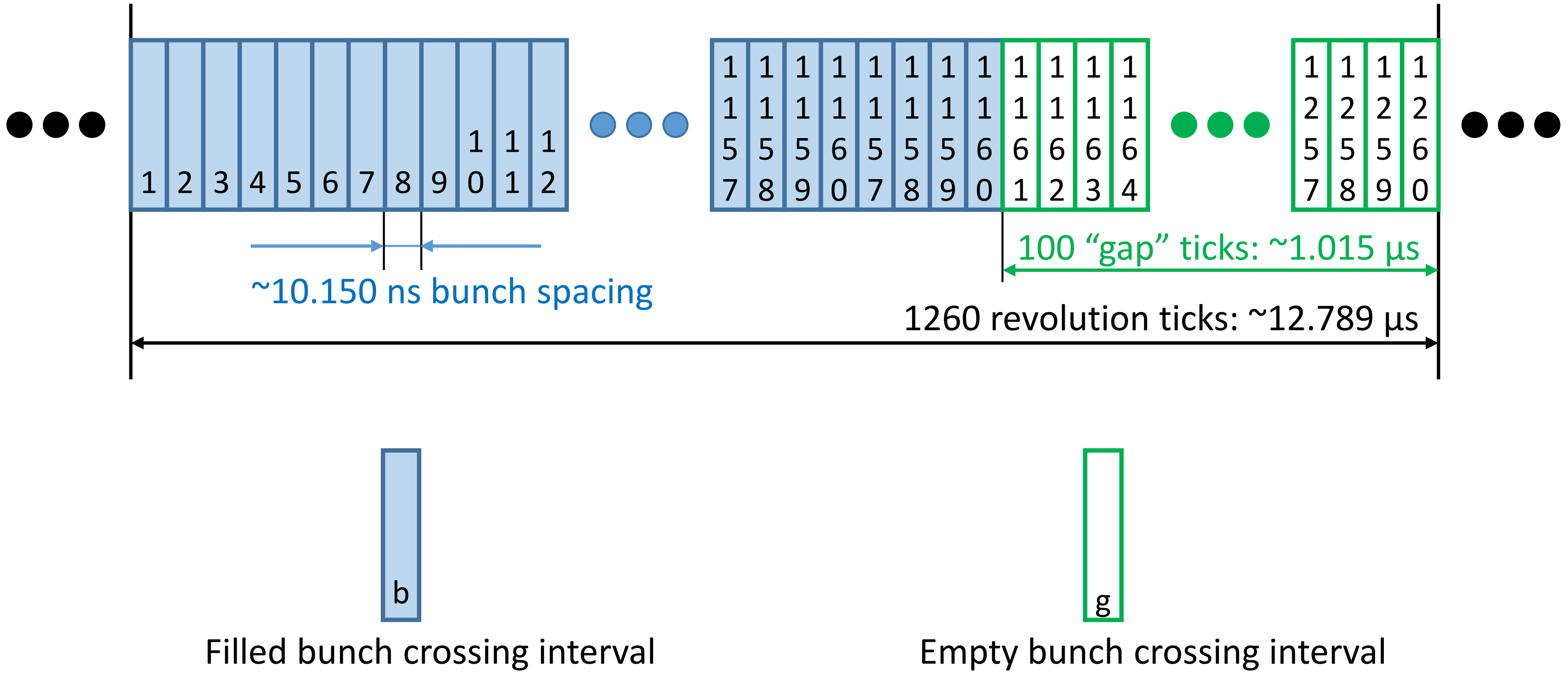


Common/specific sync command example: “Resync”

- How to recover if one of the RDOs loses synchronization?
- Should there be means to resynchronize a partition or entire experiment?
  - → Power cycling and reconfiguration might be a long process
    - ■ Should be the last resort
- A well established sequence of synchronous commands may help
  - → *e.g.* Pause, Resync, Restart
    - ■ Pause may imply finishing sending of all pipelined data and asserting busy
    - ■ Resync may imply resetting some part of the hardware without touching configuration and removing busy
      - One may avoid resetting serial links with ASICs if they are properly aligned
    - ■ Restart may indicate that RDO must resume sending data immediately after next RevTick broadcast command
      - It is sent once all partitions removed busy and are in ready state
  - → One may envisage commands initiating serial data link alignment sequence between the RDOs and ASICs
    - ■ RDO receives the fast command and conveys it to the ASICs that begin to send a predefined pattern
  - → Slow control (I2C) channel may be used too for some of the above but probably will be a slow process
- Synchronization loss may occur at various levels
  - → RDO-DAM link, FE ASIC high speed serial link
  - → Fast resynchronization mechanisms may be needed at system, sub-system and FE level
    - ■ Collect and document them

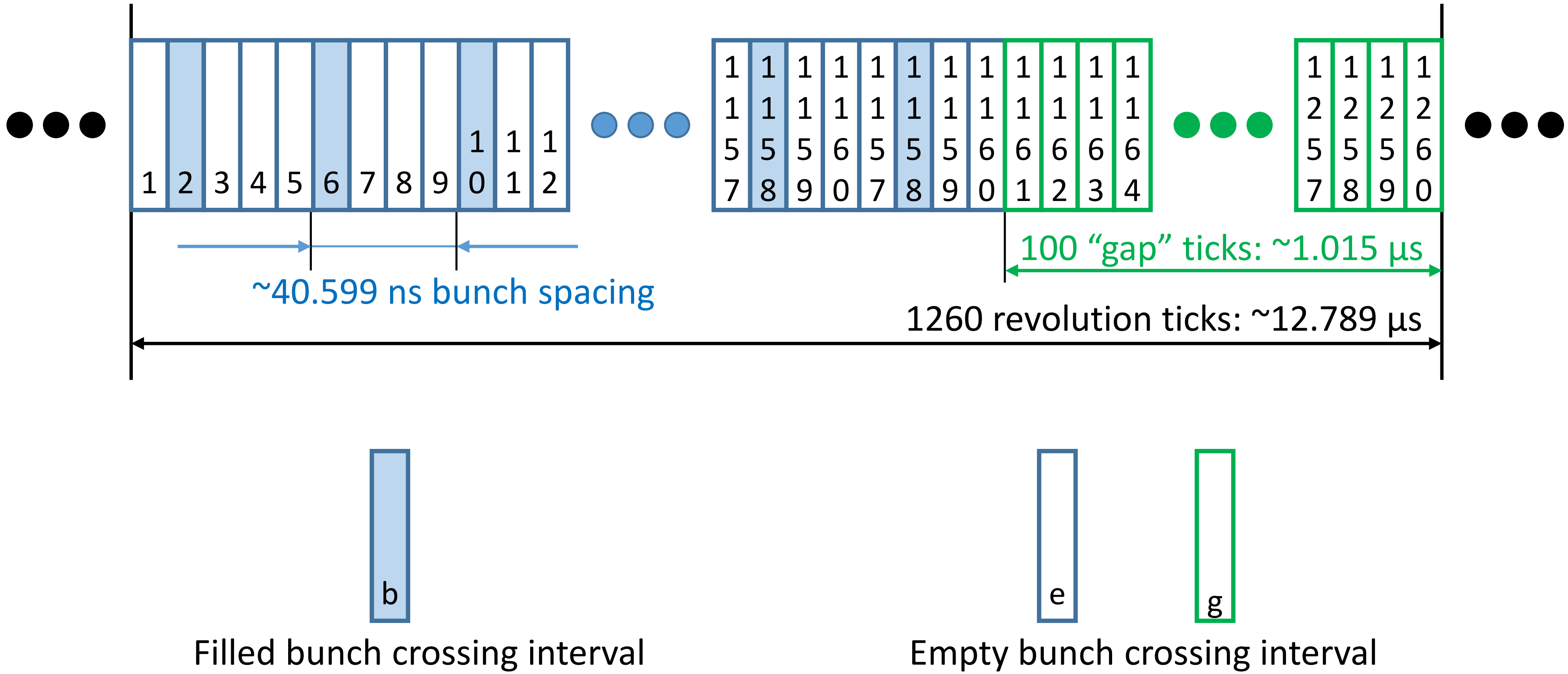
Operation with 1160 bunches

- All bunch intervals contain particles and may produce interactions



Operation with 290 bunches

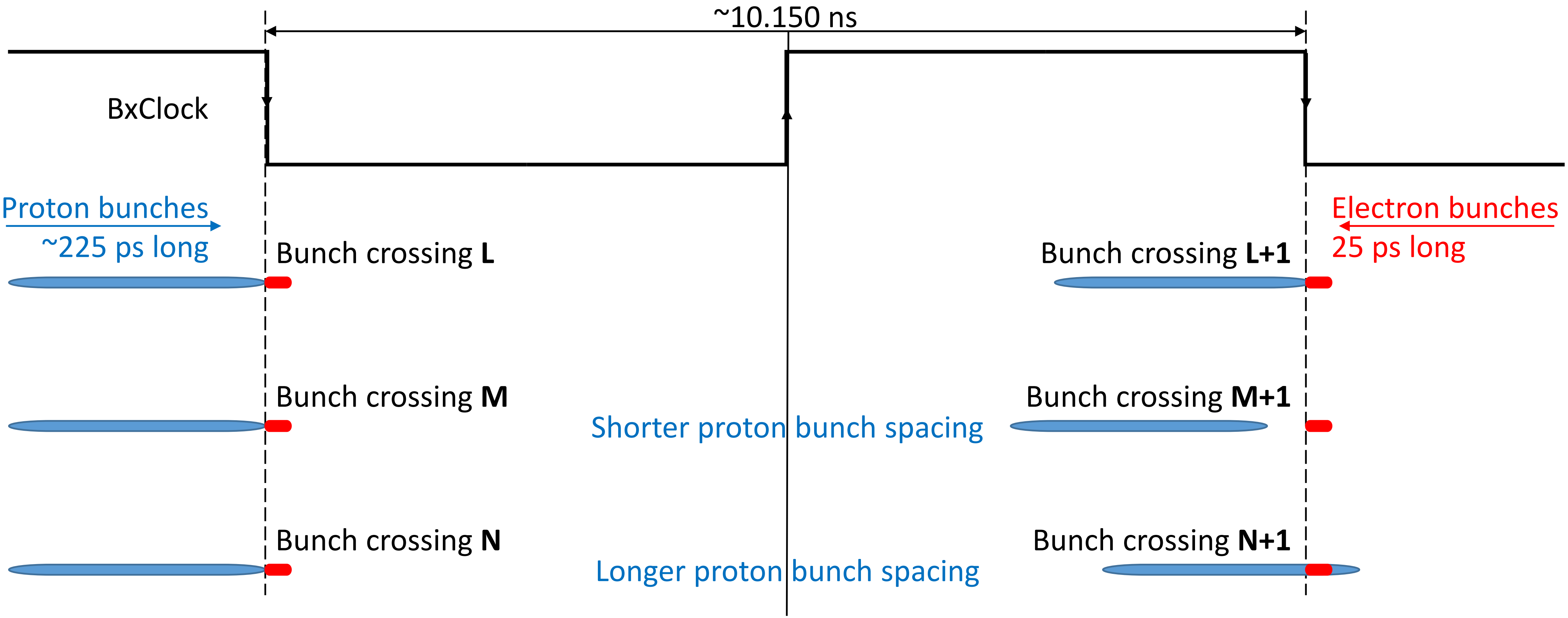
- Only every 4th bunch interval contains particles and may produce interactions



Are the position of the filled bunch intervals fixed (for good)?
Where is the very first filled bunch interval within the revolution?

Bunch crossing clock and particle-crossing

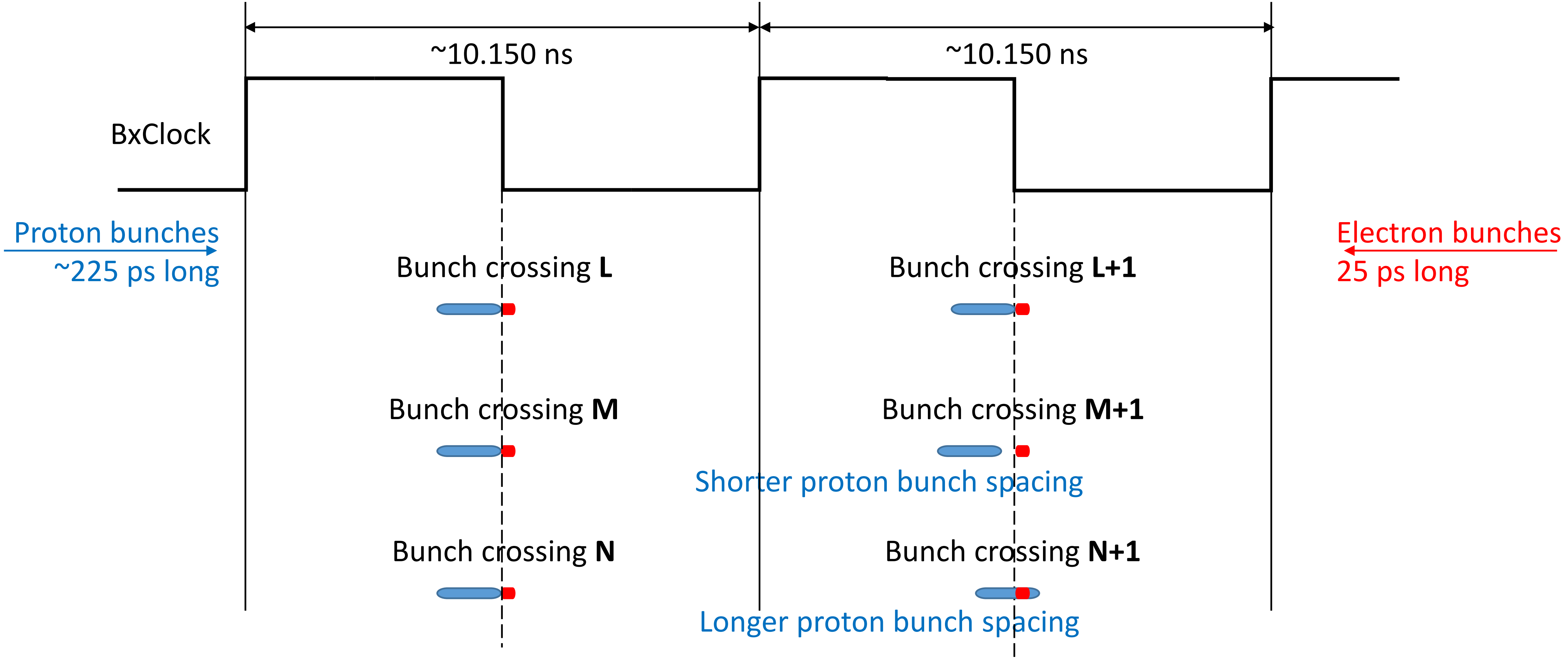
- Possible variation of actual bunch crossing times versus ideal clock intervals?
  - → Mostly due to variability in proton beam bunch spacing
  - → Electron beam timing expected to be more stable
- Can RF system of electron beam be used to derive stable **bunch crossing** clock?
  - → Distributed **system clock** can be a multiple of the bunch crossing clock



Variations within a revolution? Variations from revolution to revolution?

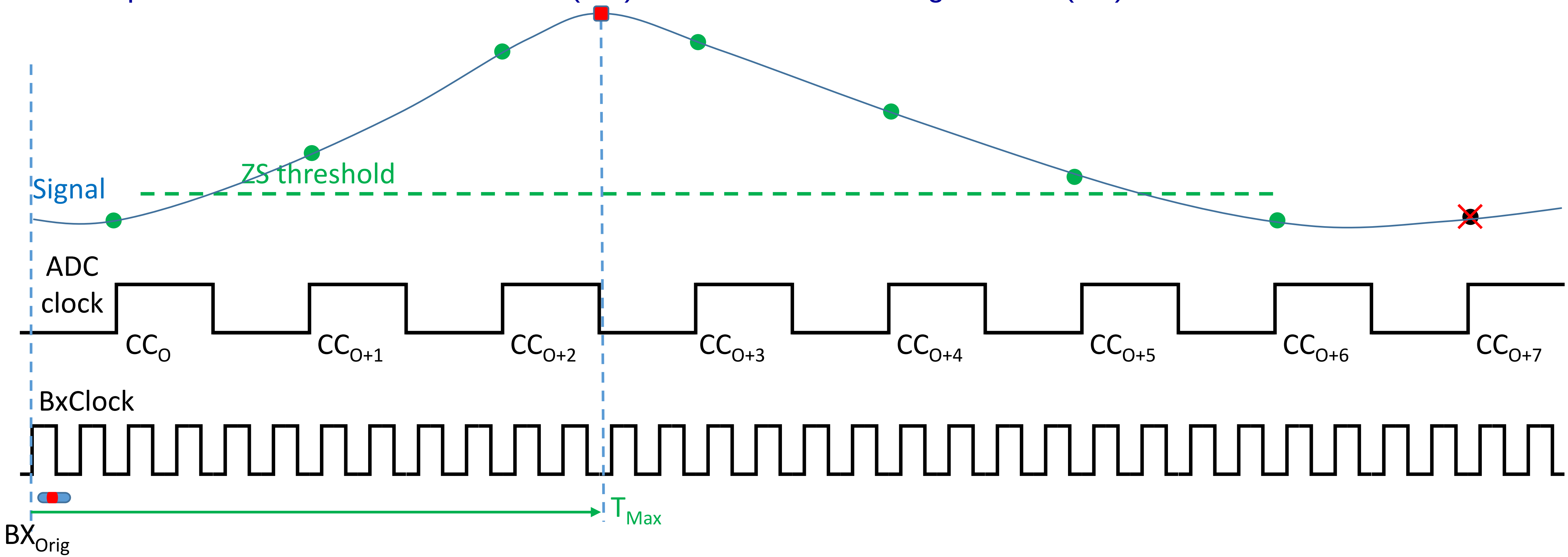
Bunch crossing clock and particle-crossing

- Bunch crossing clock is phased enough to avoid any boundary conditions
  - → Collisions contained within the bunch crossing clock period – not on the transitions
  - → Variable bunch spacing does not compromise collision data to bunch crossing association



Bunch crossing clock and timing: signal shape sampling with ADC

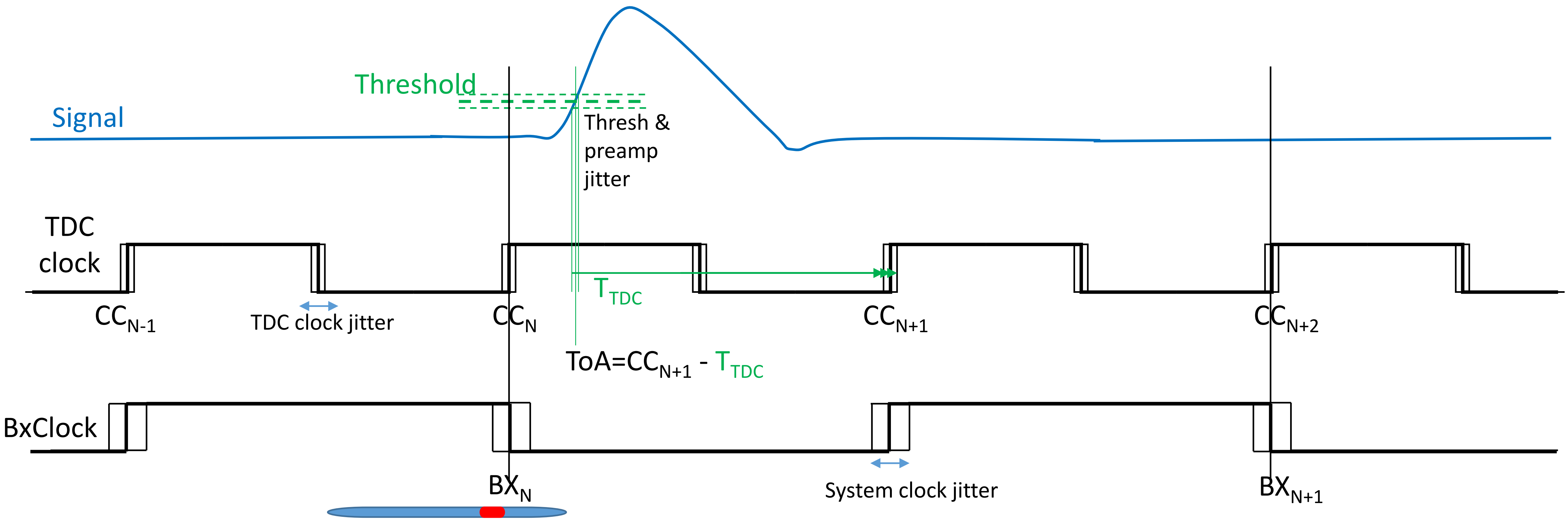
- Derive signal timing from N samples
- Complement with the coarse counter (CC) and/or bunch crossing counter (BX)



- ADC clock is derived from bunch crossing (system) clock and is its (sub)multiple (e.g. ~25 MHz) or (M / N) fractional
  → Known frequency and phase relationship exists between the two clocks
- Signal association with originating bunch crossing possible
  → Max (as on example) or time of arrival (fitting samples on rising edge)

Bunch crossing clock and timing: time of arrival (ToA) with TDC

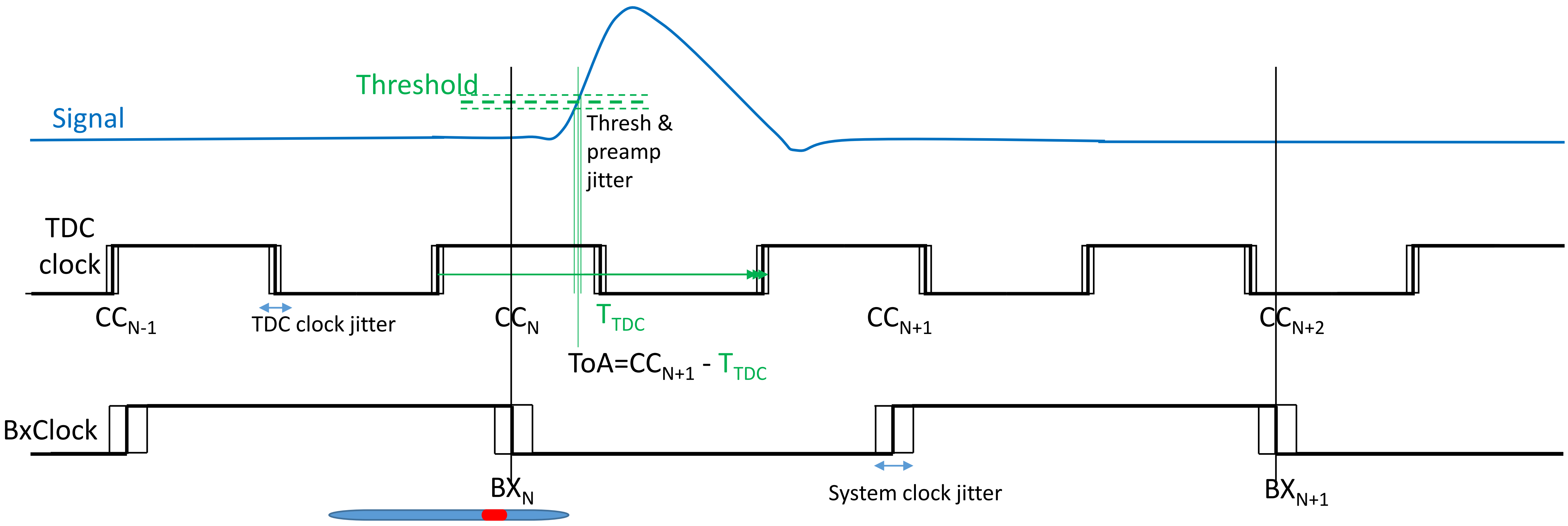
- Measure time interval between threshold crossing and next rising edge of the TDC clock
- Complement with TDC coarse counter (CC)



- TDC clock is derived from bunch crossing (system) clock and is its multiple (e.g. ~200 MHz) or (M / N) fractional
  → Known frequency and phase relationship exists between the two clocks
- Measured ToA has obvious relationship with bunch crossings

Bunch crossing clock and timing: uncorrelated or fractional TDC clock

- TDC case: Time of Arrival (ToA) measurement
  - → Measure time interval between threshold crossing and next rising edge of the TDC clock
  - → Complement with coarse counter (CC)



- TDC clock uncorrelated to system clock
  - → Phase relationship between the two clocks is not deterministic: should be monitored; how?

Is such a case considered?

Same questions are valid for signal shape sampling with ADC