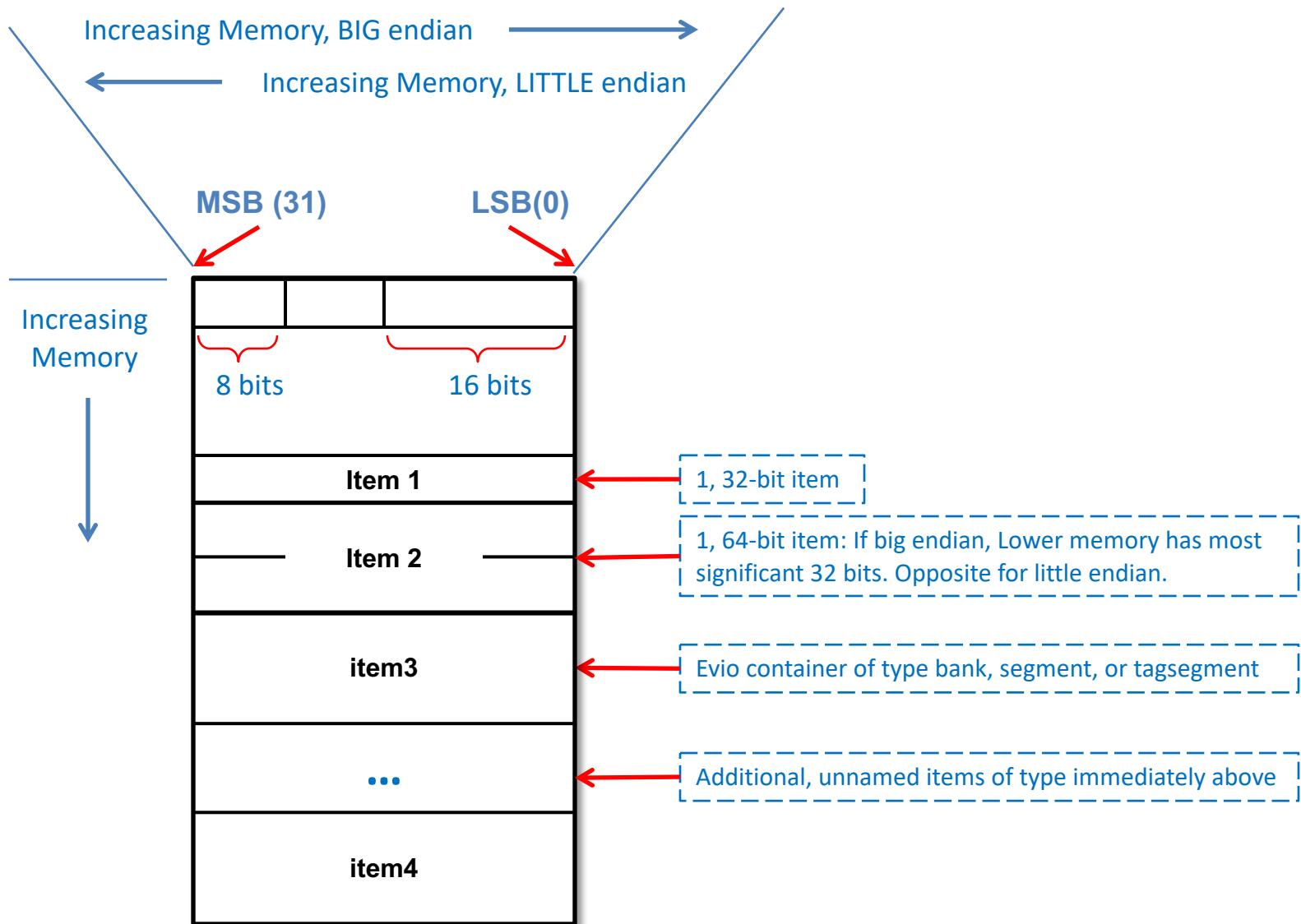


CODA Online Data Formats

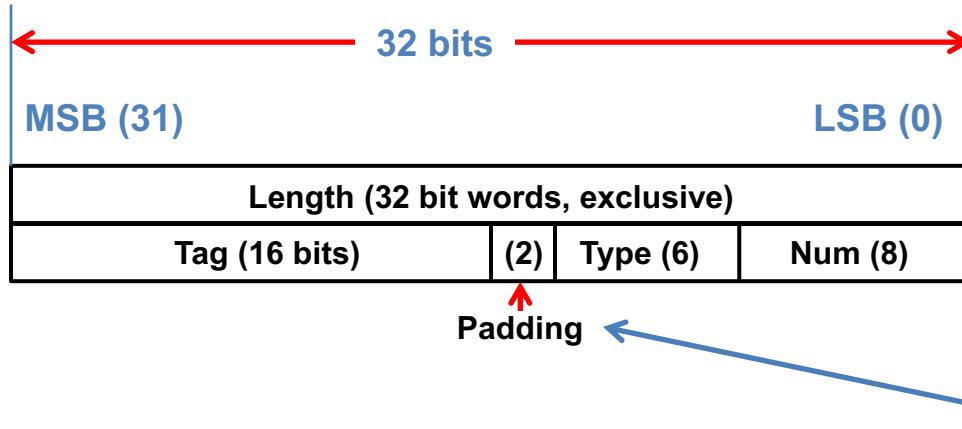
Mar 26, 2025

Key to Reading Data Layouts



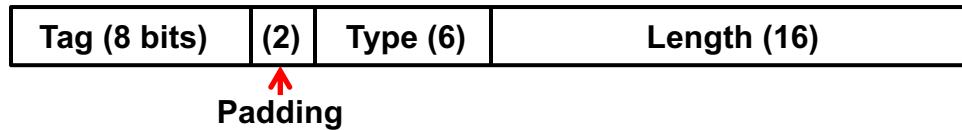
Evio Header Formats

Bank :



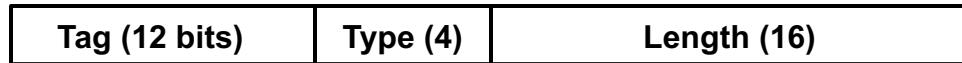
Number of unused bytes at end of following data if not a multiple of 32 bits.
For shorts, it is 0 or 2.
For chars (not strings), it is 0, 1, 2, or 3

Segment :



NOTE: there is no padding stored in tag segments, so only an even number of data elements should be used for shorts and a multiple of 4 for chars/bytes. If you add an odd amount, you'll end up with extra elements at the end.

Tag Segment :

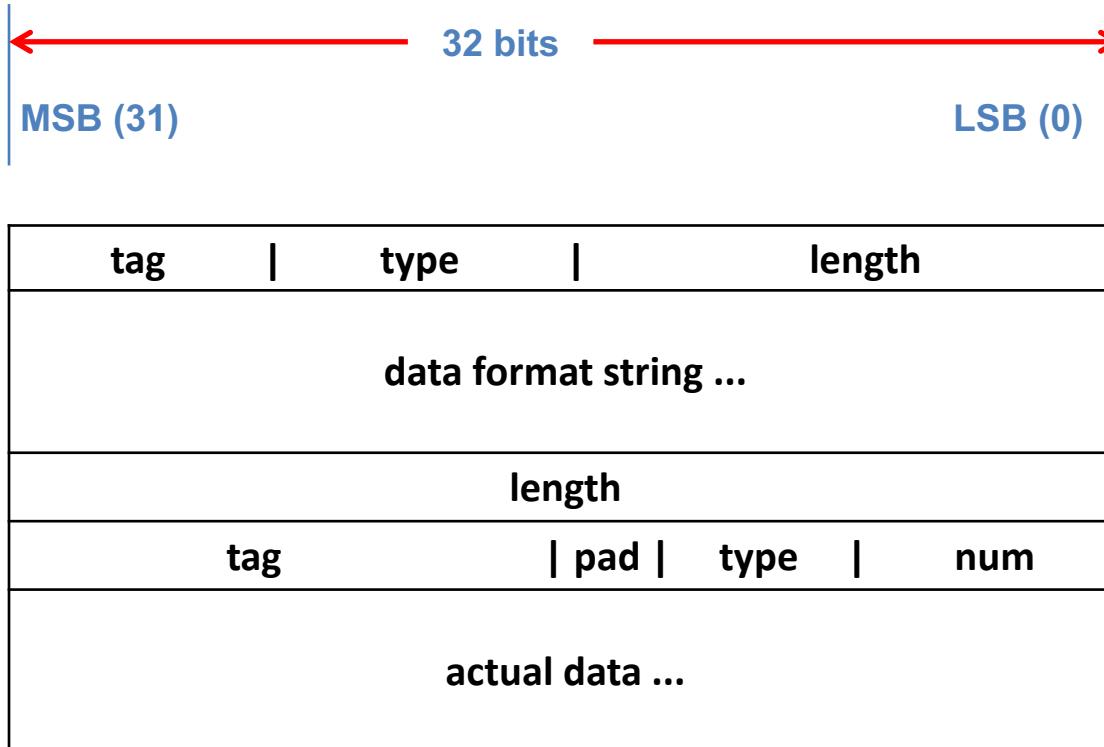


Evio Content Type Codes

Content Type	Primitive Data Type	Content Type	Primitive Data Type
0x0	32 bit unknown (not swapped)	0x21	Hollerit (Composite data internal)
0x1	32 bit unsigned int	0x22	N value (32 bit int, Composite data internal)
0x2	32 bit float	0x23	n value (16 bit int, Composite data internal)
0x3	8 bit char* (string)	0x24	m value (8 bit int, Composite data internal)
0x4	16 bit signed int		
0x5	16 bit unsigned int		
0x6	8 bit signed int		
0x7	8 bit unsigned int		
0x8	64 bit double		
0x9	64 bit signed int		
0xa	64 bit unsigned int		
0xb	32 bit signed int		
0xc	Tag Segment		
0xd	Segment		
0xe	Bank		
0xf	Composite		
0x10	Bank		
0x20	Segment		

Composite Data Type

In Hall B's new data type, the first word comprises a TAGSEGMENT header which is followed by a string describing the data to come. After this TAGSEGMENT containing the data format string, is a BANK containing the actual data.



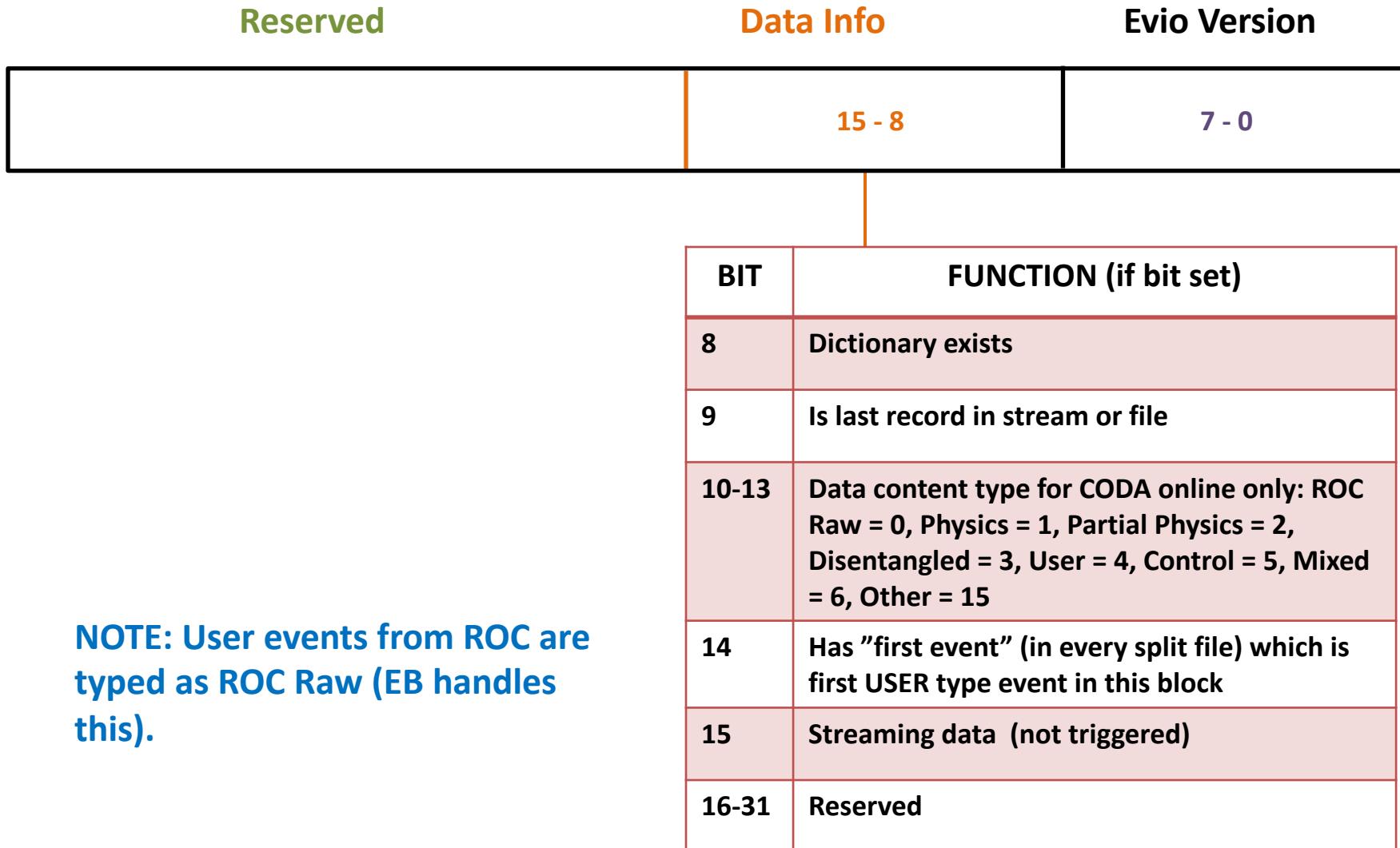
Block Header (evio format versions 1-3)

1	Block Length	Length of block in 32-bit words, inclusive
2	Block Number	Record id starting at 0
3	Header Length	Length of block header in 32-bit words (8)
4	Start	Offset in words to first event header in block relative to start of block
5	End	Number of valid words in block (header + data). Same as block length except for the last block.
6	Version	Evio format version
7	Reserved	Reserved
8	Magic Number	Number for endianness tracking (0xc0da0100)

Block Header (evio format version 4)

1	Block Length		Length of block in 32-bit words, inclusive
2	Block Number		Order of block in network transfer (record id) starting at 1. From ROC: -1 if payload banks not being built.
3	Header Length		Length of block header in 32-bit words (8)
4	Event Count		Number of evio events (payload banks) in block, not including dictionary.
5	Reserved 1		If content is being built (eg ROC Raw type), = source CODA id, else reserved
6	Bit Info	Version	Evio format version in low 8 bits. Bit Info in high 24 bits See next slide.
7	Reserved 2		Reserved
8	Magic Number		Number for endianness tracking (0xc0da0100)

Block Header, Bit Info / Version Word



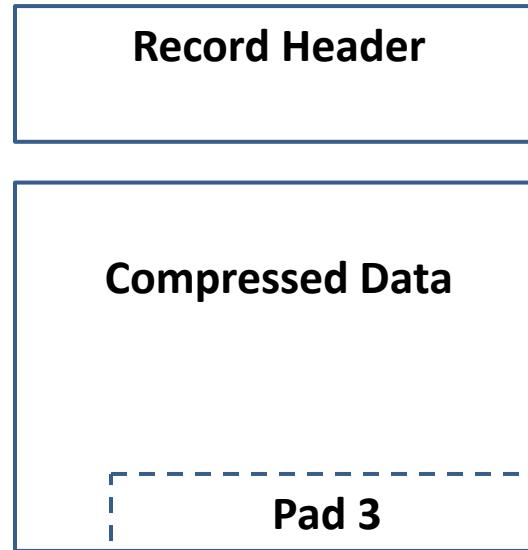
**HIPO/EVIO
FORMAT
VERSION 6**

Record

Uncompressed



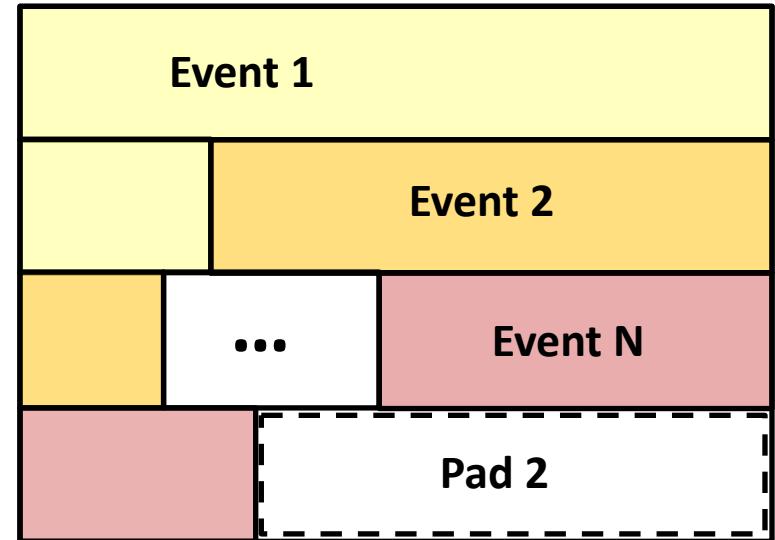
Compressed



Evio Events



HIPO Events



File Trailer

Record Header

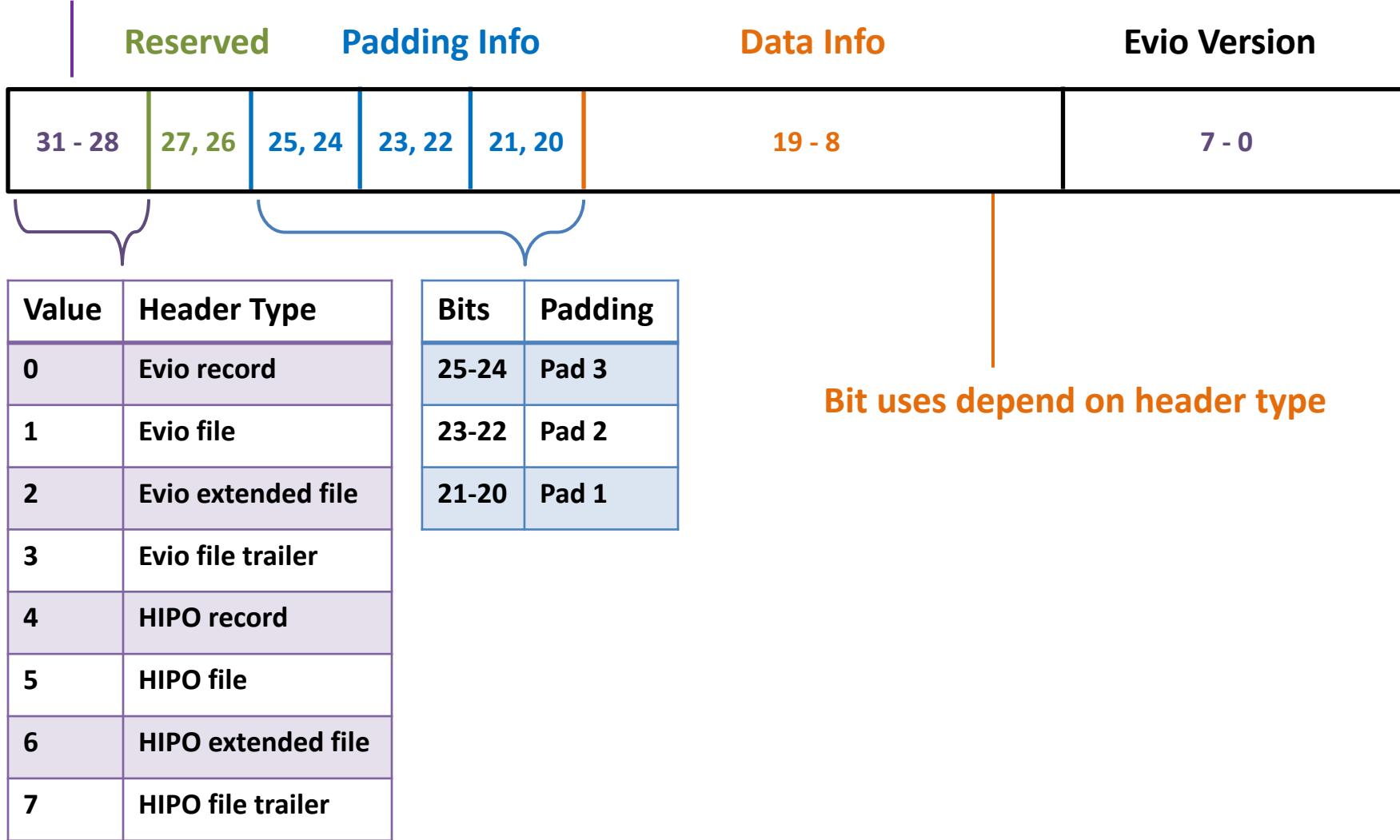
Optional Uncompressed Array:
record (not event !) length in bytes,
followed by its event count
(2 words / record)
(all records except this trailer)

Record Header

1	Record Length		Length of record in 32-bit words, inclusive. If compressed, it's the length of compressed data (including pad 3) + length of this header.
2	Record Number		Record id from 1. From ROC: -1 if payload banks not being built
3	Header Length		Length of this header in 32-bit words (always 14)
4	Event Index Count		Number of events contained (Evio: not including dictionary). Same as index array length in 32-bit words if array exists.
5	Index Array Length		Length of index array in bytes. Each array word is an event length in bytes.
6	Bit Info	Version	Evio format version in low 8 bits. Bit Info in high 24 bits
7	User Header Length		Optional user header length in bytes
8	Magic Number		Number for endianness tracking (0xc0da0100)
9	Uncompressed Data Length		Length of uncompressed index + user header + events in bytes. Does <i>not</i> include padding (pad 2).
10	Type	Compressed Data Length	Compression type in high 4 bits (0=none, 1 = LZ4, 2 = LZ4 Best, 3 = GZIP). Length of compressed data (index + user header + events) in 32-bit words (low 28 bits) which includes padding (pad 3) to 4 byte boundary .
11	User Register 1		User defined long word (64 bits)
12			
13	User Register 2		User defined long word (64 bits)
14			

File/Record Headers, Bit Info / Version Word

General Header Type

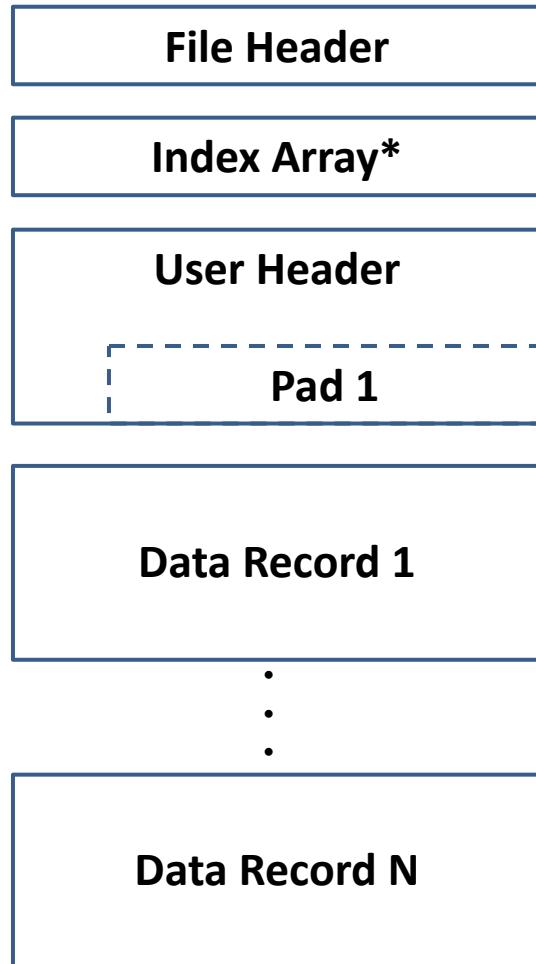


Data Info Bits for Headers

BIT (from 0)	FILE HEADER (if bit on)
8	Dictionary exists
9	Has “first event” (in every split file)
10	File trailer with index array exists
11-19	Reserved

BIT (from 0)	RECORD HEADER (if bit on)
8	Dictionary exists (first record only)
9	Is last record in stream or file
10-13	Data content type for CODA online only: ROC Raw = 0, Physics = 1, Partial Physics = 2, Disentangled = 3, User = 4, Control = 5, Mixed = 6, ROC Raw Streaming = 8, Physics streaming = 9, Other = 15
14	Has “first event” (first record only, in every split file)
15-19	Reserved

File



* Same format as file trailer index:
1 word (4 byte int) of byte length,
followed by 1 word of event count,
for each record (not event!)

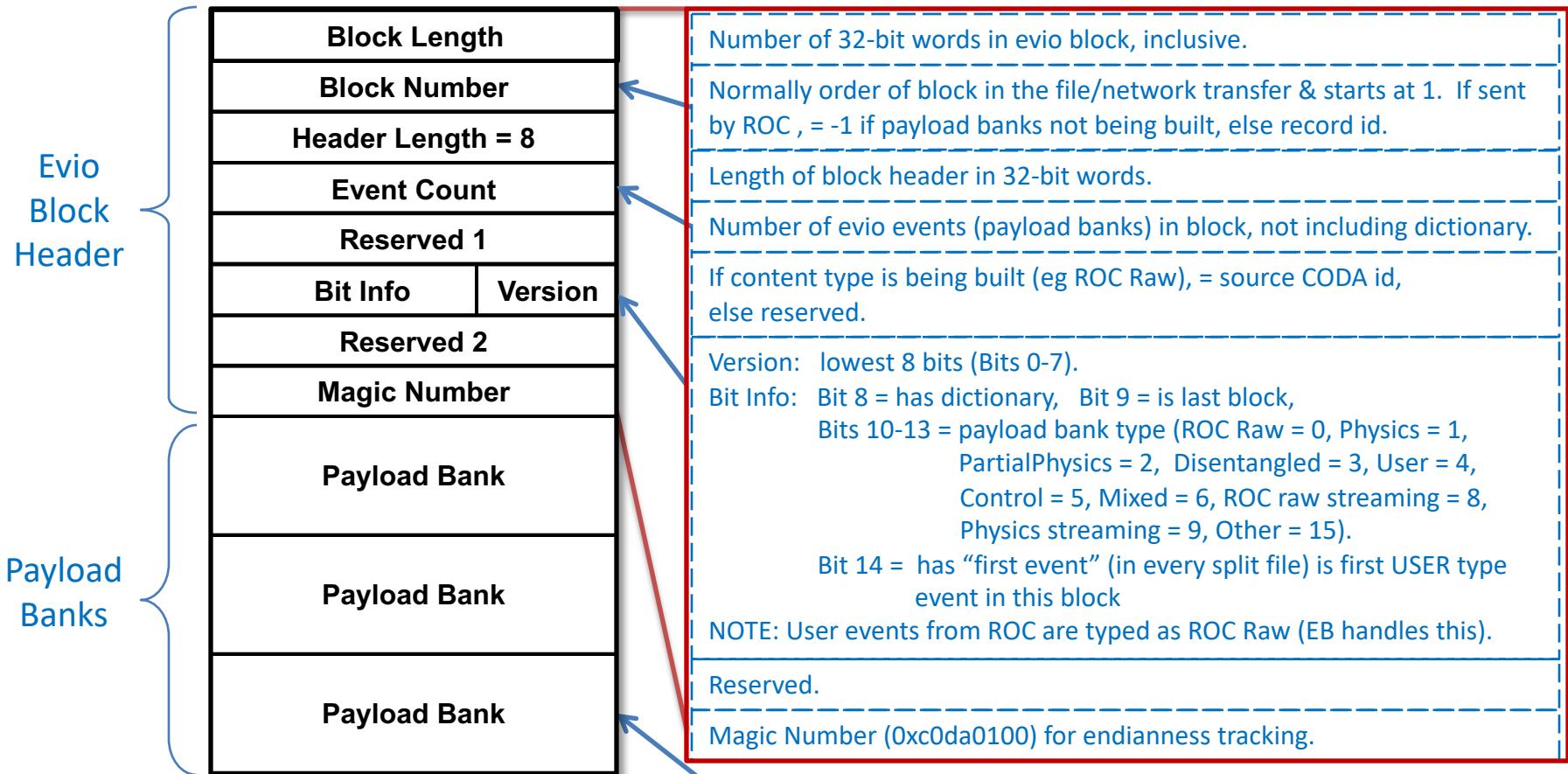
File Header

1	ID		Identification word. For Evio = 0x4556494F (EVIO in ascii). For HIPO = 0x43455248 (CERH in ascii).
2	File Number		If file being split, the split number (starting at 1)
3	Header Length		Length of this header in 32-bit words (always 14)
4	Record Count		Number of records contained. Same as index array length in 32-bit words if array exists.
5	Index Array Length		Length of index array in bytes
6	Bit Info	Version	Evio format version in low 8 bits. Bit Info in high 24 bits
7	User Header Length		Optional user header length in bytes
8	Magic Number		Number for endianness tracking (0xc0da0100)
9	User Register		64 bit register available for user
10			
11	Trailer Position		Number of bytes from beginning of file to beginning of trailer (ending general record header). Value of 0 means either no trailer exists or its position is unavailable
12			
13	User Integer 1		Integer available for user
14	User Integer 2		Integer available for user

EXTENDED File Header (Differences)

3	Header Length	Length of this header in 32-bit words GREATER THAN 14
15 +	User Integers 3+	Additional integers available for user beyond the regular general file header.

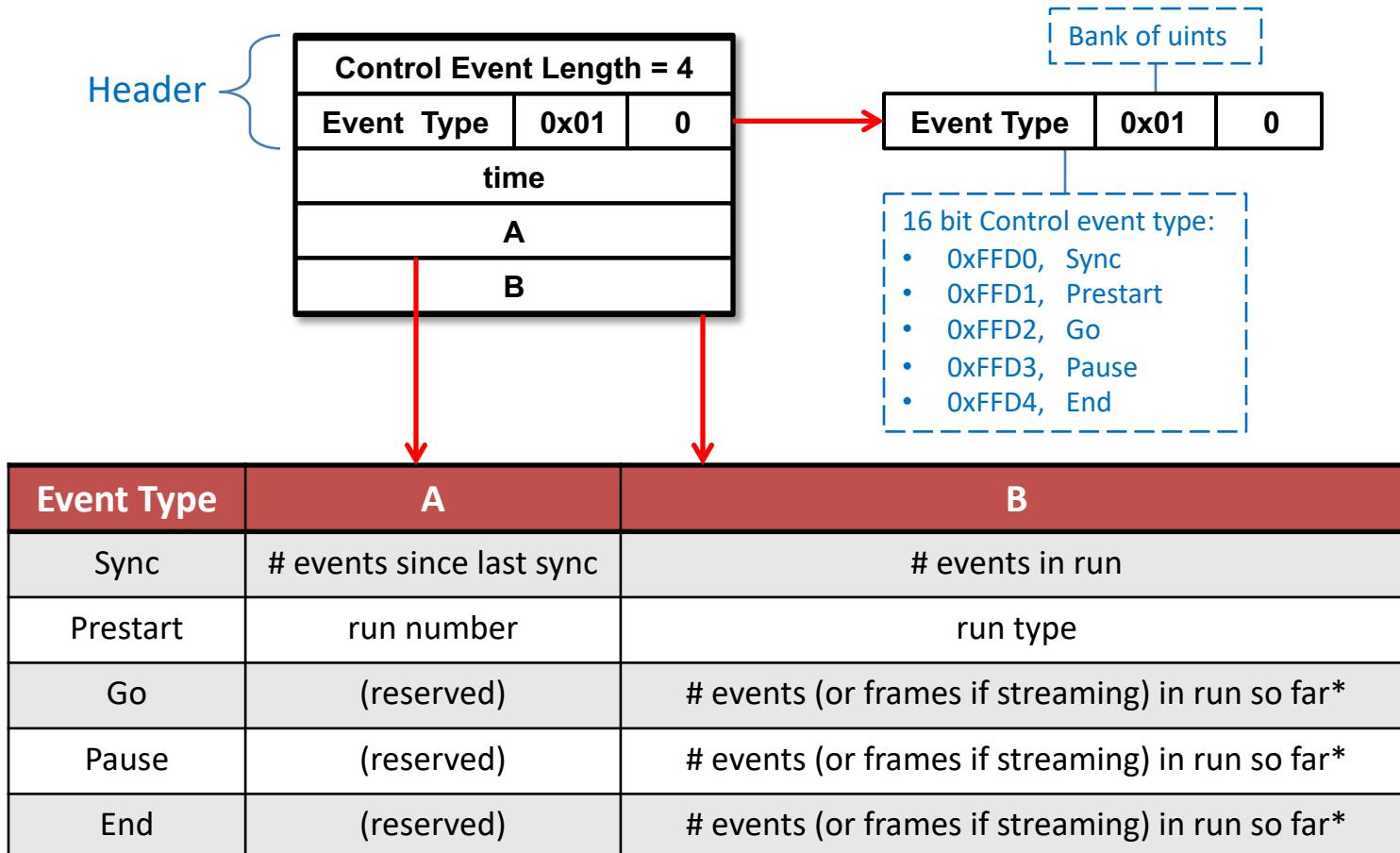
Network Transfer (Evio 4 Output) Format



Format used when sending all types of online CODA data over the network. They are in standard evio buffer/file output format with block headers.

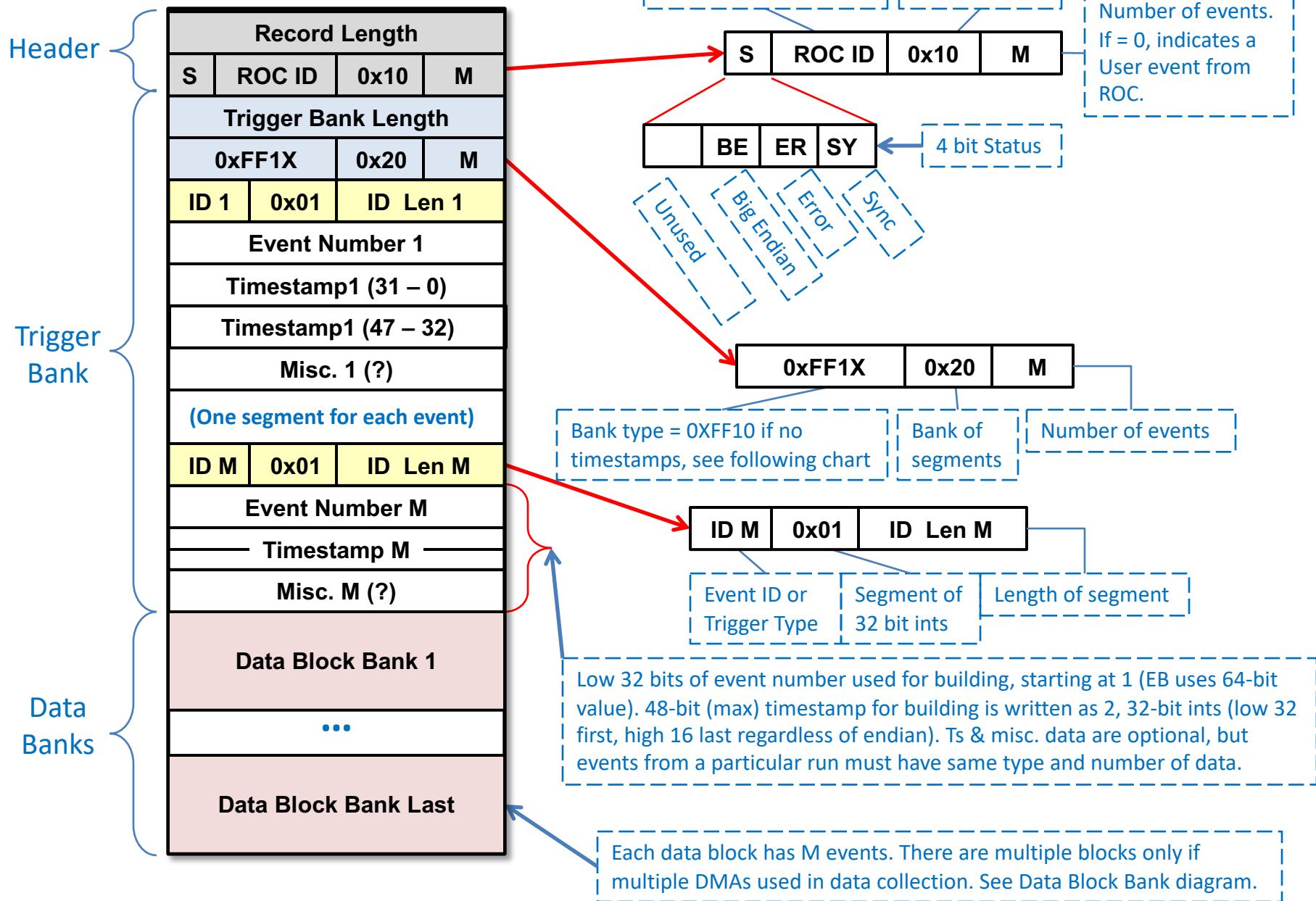
Each payload bank can be a Physics Event, ROC Raw Record, Control Event, or User event. Note: there may be a block header between any 2 payload banks.

Control Event

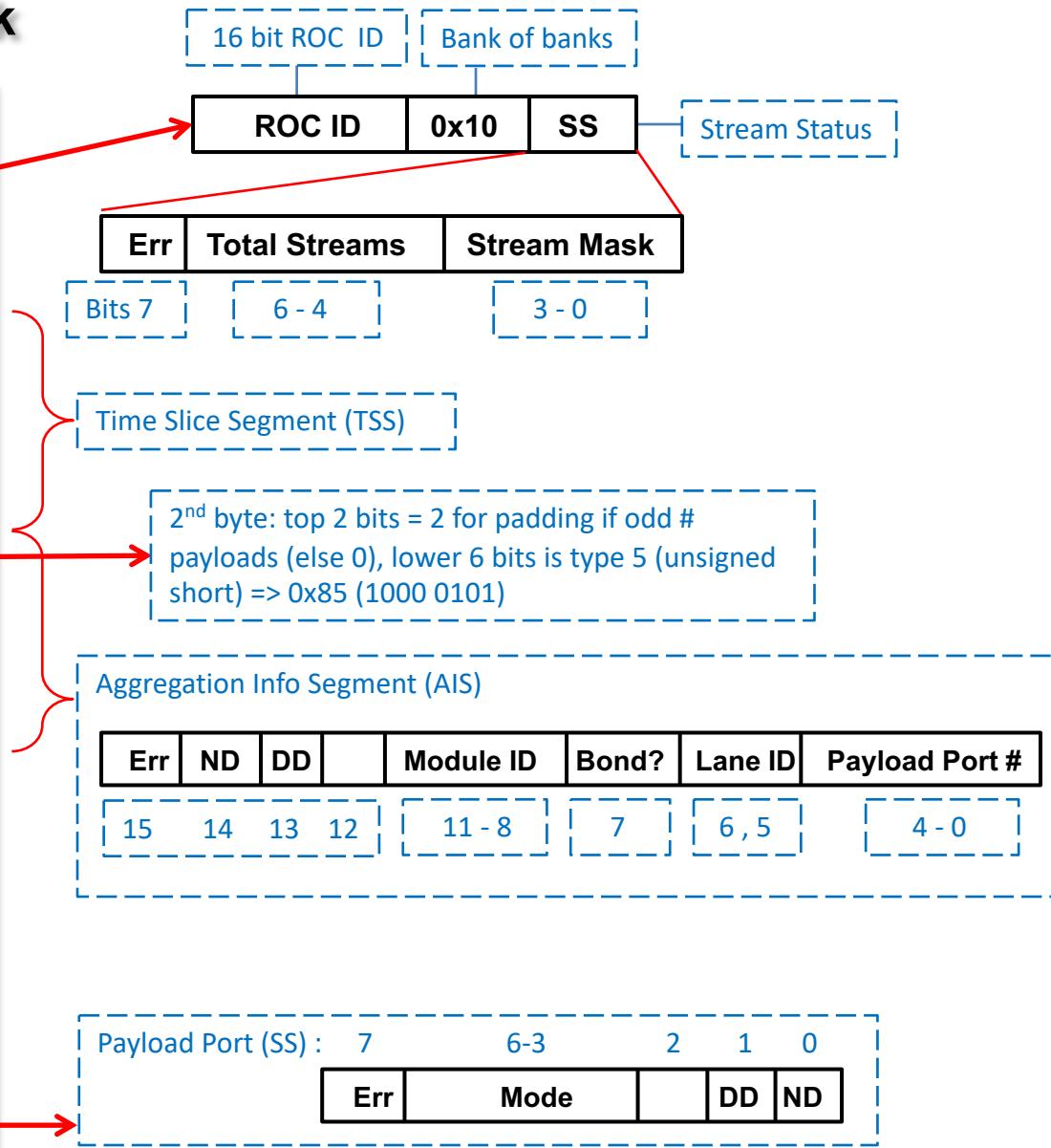


*Control events included in event count but not in frame count

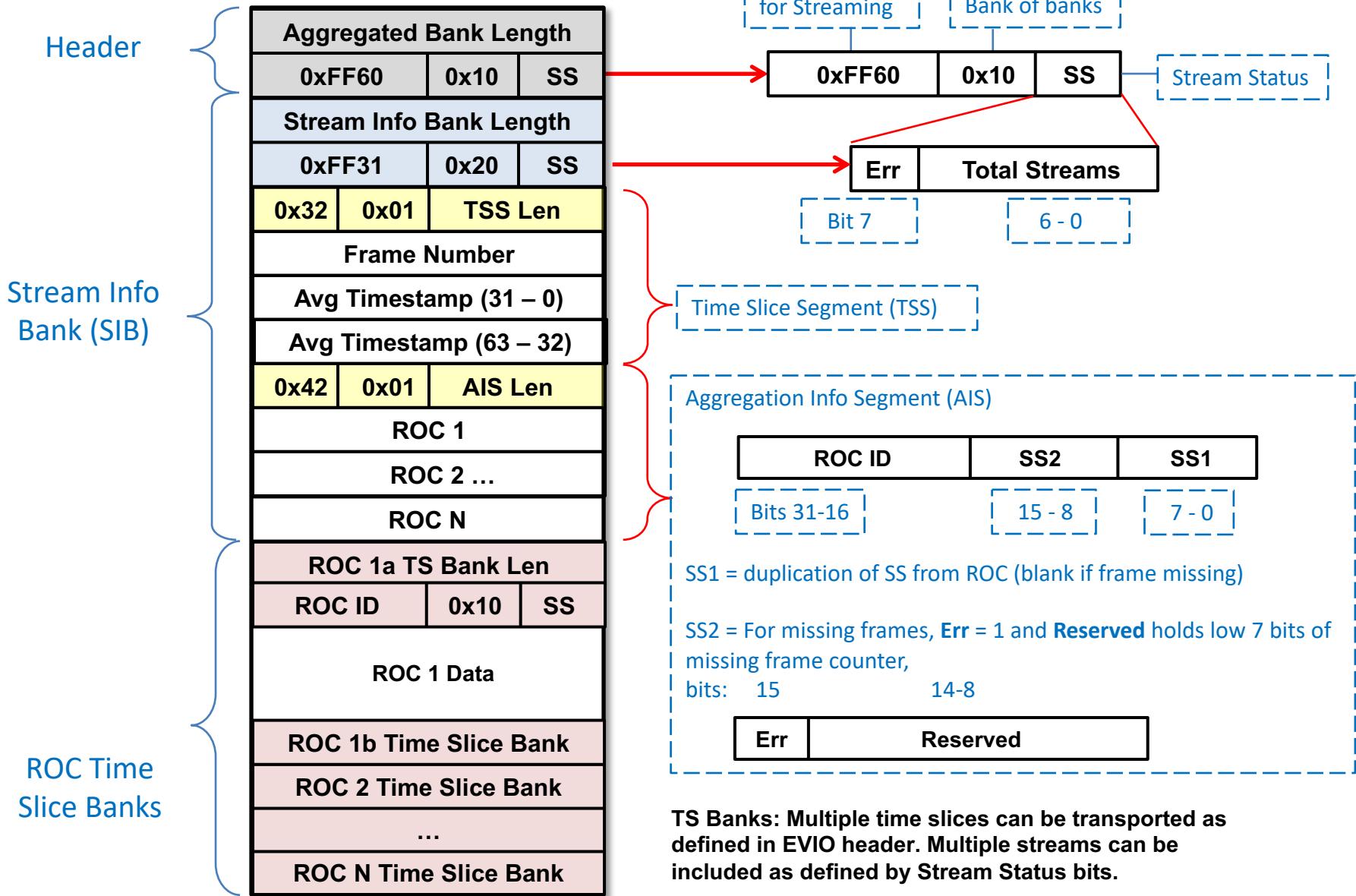
ROC Raw Data Record



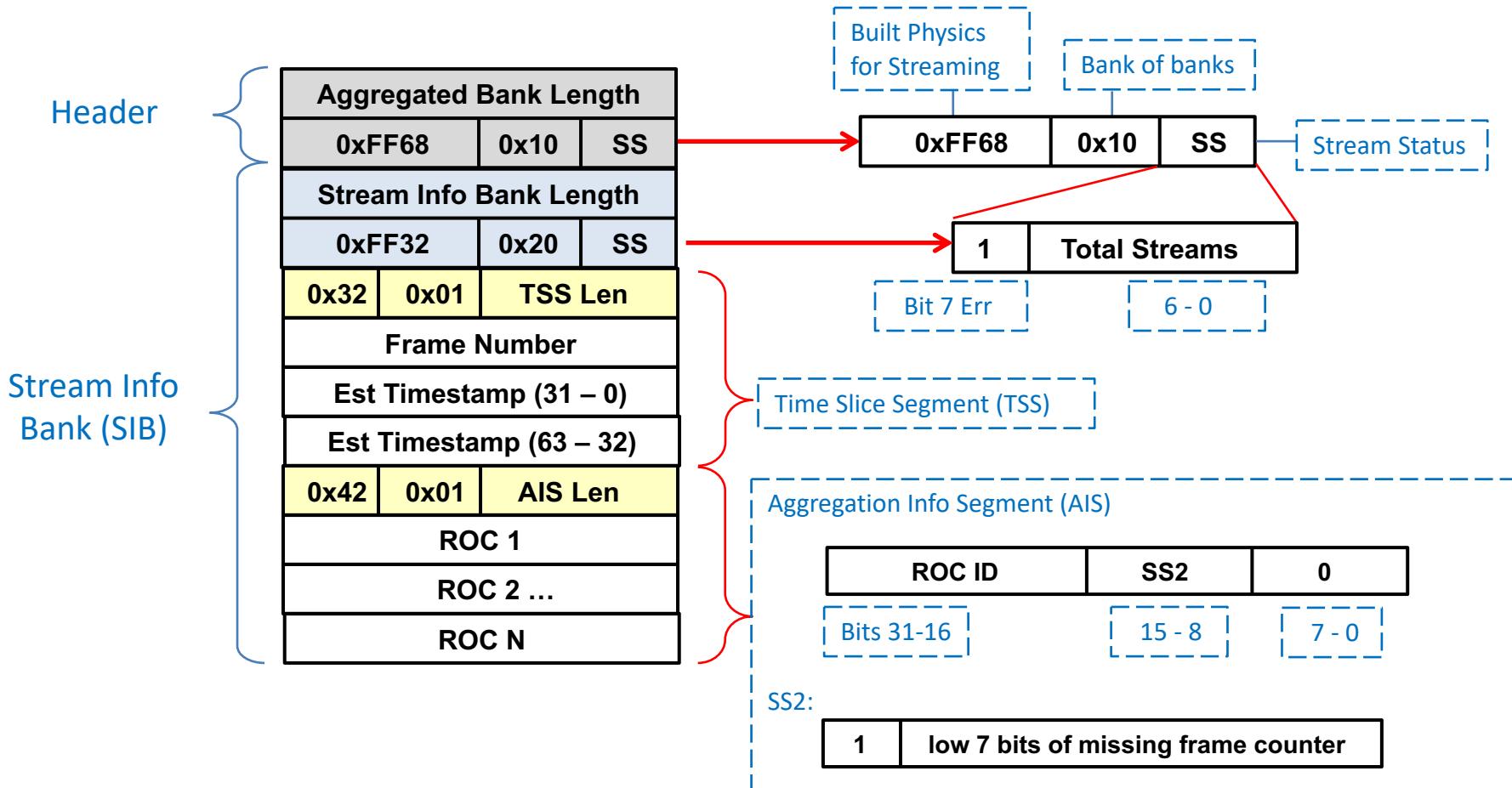
ROC Time Slice Bank



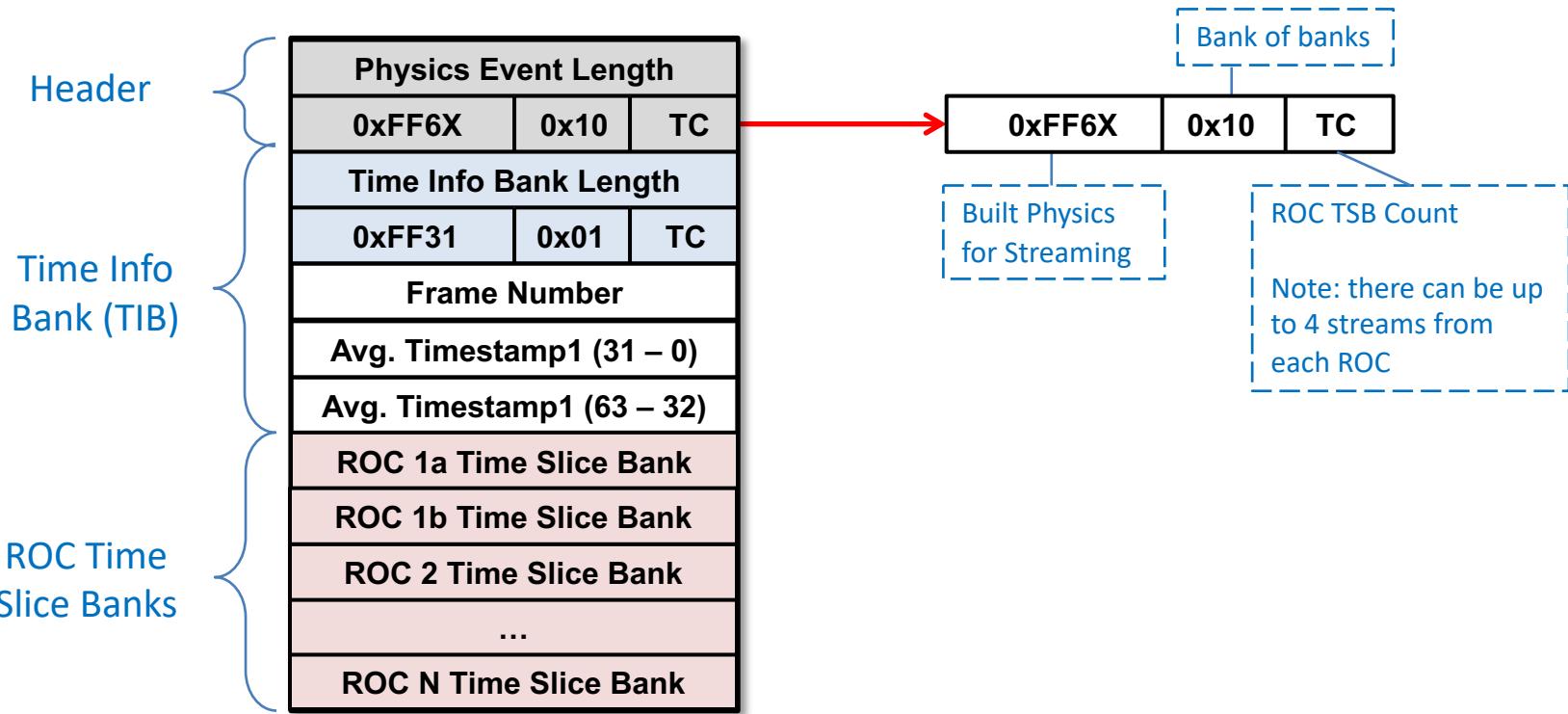
Streaming Physics Event



Streaming Physics Event Empty Frame



Streaming Physics Event (OLD)



TRIGGER BANK TAGS

Tag Value	Purpose
0xFF10	Raw trigger, no timestamps
0xFF11	Raw trigger, w/ timestamps
0xFF20	Built trigger, no timestamps, no run # & run type, includes run specific data
0xFF21	Built trigger, w/ timestamps, but no run # & run type, includes run specific data
0xFF22	Built trigger w/ run # & run type, but no timestamps, includes run specific data
0xFF23	Built trigger with timestamps and run # & run type, includes run specific data
0xFF24	Built trigger, no timestamps, no run # & run type, no run specific data
0xFF25	Built trigger, w/ timestamps, but no run # & run type, no run specific data
0xFF26	Built trigger w/ run # & run type, but no timestamps, no run specific data
0xFF27	Built trigger with timestamps and run # & run type, no run specific data
0xFF4F	Built trigger using at least one ROC with bad or nonexistent trigger bank

PHYSICS EVENT TAGS

Tag Value	Purpose
0xFF50	Event built by primary event builder
0xFF58	Event built by primary event builder with sync bit set
0xFF70	Event built by secondary event builder
0xFF78	Event built by secondary event builder with sync bit set

STREAMING TAGS

Tag Value	Purpose
0xFF30	Stream Info Bank (SIB) from ROC containing frame # followed by timestamp
0xFF31	Stream Info Bank (SIB) from Aggregator containing frame # followed by timestamp
0xFF32	Stream Info Bank (SIB) from Aggregator containing frame # followed by timestamp with non-fatal error
0x31	Time Slice Segment (TSS) from ROC containing frame # followed by timestamp
0x41	Aggregation Info Segment (AIS) from ROC
0x32	Time Slice Segment (TSS) from Aggregator containing frame # followed by timestamp
0x42	Aggregation Info Segment (AIS) from Aggregator
0xFF60	Built from ROC Raw Data streamed from ROC / VTP
0xFF61	Built from ROC Raw Data streamed from ROC / VTP with non-fatal error
0xFF62	Possibly: event built by DC event builder in streaming mode
0xFF64	Possibly: event built by secondary event builder in streaming mode
0xFF66	Possibly: event built by primary event builder in streaming mode
0xFF68	Missing Frame (no ROC data available, estimate for timestamp)

CONTROL EVENT TAGS

Tag Value	Control Event
0xFFD0	Sync
0xFFD1	Prestart
0xFFD2	Go
0xFFD3	Pause
0xFFD4	End

CODA RESERVED BANK TAGS

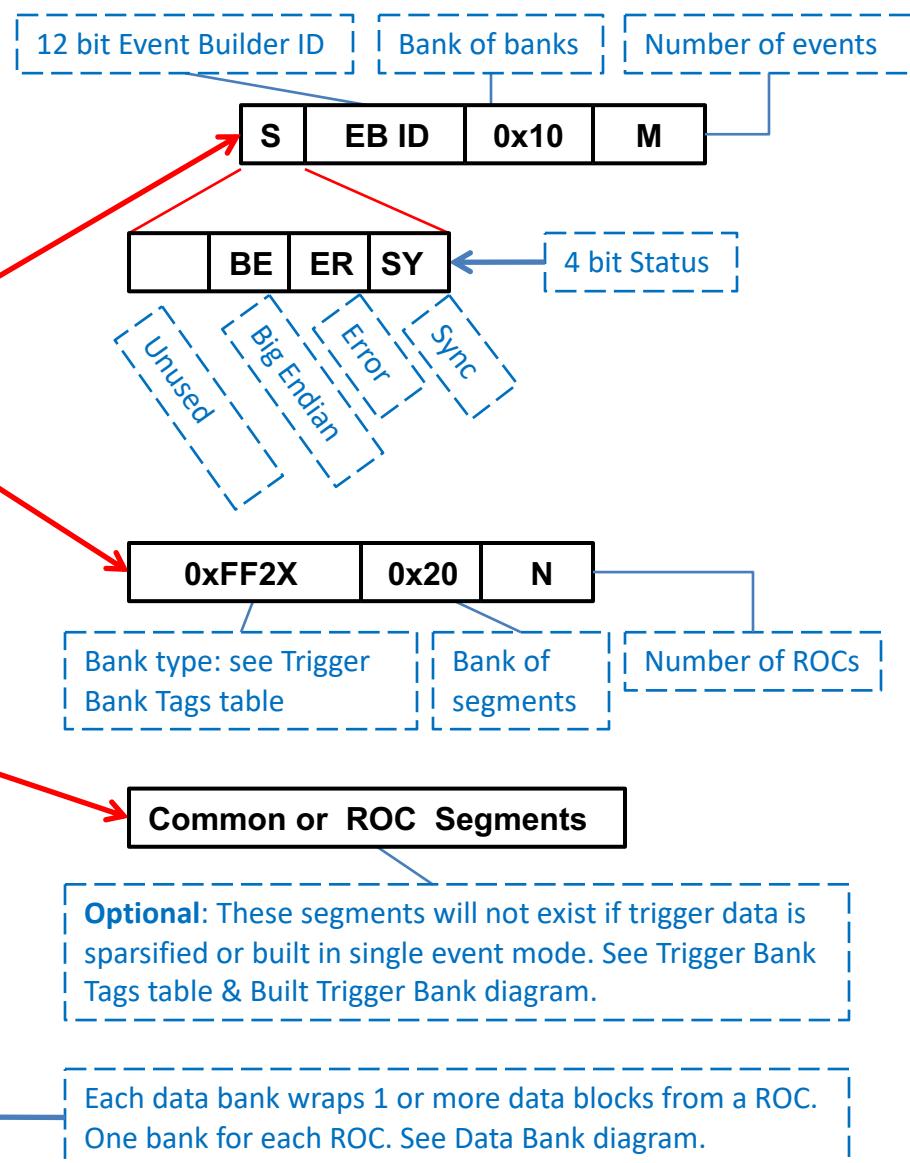
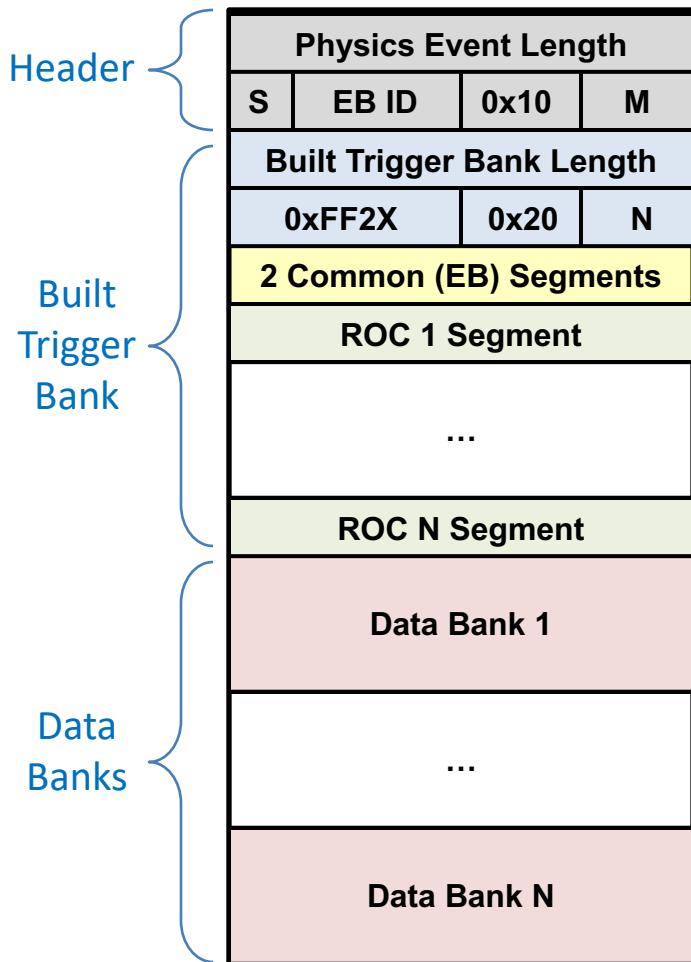
Tag Value Range	Purpose
0xFF00 - 0xFFFF	Complete range of reserved values
0xFFE0 - 0xFFFF	Undetermined
0xFFD0 - 0xFFDF	Control events
0xFF90 - 0xFFCF	Undetermined
0xFF50 - 0xFF8F	Physics events
0xFF10 - 0xFF4F	Trigger banks
0xFF00 - 0xFF0F	Undetermined

4th bit set indicates that the last event in the entangled block is a sync event

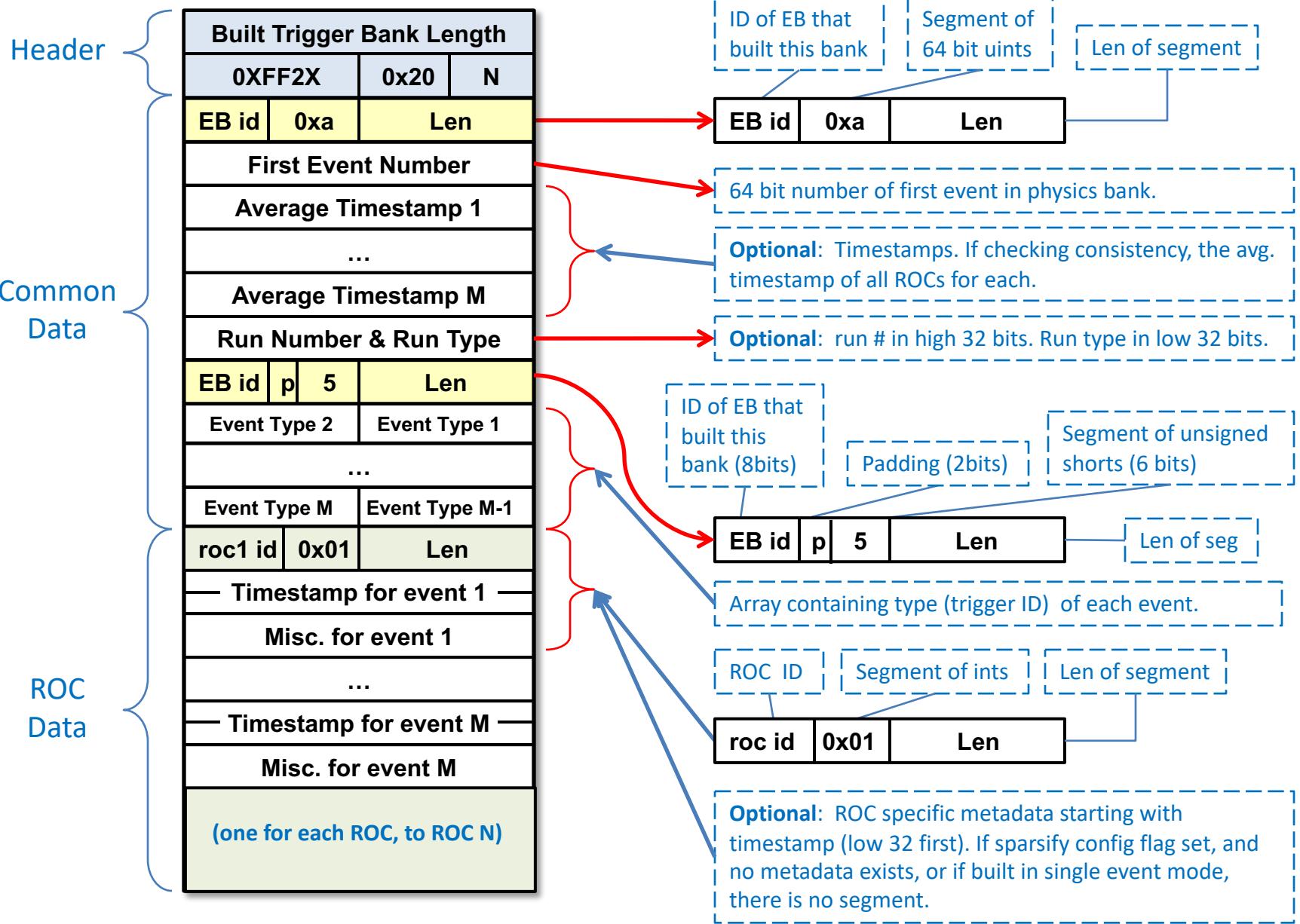
PHYSICS EVENT TAGS

Tag Value	Event Made by
0xFF50	PEB
0xFF58	PEB with sync set
0xFF70	SEB
0xFF78	SEB with sync set
0xFF60	Streaming ROC Raw
0xFF62	Streaming DC
0xFF64	Streaming SEB
0xFF66	Streaming PEB

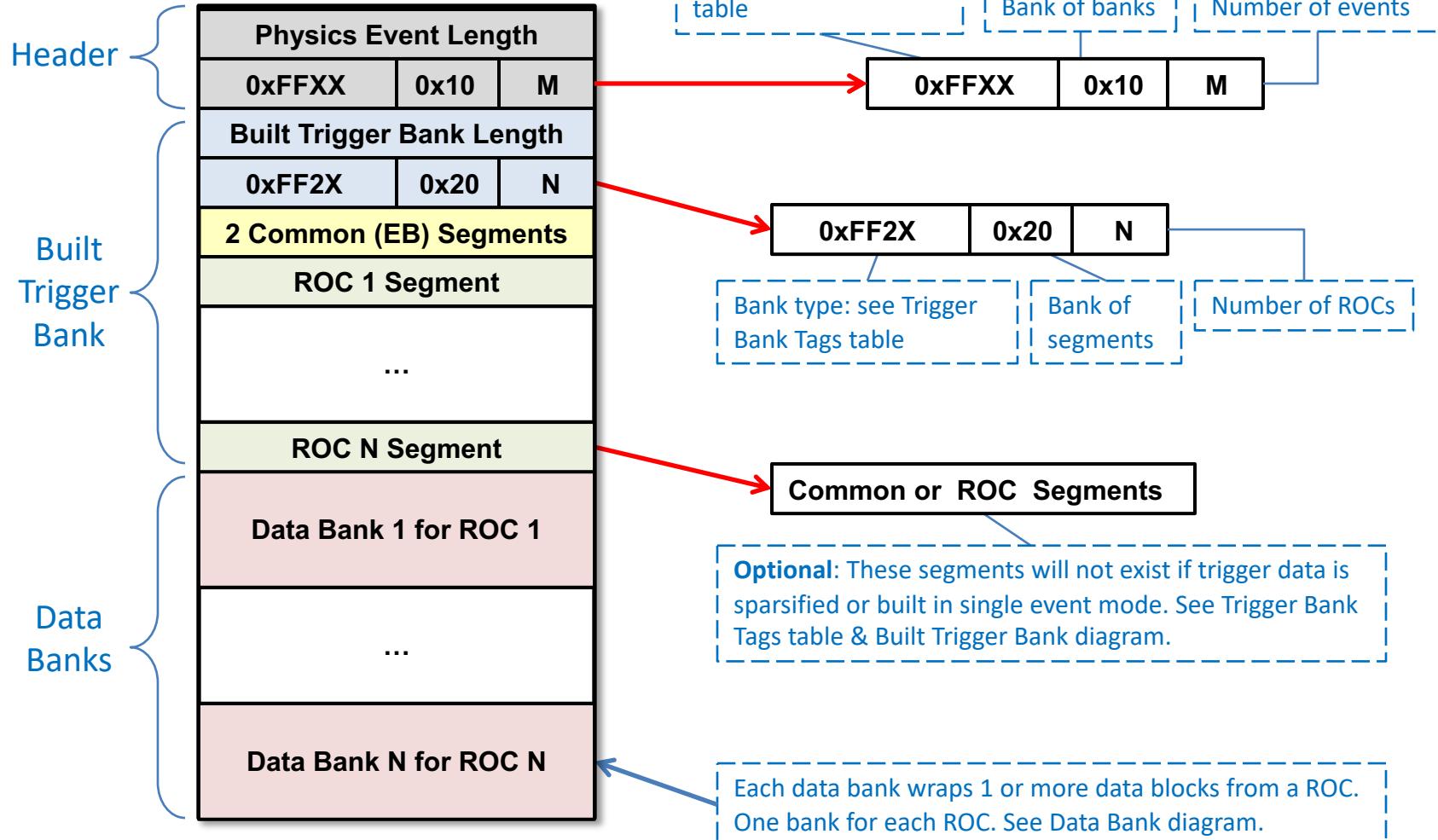
Partially-Built Physics Event (Data Concentrator Output)



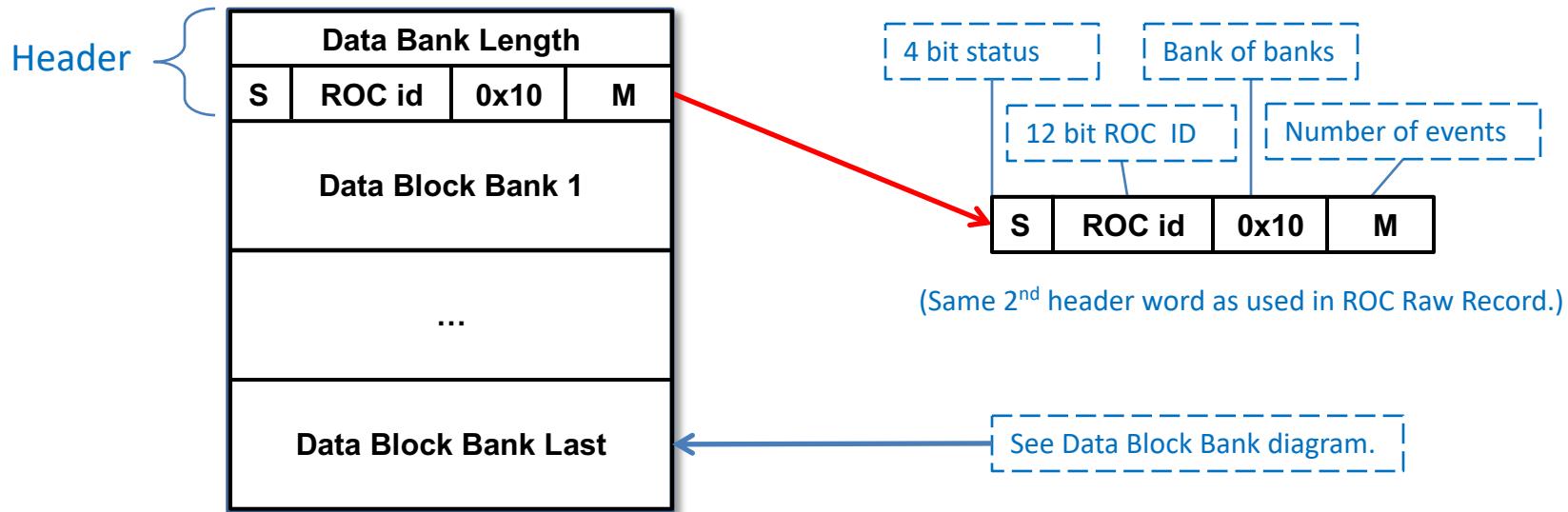
Physics Event's Built Trigger Bank



Physics Event



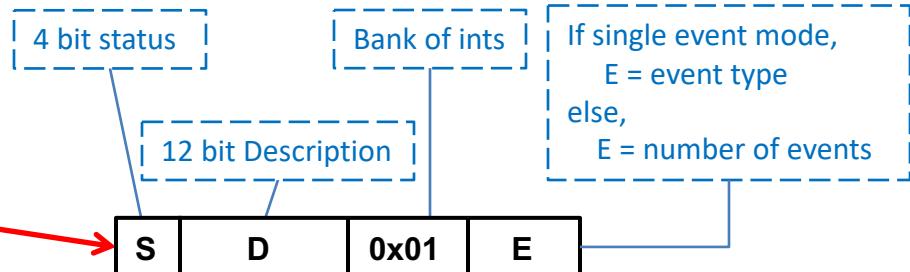
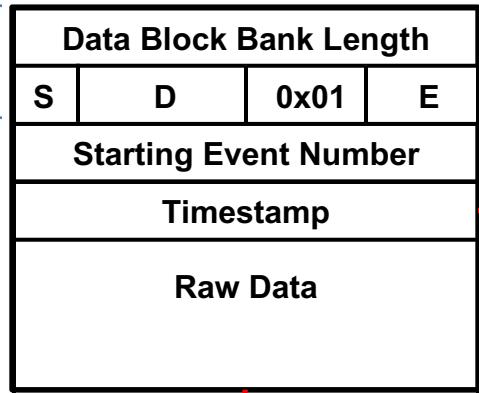
Physics Event's Data Bank



Data blocks from a single ROC are wrapped in this data bank. There should be at least one data block and there may be more if more than one DMA is used in acquiring data for this ROC. If more than one block, each contains a fragment for every one of the M events and from unique modules. In addition, the last block may have data associated only with the last event (such as scalar data).

Data Block Bank

Header



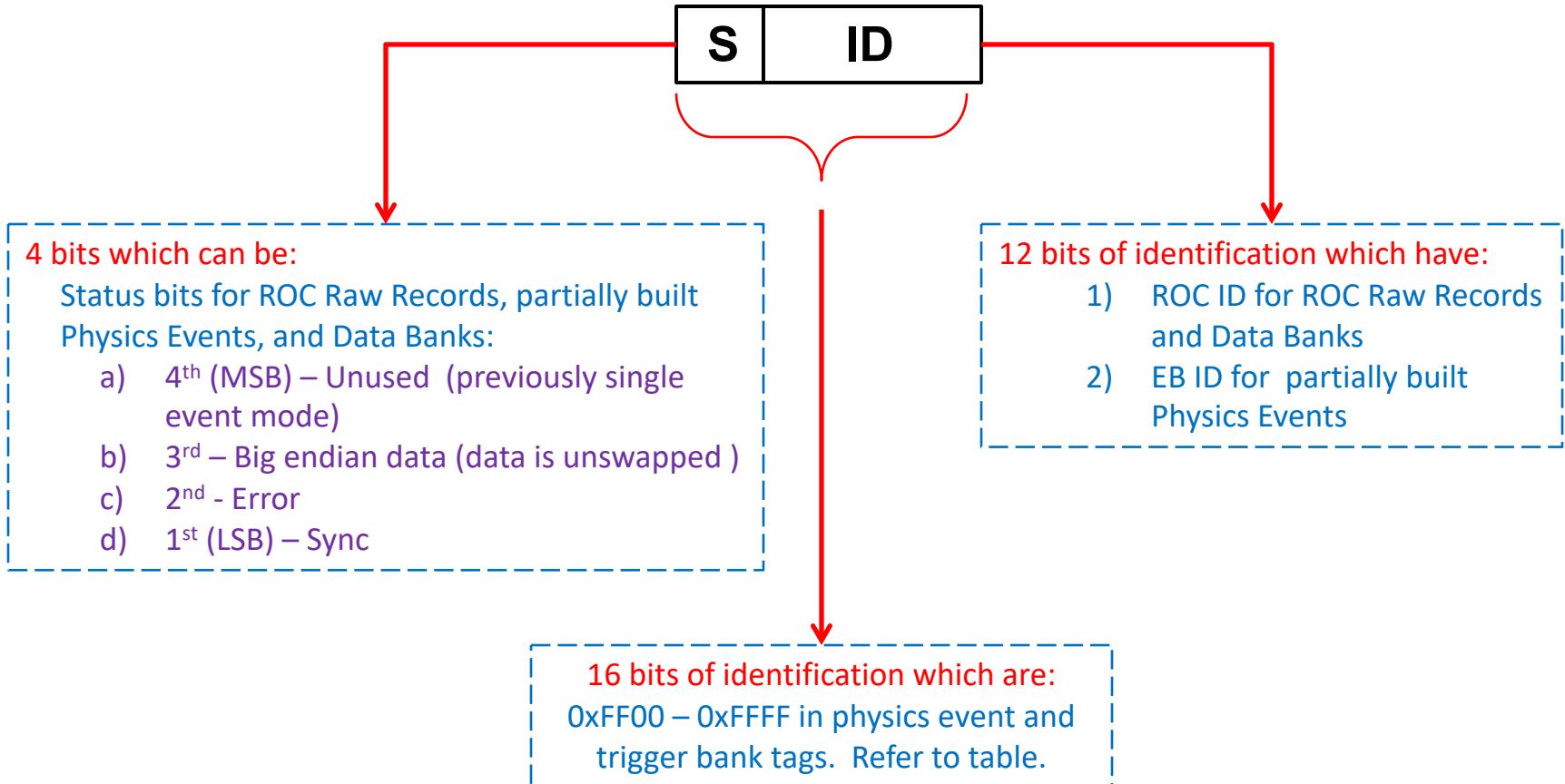
12 bit detector ID

Raw data may be an int array or it may contain evio bank structures.

Low 32 bits

Contains raw data from a single ROC containing one or more events. If this block is the last in a data bank, and there are multiple events, and E = 1, then this data is associated only with the last event (e.g. scalar readout).

16-bit EVIO CODA-Format Tag



Disentangling Built Physics Event

Physics Event

Event Length
Trigger Bank
ROC 1 Data Bank
...
ROC N Data Bank

Data Bank

Bank Length
Data Block Bank 1
...
Data Block Bank Last

Data Block Bank

Block Len
M
Start Ev Num
Raw Data

ROC 1 Raw Data (FADC 250)

Block Header
Event Header 1
Raw Data 1 ...
...
Event Header M
Raw Data M ...
Block Trailer
...
Block Header
Event Header 1
Raw Data 1 ...
...
Event Header M
Raw Data M ...
Block Trailer

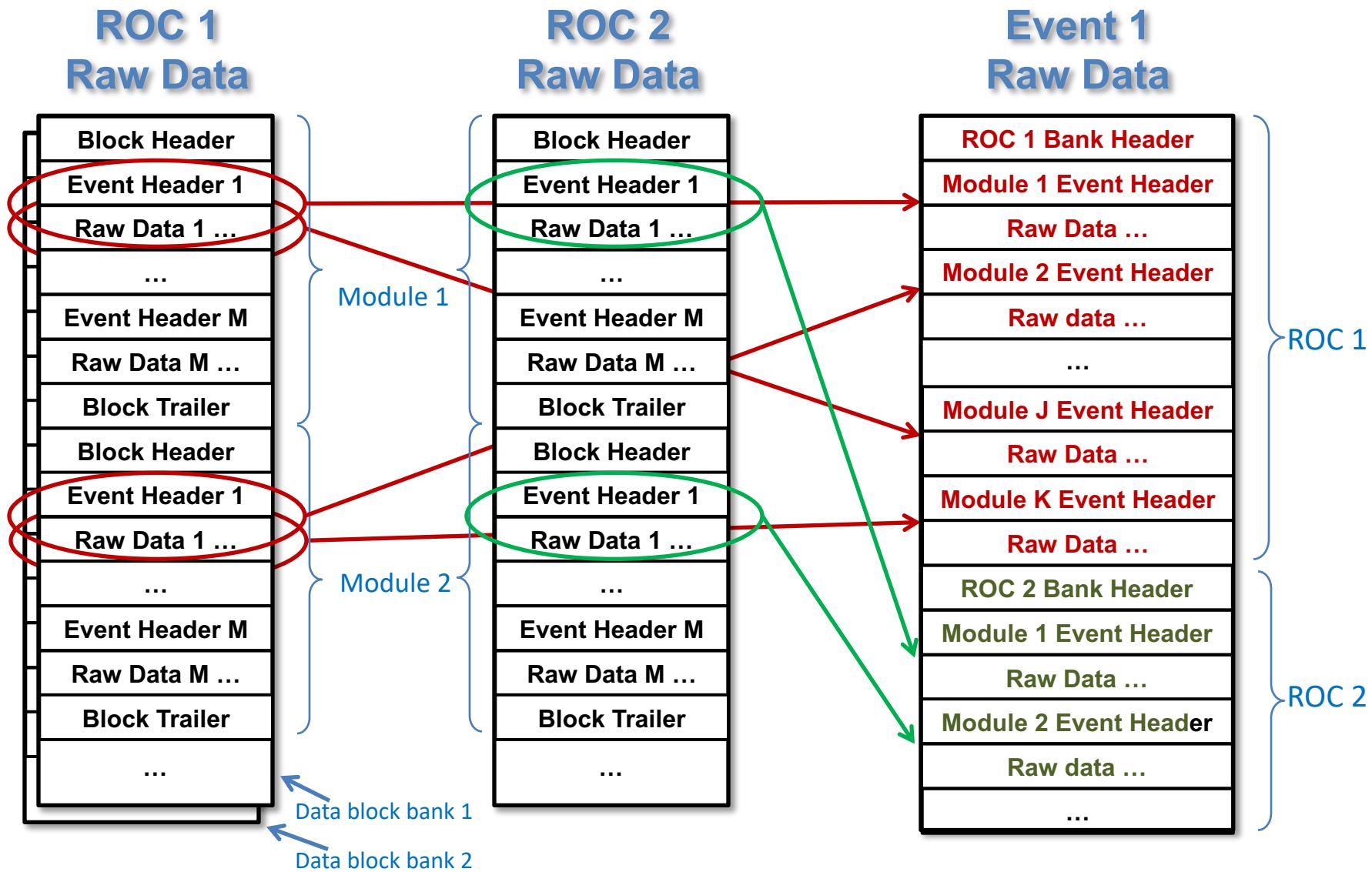
Module 1

Module K

Last block may have data associated only with last event.

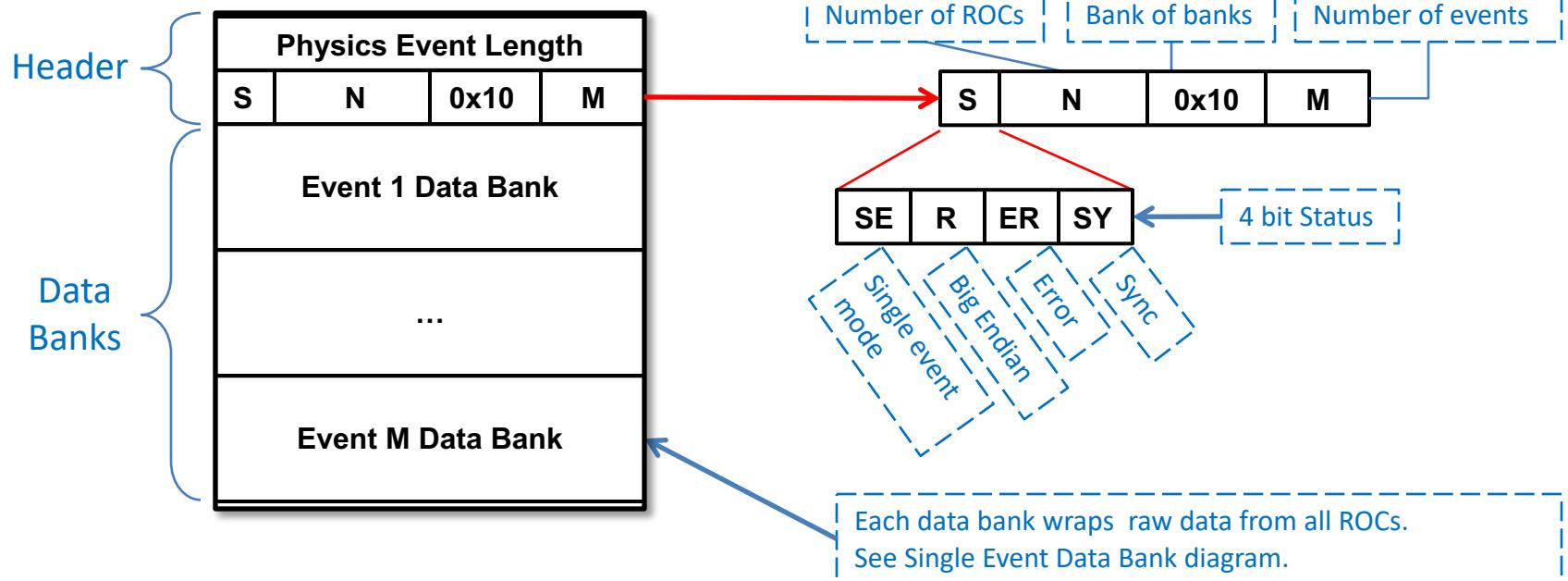
One data block for each DMA in a single ROC. Each block must read out unique modules and in the same manner. In single event mode, only one block exists.

Entangled Data Format

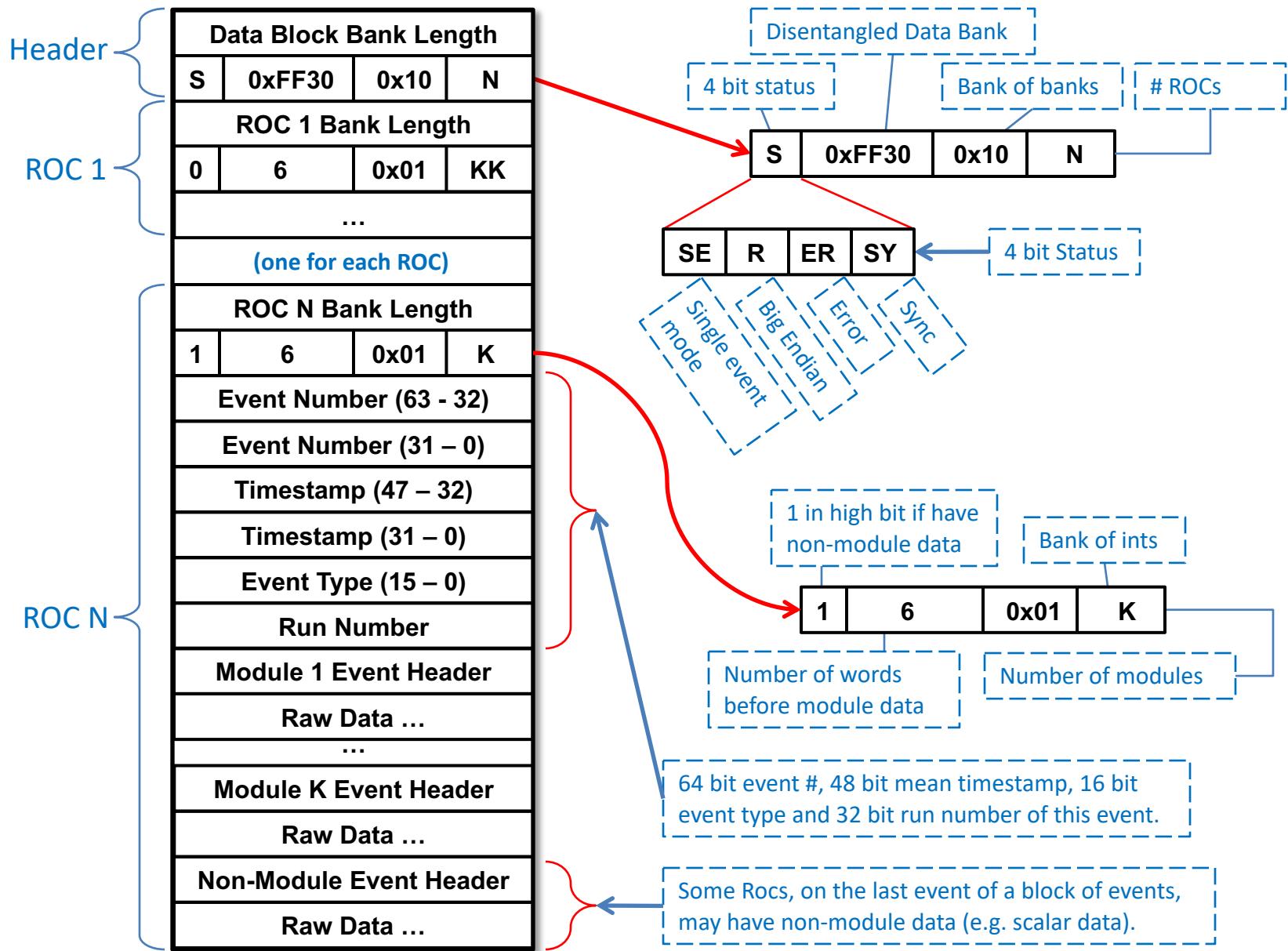


Entangled To Disentangled FADC 250 Raw Data

Disentangled Physics Event



Single Event (Disentangled) Data Bank



FADC 250

General Data Word Format

31 st bit	Bits	Usage
1	30 - 27	4-bit data type (see chart)
1	26 - 0	Data type dependent data payload
0	30 - 0	Data payload using last defined data type

Data Type Values

0 – block header	7 – pulse integral
1 – block trailer	8 – pulse time
2 – event header	9 – streaming raw data
3 – trigger time	10 – 12 user defined
4 – window raw data	13 – event trailer (debug only)
5 – window sum	14 – data not valid (empty module)
6 – pulse raw data	15 – filler (non-data) word

Block Header Word Format

Bits	Value	Comment
31	1	This is a type defining word
30 – 27	0	Data type = block header
26 – 22	Slot ID	Set by VME64 backplane
21 – 14	Event #	Number of events in block
13 – 12	Module Type	0=FADC250, etc.
11 – 0	Event block #	Used to align block when building events

Block Trailer Word Format

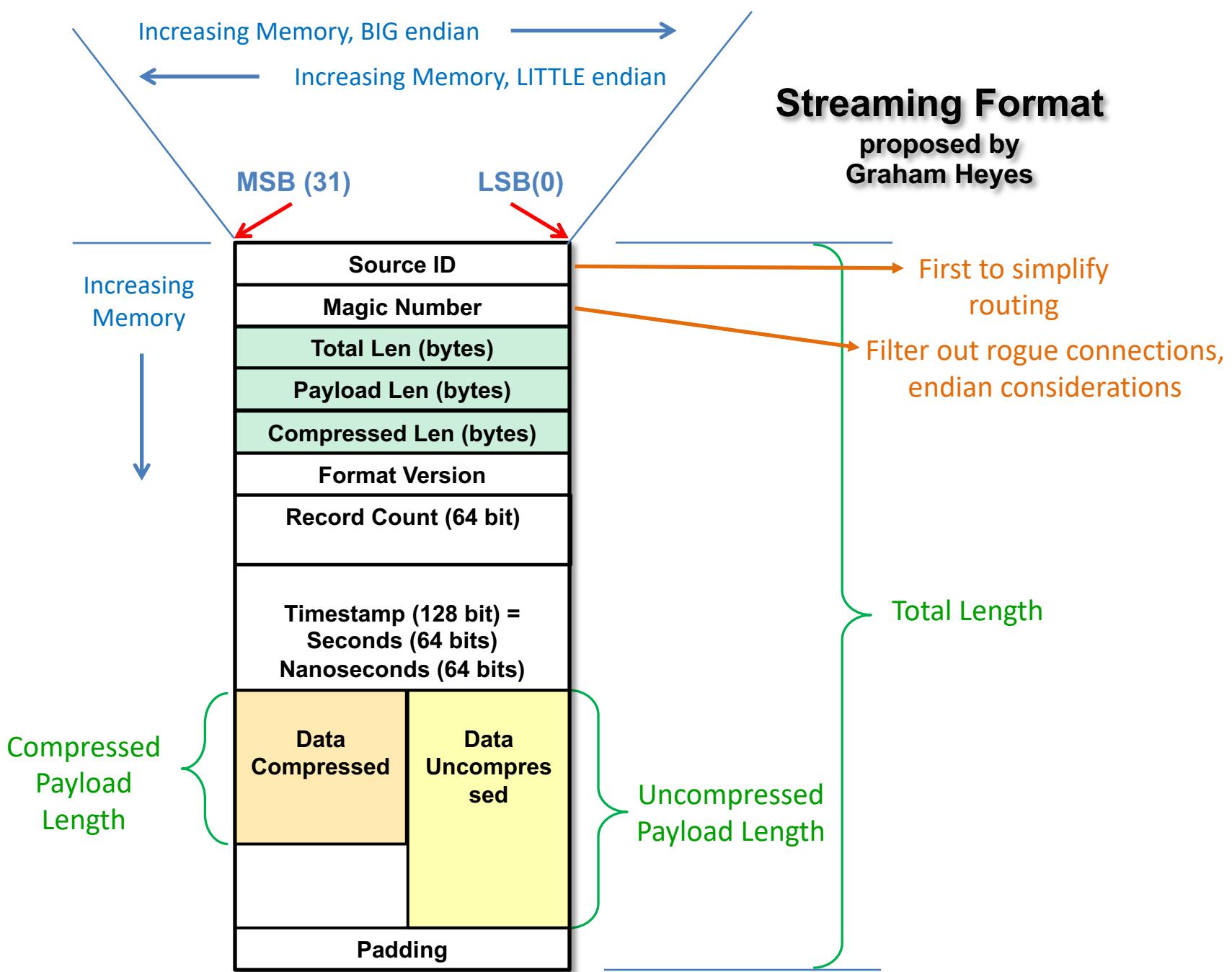
Bits	Value	Comment
31	1	This is a type defining word
30 – 27	1	Data type = block trailer
26 – 22	Slot ID	Set by VME64 backplane
21 – 0	Total # of words in block of events	Number of 32 bit words in block

Event Header Word Format

Bits	Value	Comment
31	1	This is a type defining word
30 – 27	2	Data type = event header
26 – 22	Slot ID	Set by VME64 backplane
21 – 20	Module type	0=FADC250, etc.
19 – 0	Trigger number	ADC processing chip #

Streaming Format

proposed by
Graham Heyes



Possible Streaming Format

