



Simple Enumeration and Control Protocol

Ashley Butterworth
Apple Inc.

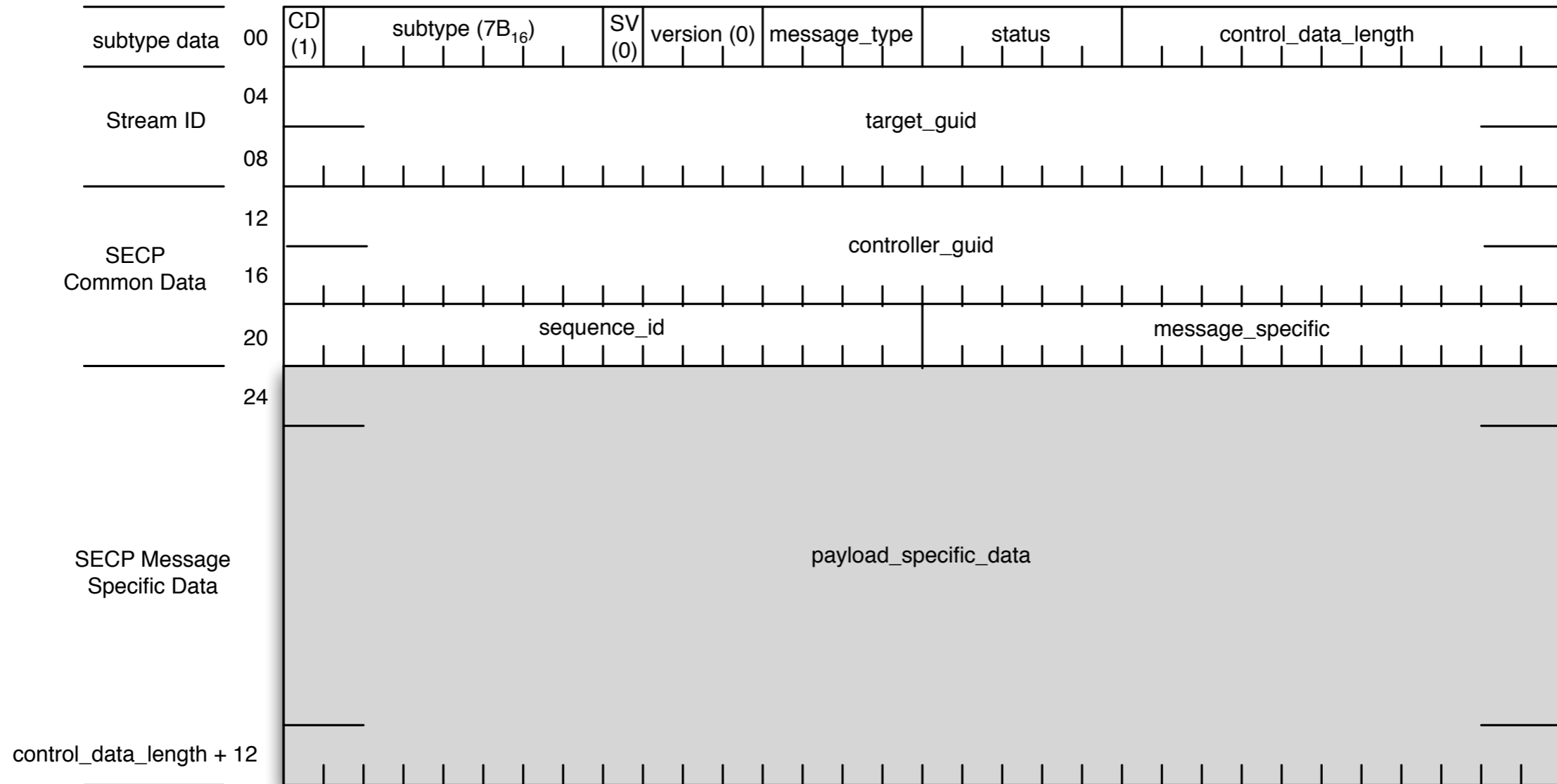
Intro

- SECP is a transaction based protocol supporting a number of enumeration and control actions.
- Uses multicast 1722 control packets
 - Doesn't require device MAC lookup
 - Allows controllers to update states without polling for changes

Supported Payloads

- 1722.1 Enumeration and Control
 - JKP?
- Memory Read and Write
 - Read or write part of a device address map
- Legacy AVC
 - Carries IEEE1394 AV/C commands and responses
- Vendor Unique
 - Allow a vendor to define a custom protocol for non-standard device features

SECPDU Format



Common Fields

- 1722 Control Header
 - control_data is message_type
 - stream_id is target_guid
- controller_guid
 - controller responsible for sending the command
- sequence_id

Message Defined Fields

- message_specific
 - usage is defined by the message type
- payload_specific_data
 - defined by message type
 - minimum of 26 bytes in size
 - maximum of 512 bytes in size

Message Types

- 17221_ECP_COMMAND
- 17221_ECP_RESPONSE
- MEM_READ_COMMAND
- MEM_READ_RESPONSE
- MEM_WRITE_COMMAND
- MEM_WRITE_RESPONSE
- AVC_COMMAND
- AVC_RESPONSE
- VENDOR_UNIQUE_COMMAND
- VENDOR_UNIQUE_RESPONSE

Common Status codes

- **SUCCESS**
 - Command was performed successfully.
- **NOT_IMPLEMENTED**
 - The device does not support the command type.
 - Optional to send, but it's nicer than timing out!

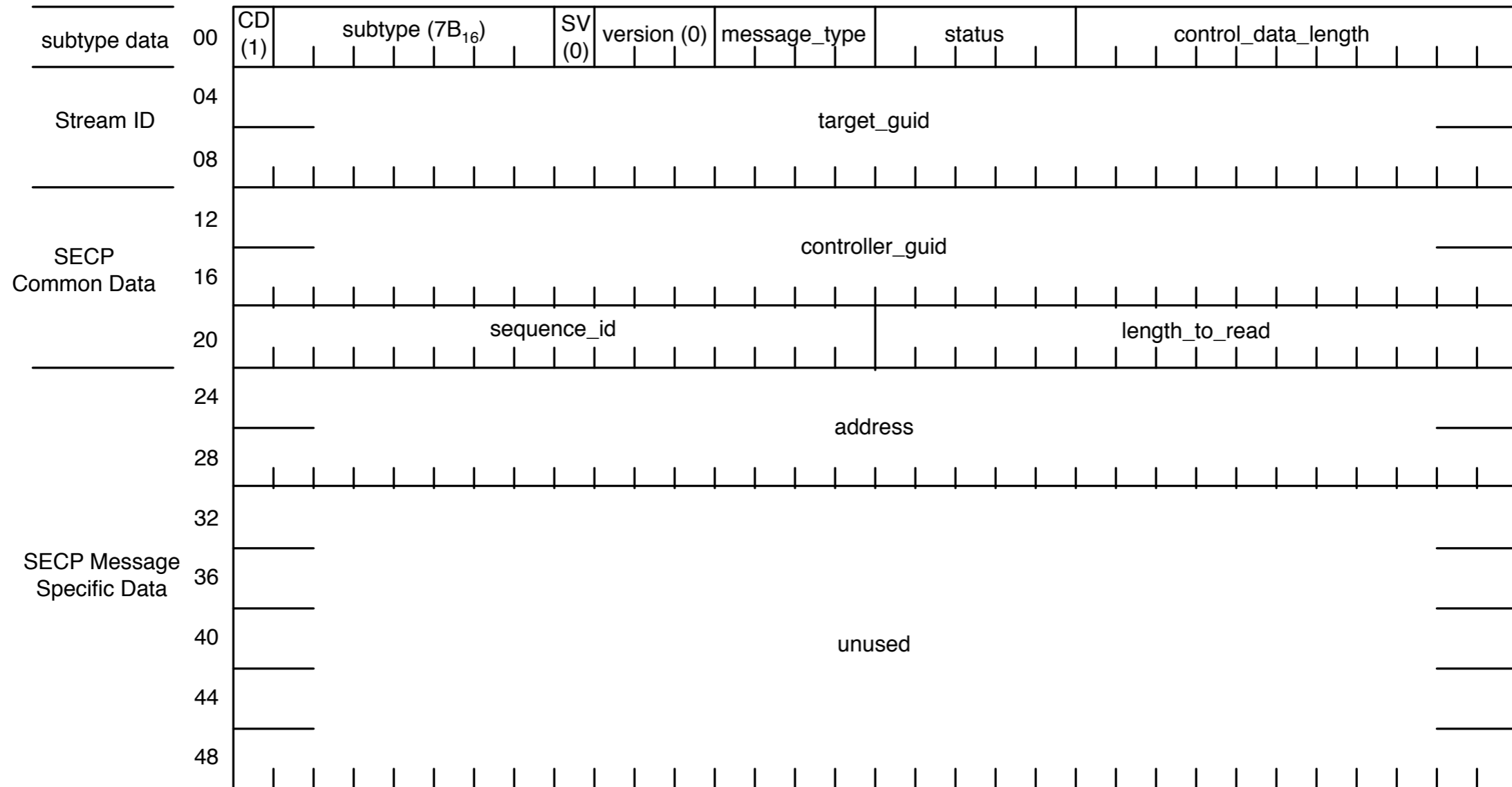
1722.1 EC Command

- Still to be defined when we have an enumeration and control protocol
- JKP command?

1722.1 EC Response

- Still to be defined when we have an enumeration and control protocol
- JKP response?

Memory Read Command SECPDU

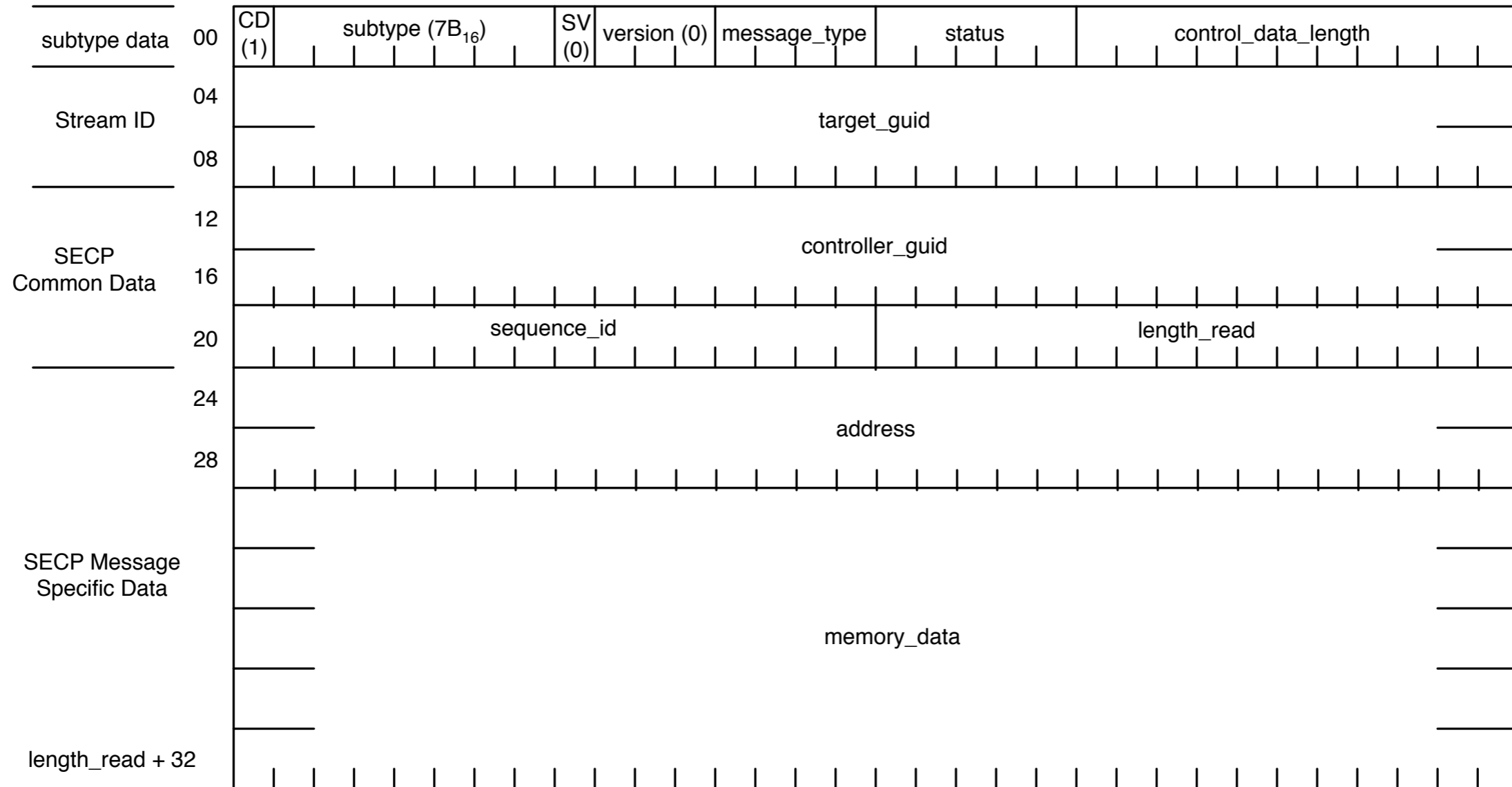


Memory Read Command

- message_specific is used as length_to_read
- address contains base address of the read
- message is padded with unused (reserved?) quadlets to be at least 64 bytes in length

- Can be unicast or multicast

Memory Read Response SECPDU

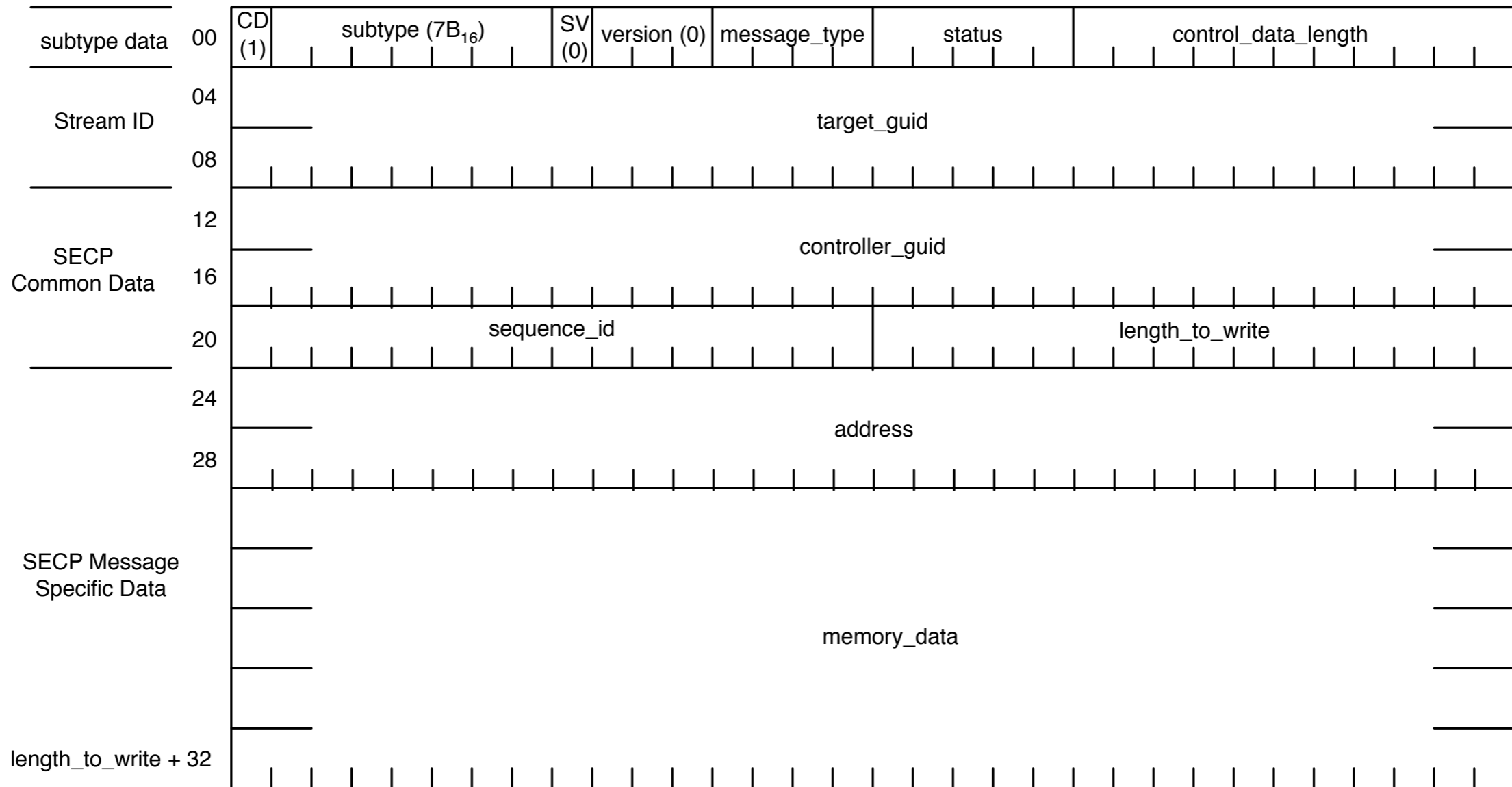


Memory Read Response

- `message_specific` is used as `length_read`
- `address` contains base address of the read
- `memory_data` must be padded with 0 bytes to obtain minimum 64 byte packet

- If command was unicast then the response is unicast, otherwise it is multicast

Memory Write Command SECPDU

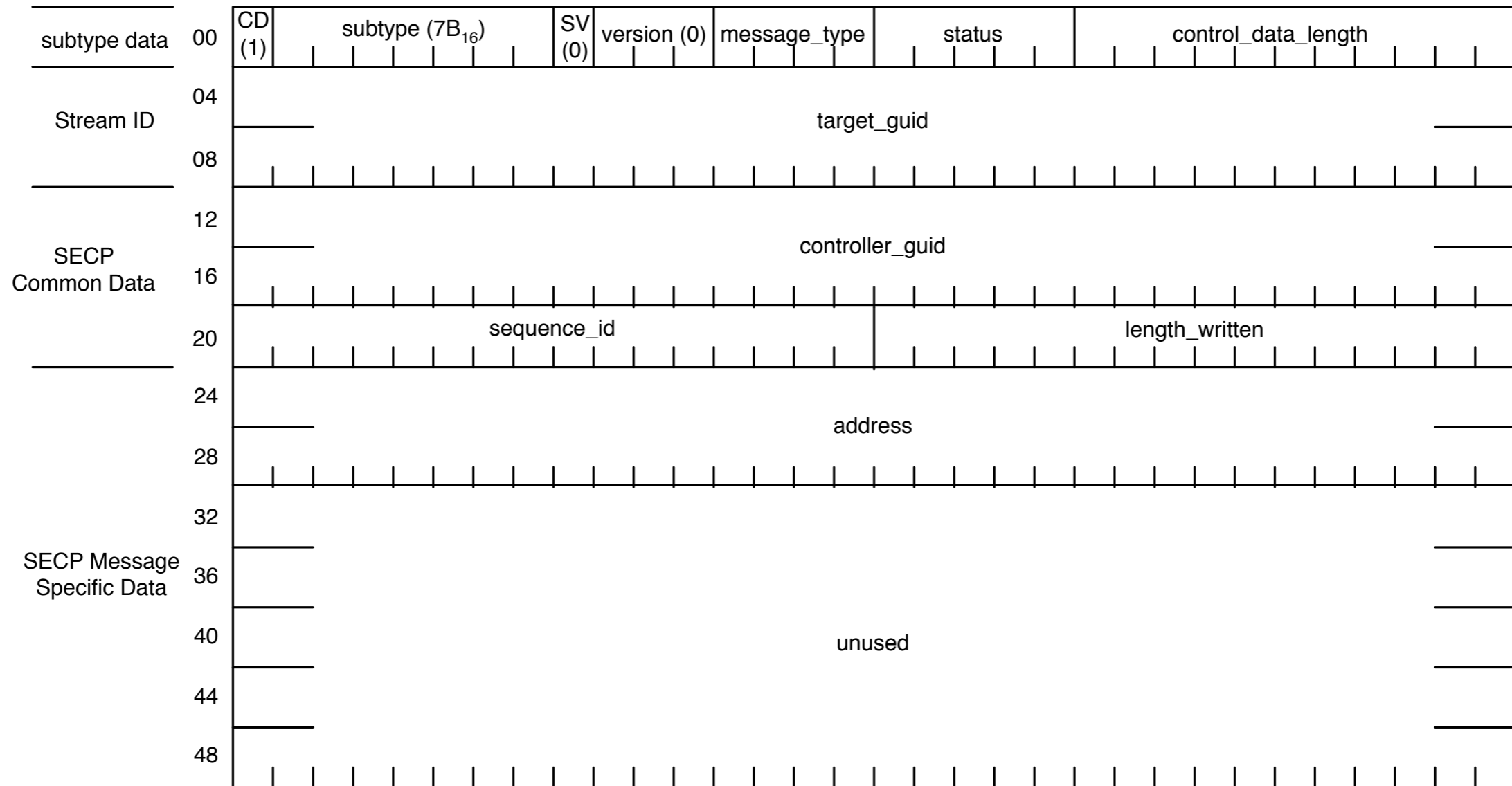


Memory Write Command

- `message_specific` is used as `length_to_write`
- `address` contains base address of the write
- `memory_data` must be padded with 0 bytes to obtain minimum 64 byte packet

- Can be unicast or multicast

Memory Write Response SECPDU

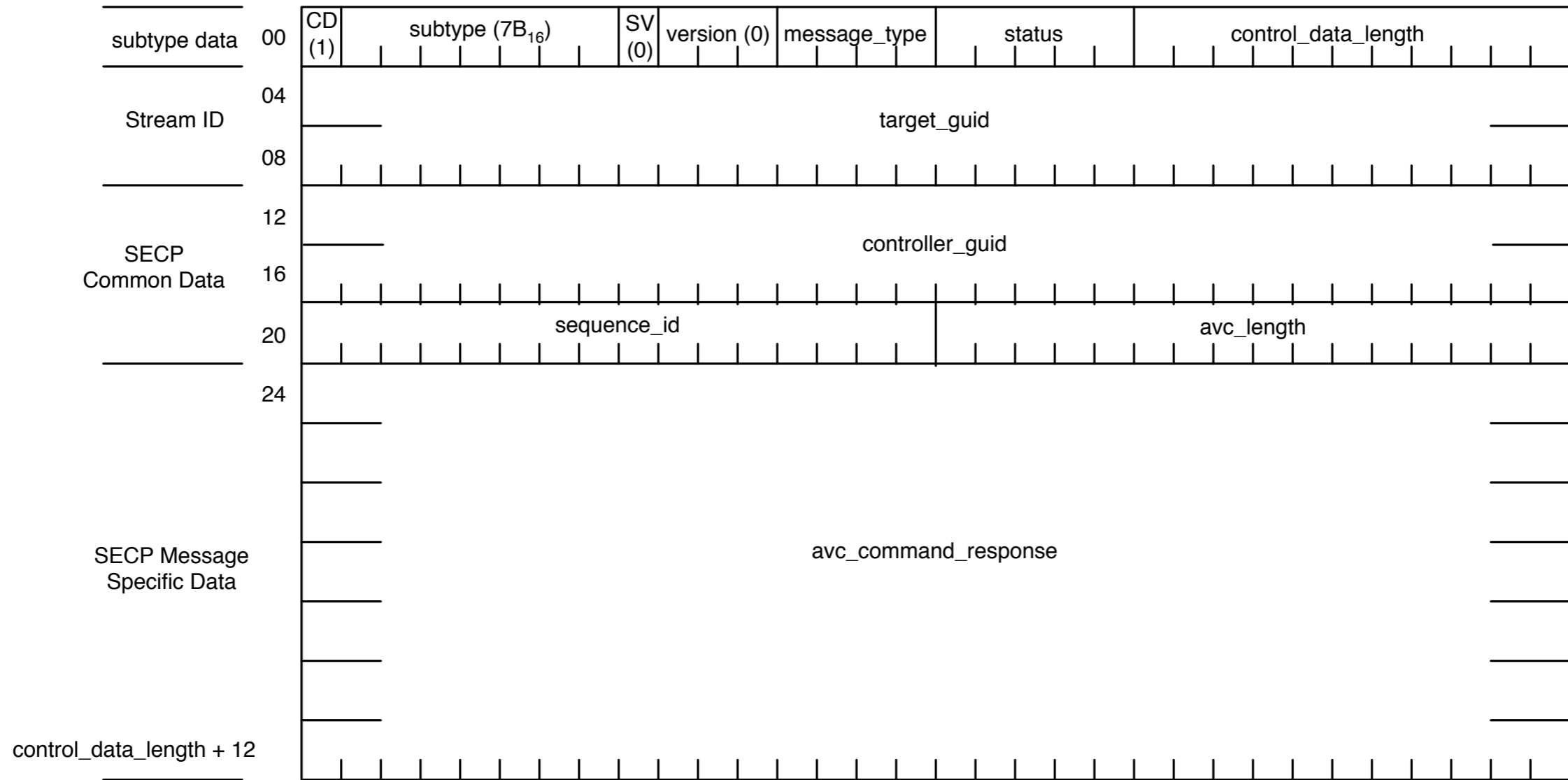


Memory Write Response

- message_specific is used as length_written
- address contains base address of the write
- message is padded with unused (reserved?) quadlets to be at least 64 bytes in length

- If command was unicast then the response is unicast, otherwise it is multicast

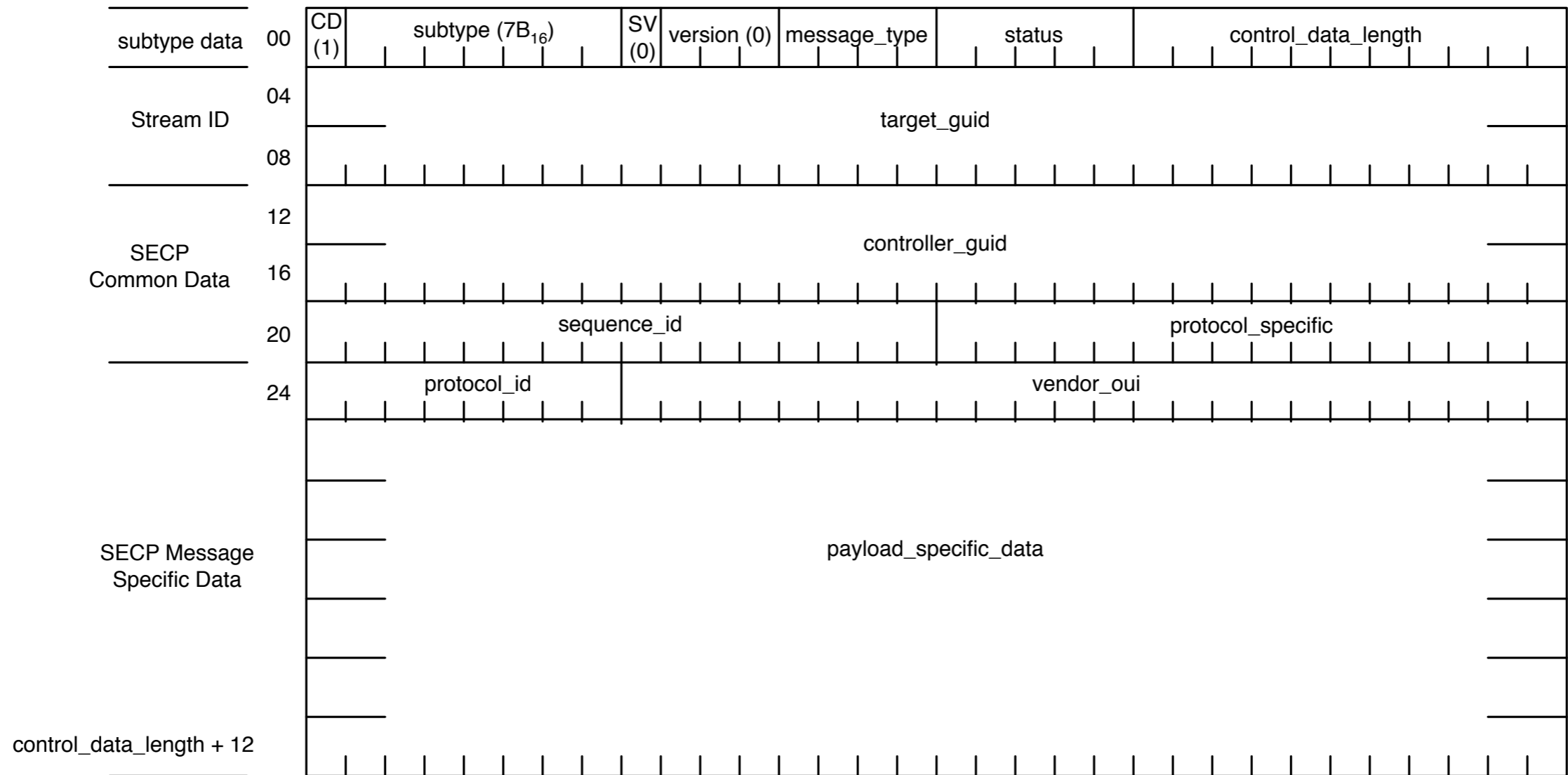
AVC Command/Response SECPDU



AVC Command/Response

- `message_specific` is used as `avc_length`
 - this is the length of valid AV/C data in command/response
- `avc_command_response` contains AV/C command or response data as defined by IEEE1394 standards.
 - command/response data is padded with 0s to be minimum 64 byte packet.

Vendor Unique SECPDU



Vendor Unique

- message_specific is protocol_specific
- protocol_id is 8 bit ID for protocol assigned by vendor
 - allows vendor to define multiple protocols
- vendor_oui
 - OUI of the vendor who designed the protocol
- payload_specific_data
 - minimum of 22 bytes of data