Mark 5C Data-Frame Specification

(V1.0   19 Feb 2008)

MIT Haystack Observatory   National Radio Astronomy Observatory

Westford, MA 01886   Socorro, NM 87801

1. Introduction

The Mark 5C is basically a 'formatless' Ethernet packet recorder, so it is up to the user to define the data contents and format. As described in the "Mark 5C Specification" document, each Ethernet packet carries one Data Frame to be recorded to disk; due to potential missing or out-of-order packets, each Data Frame must stand on its own with respect to time-tagging and identification. This document describes the contents and format of a Mark 5C Data Frame that meets these requirements. It is the responsibility of the Digital Data Source (DDS), such as a VLBI digital backend system, to create Data Frames in the prescribed format. Also described is a Data Frame format to create disk modules which are compatible with the Mark 5B playback system required by legacy hardware correlators.

2. Data-Organization philosophy

Previous generations of VLBI data recorders, such as Mark 3/3A/4/5A/5B have generally formatted multiple observing frequency channels as multiple-parallel bit-streams representing simultaneous samples from all channels. Also, typically, the per-channel data rate is constrained to 2^n Mbps and the number of channels is also usually constrained to be integer powers of 2, thereby also constraining total data rates to be 2^n Mbps. This type of data organization has been satisfactory for the type of hardware correlators that have been predominant over the past 30 years, but is not well matched to today's needs.

Modern VLBI data management relies increasingly heavily on the use of standardized local and global computer networks whose capacities are not well matched to the 2^n Mbps paradigm of established VLBI data systems. Furthermore, VLBI correlation processing is moving rapidly towards fully software solutions operating on clustered computer networks. Both of these trends argue for improved flexibility in aggregate VLBI data rates and repackaging of data into more convenient forms for processing by standard computer hardware.

3. Mark 5C VLBI Data-Frame format

In order to meet the needs of modern VLBI, the Mark 5C data format has been designed according to the following guidelines:

1. Each Ethernet packet arriving at the Mark 5C contains one self-identifying Data Frame.
2. Each Data Frame contains a 16-byte Data Frame Header followed by a Data Array.
3. The Data Frame length must be constant during a given scan (defined as the period between starting and ending the recording of a particular observation), but can vary from scan to scan if necessary.
4. Each Data Frame contains sample data from one frequency channel only.
5. The Data Frame length (including 16-byte header) must meet the following three criteria:
   a. Must be within the range 64-9000 bytes
   b. Must be a multiple of 8 bytes for compatibility with the StreamStor disk-addressing algorithm
   c. Must result in an integer number of complete Data Frames per second

3.1 Data Frame Header Format

The proposed format of a Mark 5C data frame is shown in Table 1.

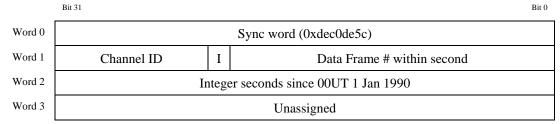| Bit 31			Bit 0
Word 0	Sync word (0xdec0de5c)		
Word 1	Channel ID	I	Data Frame # within second
Word 2	Integer seconds since 00UT 1 Jan 1990		
Word 3	Unassigned		

Table 1: Data Frame Header format

The words within the Data Frame Header are assigned as follows:

   Word 0 – A fixed synchronization word

   Word 1

   Bits 31-24: Channel ID (normally a frequency-channel number, starting at zero)

   Bit 23: I – Invalid data (i.e. data in this Data Frame is tagged Invalid); not to be confused with the invalid bit in the Packet Serial Number which is used to prevent a Data Frame from being written to disk.

   Bits 22-0: Data Frame # within second (resets to zero on each second tick)

   Word 2 – Integer seconds since 00UT 1 Jan 1990 (same as RTP time-stamp used by VSI-E)

   Word 3: unassigned

Word 3 can be assigned by the user as desired. For example, the Frame Sequence Number, as described in the 'Mark 5C Specification' document, could be assigned to Word 3 and recorded as part of the Data Frame Header.

3.2 Data Array Format

The Data Array, which immediately follows each Data Frame Header, contains the sample data from a *single frequency channel*.

The Data Array format for 1, 2, 3, and 4 bits/sample are shown in Tables 2 through 5; larger number of bits/sample can be allocated similarly, but are not considered here. The first sample in the Data Array corresponds precisely to the time indicated in the Data Frame Header. Like the bits comprising frame headers, unused data-word bits do not contribute to the rated aggregate data bit rate.

Bit 31																															Bit 0
31																															0

   Table 2: 1-bit/sample data-word format (numbers in boxes indicate relative sample #'s in time order)

Bit 31															Bit 0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 3: 2 bits/sample data-word format

Bit 31										Bit 0
xx	9	8	7	6	5	4	3	2	1	0

Table 4: 3 bits/sample data-word format. 'xx' indicates unused bits

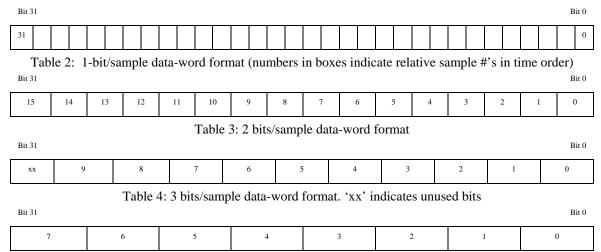| Bit 31							Bit 0
7	6	5	4	3	2	1	0

Table 5: 4 bits/sample data-word format

This general scheme can be extended to an arbitrary number of bits/sample, but a subtlety arises which must be recognized: For standard VLBI use, the number of samples/word must divide evenly into 2^n*10^6 samples/sec, which can never be satisfied for 5, 9 or 10 bits/sample. So the only legal bits/sample in this regard are: 1-4, 6, 7 and 11-32. Interestingly, of all of these possibilities, only 3-bit and 6-bit samples have any storage-efficiency improvements over padding the bits/sample to the next power of 2. Of course, other applications that use non-VLBI-standard sample rates may have different restrictions.

The length of the Data Array may be specified by the user, but must be a multiple of 8 bytes. In some cases of low aggregate data rate, such as 2 Mbps or 4 Mbps, care must be taken to choose a length which guarantees an integer number of Data Frames per second. For example, a Data Array length of 8000 bytes cannot be supported at 2 Mbps (corresponds to 31.25 Data Frames per second), but a length of 5000 bytes (50 Data Frames per second) is suitable.

3.3 Sample representation

Samples delivered by a modern DDS have already passed through substantial digital processing, and potentially can be carried with numerous bits of precision in cases where bandwidth limitations are not a consideration. Thus, it makes little sense to define specialized coding schemes based on the input sample thresholds. Further, Mark 5 recording is not subject to the paired-bit errors that caused the VLBA and Mark 4 tape formats to adopt a sign-symmetrized coding scheme. Accordingly, we suggest that Mark 5C data samples be represented simply by the desired number of bits of a fixed-point twos-complement numerical value. For example, 2-bit/sample coding is (in order from most negative to most positive) 10, 11, 00, 01. For all multi-bit samples, the LSB of each sample occupies the rightmost bit (i.e. lowest bit number) of the sample field.

3.4 'Fill-pattern' Data Frame format

Under certain circumstances, the Mark 5C writes a 'fill-pattern Data Frame' in place of missing data. The fill-pattern Data Frame consists of a normal-length Data Frame *completely* filled with a user-specified 32-bit pattern. The correlator recognizes this fill-pattern data to prevent the correlator from processing the corresponding data segment.

4. Mark 5B-Emulation Data Frame Format

In order to create recorded disk modules which can be re-played on a Mark 5B playback system, the Digital Data Source (DDS) must also be capable of creating Data Frame compatible with the Mark 5B playback system. Similar to the Mark 5C, the Mark 5B Data Frame contains a 16-byte Data Frame Header followed by a fixed 10,000-byte Data Array.

4.1 Mark5B-emulation Data Frame Header Format

The format of the non-data-replacement Mark5B-compatible Disk Frame header is shown in Table 6; each Disk Frame header contains the following information:

    Word 0 – A fixed synchronization word (0xabaddeed)
    Word 1
        Bits 31-28: Integer years since 2000 (to resolve year ambiguity of VLBA time code in Word 2)
        Bits 27-16: User-specified data
        Bit 15: T – Data are from test-vector generator (tvg)
        Bits 14-0: Data Frame # within second (starts at zero on second tick)
    Word 2 – VLBA BCD Time code to one-second resolution
    Word 3 – Unassigned

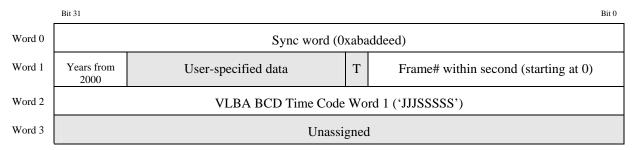|	Bit 31				Bit 0
Word 0	Sync word (0xabaddeed)				
Word 1	Years from 2000	User-specified data	T	Frame# within second (starting at 0)	
Word 2	VLBA BCD Time Code Word 1 ('JJJSSSSS')				
Word 3	Unassigned				

Table 6: Mark5B-emulation Disk Frame Header format

The shaded fields in Table 6 are ignored by the Mark 5B playback system and need not be written by the Mark 5C when creating disk data that emulates the Mark 5B data format. (The standard Mark 5B Disk Frame Header prescribes Word 3 as the VLBA sub-second time code as a 4-digit BCD number plus a 16-bit CRCC. This 4-digit BCD representation is sometimes insufficient for precise time-tagging of the data, but was needed by the Mark 5A+ system to create a VLBA tape-track format during playback of Mark 5B data disks. Users may use Word 3 as desired by appropriate design of the DDS that supplies data to the Mark 5C.)

4.2 Mark 5B Data Array Format

The Data Array format for 1, 2, 4, 8, 16 and 32 active bit-streams are shown in Tables 7 through 12, respectively. Each 32-bit data word contains one or more bits from *all* active bit streams. The first data bit from each active bit-stream in the Data Array corresponds precisely to the time indicated in the Frame Header.

Bit 31   Bit 0

31																															0

Table 7: 1-bit-stream data word format (32 samples from single bit stream)

Bit 31   Bit 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 8: 2 bit-stream data word format (16 samples from each of 2 bit-streams)

Bit 31   Bit 0

7	6	5	4	3	2	1	0

Table 9: 4 bit-stream data word format (8 samples from each of 4 bit streams)

Bit 31   Bit 0

3	3	1	0

Table 10: 8 bit-stream data word format (4 samples from each of 8 bit streams)

Bit 31   Bit 0

1	0

Table 11: 16 bit-stream data word format (2 samples from each of 16 bit streams)

Bit 31   Bit 0
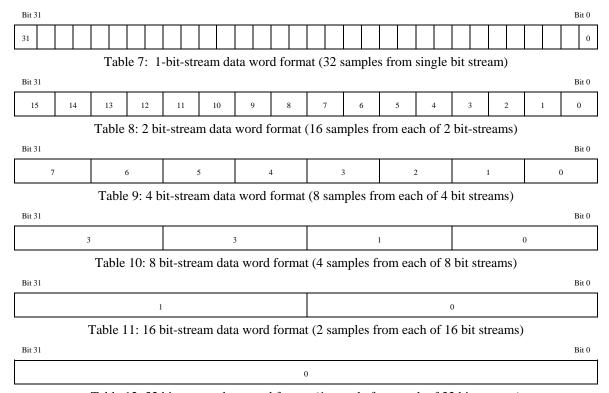
0

Table 12: 32 bit-stream data word format (1 sample from each of 32 bit streams)

4.3 <u>Sample coding</u>

Mark 5B emulation mode is limited to one or two bits per sample. This mode uses the same coding as is used for actual Mark 5B recording, which is shown in Table 13. For 1-bit samples, only the Sign bit is encoded.

Signal level	Sample code	
	Sign (S)	Mag (M)
>Threshold	1	1
0→Threshold	1	0
0→ neg Threshold	0	1
< neg Threshold	0	0

Table 13: Sample codes for Mark 5B emulation mode

4.4 <u>Recording in Mark 5B Emulation mode</u>

Several precautions should be taken when writing Mark 5B-compatible data to a Mark 5C system:

1. The Mark 5B Data Frame is longer than can be contained in a single Ethernet packet, therefore it must be written in two or more equal-length segments (such as two 5008-byte segments).

2. The Mark 5B playback system demands that all Data Frames be present and in order, though fill-pattern Data Frames may be written in place of missing Data Frames. For this reason, the use of the Mark 5C 'Frame Sequence Number monitor mode 1' is highly recommended during recording. This mode writes fill-pattern Data Frames in place of any missing frames.

3. Mark 5B fill-pattern Data Frames are generated exactly like Mark 5C, but with a fixed length of 10016 bytes. The Mark 5B fill-pattern word is (0x11223344).

<u>References</u>

"Mark 5C Specification", authored jointly by Haystack and NRAO, available at
http://www.haystack.edu/tech/vlbi/mark5/memo.html as memo #57 and also at
http://www.vlba.nrao.edu/memos/sensi/sensimemo13.pdf

"Mark 5B System User's Manual", available at
http://www.haystack.edu/tech/vlbi/mark5/docs/Mark%205B%20users%20manual.pdf