Introduction to the K5 System

Mamoru Sekido, Yasuhiro Koyama and Tetsuro Kondo

Kashima Space Research Center
National Institute of Information and Communications Technology, Japan

1. Concept of the K5 System

The K5 VLBI system is designed to perform real-time or near-real-time VLBI observations and correlation processing using Internet Protocol over commonly used shared network lines. Various components are being developed to realize the target goal in various sampling modes and speeds. The entire system will cover various combinations of sampling rates, number of channels, and number of sampling bits. All of the conventional geodetic VLBI observation modes will be supported as well as the other applications like single-dish spectroscopic measurements or pulsar timing observations will also be supported. The concept as the family of the K5 system is show in the Fig. 1.

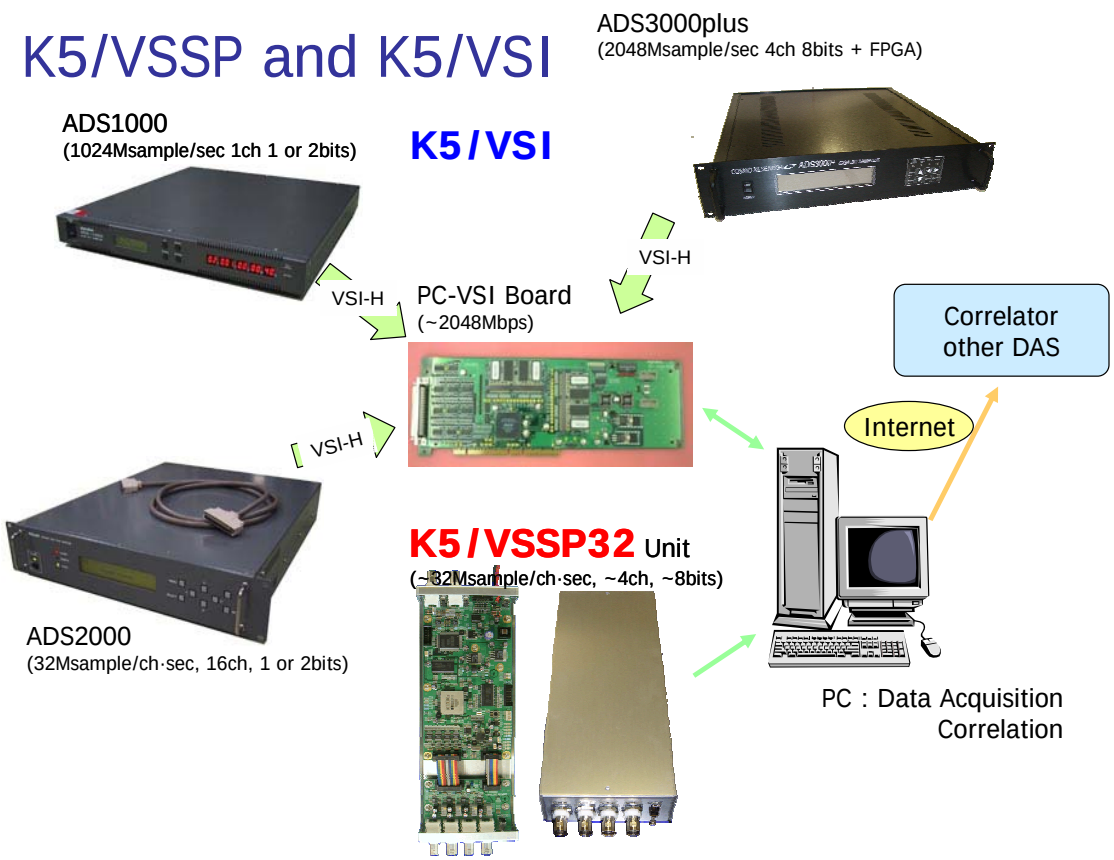

Fig. 1 Concept of the entire K5 System.

As shown in the Fig. 1, there are two sorts of VLBI systems. One is a series of DAS system s using combination of ADS sampler device s with VSI-H interface and PC-VSI card as interface for PC to capture VLBI observation data. This system is called K5/VSI system. Another one is DAS system using IP-VLBI sampler unit with 4 channels of data sampler per one unit. This system is called K5/VSSP or K5/VSSP32 systems. These two types of DAS system have their own software correlation software. Though the data format of the two types of DAS is different at present, new standard VLBI data format (VDIF[1]) will be supported for both DASs.

1.1 K5/VSI Data Acquisition System

K5/VSI is the name for DAS system with VSI (VLBI Standard Interface) as hardware data interface between sampler and PC. NICT has developed three kinds of VLBI data sampler system named ADS-1000, ADS-2000, and ADS3000/ADS3000plus. All of these sampler systems have VSI-H interface [2] as output. These samplers are used by different observation modes for their purposes.

Overview of the specification of the samplers is listed in table 1.

Table 1. Specifications of the each VSI samplers.

	ADS1000	ADS2000	ADS3000	ADS3000Plus
Sampling Speed	1024Msps	64Msps	2048Msps	～4 Gbps
Sampling Bits	1 bit or 2 bits	1 bit or 2 bits	8 bits	2/4/8 bit
No. of Input	1	16	1	2
No. Channels	1	16	Programmable	Programmable
Max. Data Rate	2048Mbps	2048Mbps	4096Mbps	8192Mbps
Interface	VSI-H (2 ports)	VSI-H (2 ports)	VSI-H (2 ports)	VSI-H (4 ports)

ADS1000

ADS2000

ADS3000

ADS3000Plus

ADS-1000 has one channel input with 1024MHz sampling. This system is mainly used for domestic astronomical observation to get wide frequency band to get higher sensitivity. This system has been employed for VERA project of NAOJ(National Astronomical Observatory of Japan) in conjunction with digital filter backend, where 2 narrower frequency channels are extracted. The ADS1000 is also employed in Korean VLBI network (KVN) observation system.

ADS2000 has 16 video signal inputs with 32MHz sampling each. Since there are some

trouble in 64MHz sampling mode, where 2048Mbps (64Msps/2bit/16ch) sampling is designed, constant 32MHz sampling is used at present. Jointly using anti-aliasing analog video filter in front of sampler and down sampling of data after the sampler, variety of observation modes is supported. Some of typical observation modes are 32Msps/2bit/16ch (=1024Mbps), 32Msps/2bit/8ch (=512Mbps), and 16Msps/2bit/8ch (=256Mbps). Mark5B emulator has been developed by using ADS-2000 and joint international eVLBI observations are now available.

ADS3000/ADS3000plus is a new generation VLBI sampler with digital base-band conversion (DBBC) function. FPGA logic ICs are used for implementation of the data processing algorithm and it can be easily modified by re-loading the hardware program of FPGA. Conventional geodetic VLBI observation mode with 16 frequency channels can be realized by the DBBC function of ADS3000plus.

VSI-board is the common interface board for PC by using PCI-X bus. Since this board is VSI-H compliant, any other samplers can be connected used. In fact Makr5B sampler is compliant to the VSI-H specification, thus it could be connected to the PC-VSI card and used for VLBI experiment with Kashima. High Speed software correlator for K5/VSI named 'GICO-3' has been developed by M.Kimura[3]. And the GICO-3 software correlator is being implemented for backup software correlator for VERA project of NAOJ.

1.2 K5/VSSP32 Data Acquisition System

K5/VSSP32 is another VLBI sampler. Its other name is IP-VLBI sampler. It is designed with 4 video signal inputs per one unit. Its first version is called K5/VSSP and the second version is named K5/VSSP32[4] (Fig. 1).

Fig. 2. K5/VSSP(left) and K5/VSSP32(right) sampler boards.

Table 2. Comparison of the old (K5/VSSP) and new(K5/VSSP32) samplers.

Sampling Freq. (MHz)	0.04, 0.1, 0.2, 0.5, 1, 2, 4, 8, 16	0.04, 0.1, 0.2, 0.5, 1, 2, 4, 8, 16, 32, 64
Built-in digital LPF(MHz)	---	2, 4, 8, 16, through
Analog Input range	-1V - +1V	-1V - +1V
Analog Bandwidth of A/D	100MHz	300MHz
AD resolution bit	1,2,4,8	1,2,4,8
No. of Channels per unit	1,4	1,4
Maximum data rate per unit	64Mbps	256Mbps
DC offset adjustment	NA	Available from host PC
Reference signal input	1PPS, 10MHz	1PPS, 10MHz or 5MHz

Data Interface	PCI bus	USB 2.0

The Comparison of characteristics between two samplers is indicated in Table. 2. Now only K5/VSSP32 is regularly used for geodetic VLBI observation of IVS project. VSSP is an acronym of the Versatile Scientific Sampling Processor. This name is used because the system is designed to be used for general scientific measurements. The sampler has variety of sampling rate and quantization bits (Table 2). Combination use of multiple K5/VSSP32 samplers enables variety of observation modes for versatile purposes. Sampled data is stored on file system of each PC as standard binary file on that operating system. When 16 channels of observation are performed, four set of VLBI data is generated in parallel and stored for each scans. The data is stored in frame format, where 32 byte header part and following data part containing 1 sec of data. And that frame for 1 sec is repeated every seconds. The structure of the header part is indicated in Fig. 3.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits : 0-63)						total day (9bits)								
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06	LPF frequency (MHz: 0 means through)								AUX FIELD format # (1)							
0x07	station ID (max 2 charcters)															
0x08	station name (max 8 characters)															
0x09																
0x0A																
0x0B																
0x0C	PC host name (max 8 characters)															
0x0D																
0x0E																
0x0F																

eflg: set when error occurred in a previous frame

Fig. 3 Header part of K5/VSSP32 data

More detailed documentation on K5/VSSP is available on the web[5]. Observation software and source code is available from the web page.

Fig.4 is one of the combined set of the K5 system with 4 units of K5/VSSP samplers are equipped. The system has a capability to sample analog data stream by using the external frequency standard signal and the precise information of the sampled timing. The system is also used to process the sampled data. For geodetic VLBI observations, software correlation program runs on the K5/VSSP system. Therefore, it can be said that the functions of the formatter, the data recorder, and the correlator are combined into the single system. It is consist of four Linux PC systems. Each Linux PC system has one K5/VSSP sampler. The total recording speed with 4 units of sampler is 1024 Mbps.

Fig. 4 Picture of the K5/VSSP system.

To process the data sampled with the K5 data acquisition system, software correlation processing program has been developed on conventional PC systems. The correlation processing program shares the data via Network File System (NFS) over local area network (LAN), then it computes cross correlation functions without any specially designed hardware. Since easily re-writable software programs and general PC systems are used, the processing capacity and the function of the correlator can be easily expanded and upgraded.

2. Example of Observation Operation with K5/VSSP32

Since the most geodetic VLBI observations including IVS sessions are performed with K5/VSSP32 system at present, some example of observation operation procedure with K5/VSSP32 is explained here.

A) Capturing data following to the schedule file.

Software tool ‘autoobs’ is used for this purpose. Generally, just typing the command name will show simple usage of that command. Usually ‘autoobs’ is used as follows:

autoobs –‘config file’

The “config file” contains ‘path to the observation schedule file’, ‘channel group of recording’, ‘station ID’, ‘entries of paths of the directory to store the data’, ‘Observation mode’, ‘output file naming convention type’. Example of the ‘config file’ is indicated in Appnedix-B.

B) Capturing data for a certain length of period, manually

sampling <sfreq>[:lpf] adbit[:bitshift] numch [filename [logfile]]
ex.) sampling 30 4 2 4 test.dat
This means 4MHz/2bit/4ch sampling for 30 sec will be recorded on test.dat file.
More explanation on option is available by just typing ‘sampling’.

C) Checking recorded data
‘datachk’ command is used for checking K5/VSSP data
datachk datafile [mode [logfile [errlog [keepmode]]]]
ex.) datachk k5test.dat 1
will shows the time-code, sampling mode and statistics for every seconds.
More explanation on option is available by just typing ‘datachk’.

‘m5check’ command is used for checking Mark5A data.
m5check m5file [mode]
ex.) m5check mark5data.dat
More explanation on option is available by just typing ‘m5check’.

D) Conversion from Mark5A data to K5/VSSP data
Command ‘m5tok5’ is used for this purpose.
ex.) m5tok5 mark5_fle.dat –g 1 -i m5tok5info256.txt –o k5file_grp1.dat
This command converts ‘mark5_file.dat’ to ‘k5file_grp1.dat’ which will contain data of channel 1-4. The information file ‘m5tok5info256.dat’ is necessary to describe the mark5 data information. Example of this file is at Appendix C.

E) Conversion from K5/VSSP data to Mark5A data
Command ‘k5tom5’ is used for this purpose.
ex.) k5tok5 2009114 k5file1 k5file2 k5file3 k5file4 -i k5tom5info256.txt –o mark5file.dat
This command converts a set of k5 files to ‘mark5file.dat’ which contains data of 16 channels. The information file ‘k5tom5info256.dat’ is necessary to describe the mark5 data configuration. Example of this file is at Appendix D.

Software library and documentation to operate the K5/VSSP(32) is developed and maintained mainly by T.Kondo. That software library includes (1) Observation with K5/VSSP(32), (2) Data conversion between K5/VSSP(32) ⇔ Mark5A. (3) Miscellaneous tools for checking the data, checking the schedule file, computing spectrum from the data, extracting one channel data from a data set containing 4 ch data, merging data from multiple one channel data sets to 4 channel data set, cutting out a chunk of data from long observation, and so on. The names of K5/VSSP tools are listed in the table of Appendix 1. These software tools and documentation is available from web page [5].

3. Future Direction

NICT as a technology development center of IVS has been developed two types of VLBI DAS system. They have been developed from different motivation, one is for wide band high speed sampling observation, and another one is for multi channel versatile purpose data sampler

utilizing IP network for data transfer and processing. Although, their data format is different at present, we are going to unify the data format with VDIF format or enable interchanging with other data formats.
In near future, when digital base band converter function of ADS3000plus will be completed, both high speed wide band data sampling and multi channel observation will become available with single sampling unit.

Appendix A. List of utilities used with the K5/VSSP system

command	function
pctimeset	Adjust the time of PC system.
signalcheck	Check the presence of external 1PPS and 10MHz signals.
timedisp	Display the time set on the board
timesettk	Set the time on the board and synchronize the clock to the external 1PPS signal.
timesinc	Synchronize the clock on the board.
monit	Display histogram of the input signal(s).
skdchk	Read the schedule file and calculates necessary disk storage.
autoobs	Start data acquisition according to the schedule file.
sampling	Record single scan of data and generate a recorded data file.
datachk	Read the recorded data file and check whether any data were lost within each data segment.
extdata	Read the recorded data file and extract the data in the ASCII format.
datacut	Split the recorded data file.
four2one	Generate one channel data file from the recorded data file with four channels of input.
speana	Calculates power spectrum and display the results from recorded data file.
adbitconv	Change the sampling bits in the recorded data file.
one2four	Combine four recorded data files with one channel of input for each file, and generates a data file with four channels of input.
data_half	Convert the sampling data rate half of the original file.
data_double	Convert the sampling data rate double of the original file.

Appnedix-B Example of config file for ‘autoobs’

```
** Sample K5 run control file Ver 3.6
**
$SKED
/home/vlbi/schedule/u8193f.skd                  * schedule file    VEX file is allowed
$STATION_ID
O                               * station ID.         1 chars for SKED and 2 chars for VEX
$LOGDIR
/home/vlbi/ipvlbi/log     * log directory
$OUTDIR                     * up to 10 entries
/k55a/ad5/u8193f/kas34       * 1st out put directory candidate
/k55a/ad6/u8193f/kas34       * 2nd out put directory candidate
$SAMPLE
span=0       * obs span (sec), 0 means as schduled
sfreq=32     * sampling frequency   40,100,200,500 (for kHz)   1,2,4,8,16 (for MHz)
adbit=1     * A/D bits   1,2,4,8
numch=4     * # of channels     1,4
$NAMING_TYPE
*     out file naming type selection
*1 ** Type I          XDDDNNNN.dat
*                   where X       -- satation id (1 char)
*                             DDD    -- total day at 1st scan (3 digits)
*                             NNNN -- obs number    (4digits)
*-1 ** Type –I       XDDDNNNN.#ch.dat
*                   where #ch    -- number of channels in data
2   ** Type II       sidDDDHHMMSSG.dat
*                   where sid      -- station id    (1 char or 2 char)
*                             DDD      -- total day at current scan (3digits)
*                             HH       -- hour at the start of scan (2digits)
*                             MM       -- minute at the start of scan (2digits)
*                             SS       -- second at the start of scan (2 digits)
*                             G         -- frequency group id (a|b|c|d) or null
*-2   ** Type –II   sidDDDHHMMSSG.#ch.dat
*                   where #ch    -- number of channels in data
*3 ** Type III   (compliant with e-VLBI file-naming conventions)
*          expid_sid[G]_scanid_YYYYDDDDHHMMSS.k5
*                   where expid    -- experiment code
*                             sid       -- station ID (2 lower-case characters)
*                             G           -- PC id (1|2|3|4)
*                             scanid -- scan id
*                             YYYY      -- year (4digits)
*                             DDD       -- total day at current scan (3digits)
*                             HH         -- hour at the start of scan (2digits)
*                             MM         -- minute at the start of scan (2digits)
*                             SS         -- second at the start of scan (2 digits)
$FREQ_G
*     set frequency group used in type II naming rule
*        or PC id used in type III naming rule
*     if omitted    null character is used, i.e., file name
*     will be sidDDDHHMMSS.dat
*          1,2,3,4    or a,b,c,d is possible
*          a,b,c,d is automaticaly converted to 1,2,3,4 in type III naming rule
1   **    means 'a'
*a     **    also OK for 'a'

$SUBNET
*     subnet mode selection    on | off    (default on)
on

$FILE_SIZE_LIMIT
*     file size limitation      on | off    (default on)
*     if set to "on", big file is divided into 2GB each.
*     if set to "off", no limitation on 1 file size.
off
```

Appnedix-C Example of Information file for ‘m5tok5’

```
*** mk5tok5 information file created by m5tok5 (Ver 2.03  2005-01-13)
***       on Sun Jun 11 11:27:25 2006
***   (head stack number included in track info)
***   analyzed VEX file     : ./k06161.vex
***   analyzed Mark-5 file : ./k06161_0059+581_161-0740
***   station : WETTZELL     (Wz)
***   mode (for scan # 1)  : GEOSX4F-4F
***
$CHANNEL;  * channel-track info block
   adbit  = 1 ;  *   A/D resolution
   sample = 16000000.000000 ;  *   Sampling frequency
   fanout = 2 ;  *   Fanout
** default pass = A
**
**      nn    => channel #
**      h-ss => h: head stack #, ss: sign bit track #
**      h-mm => h: head stack #, mm: magnitude bit track #
** ch = nn : h-ss : h-ss
   ch = 01 : 1-02 : 1-04 ;
   ch = 02 : 1-10 : 1-12 ;
   ch = 03 : 1-14 : 1-16 ;
   ch = 04 : 1-18 : 1-20 ;
   ch = 05 : 1-22 : 1-24 ;
   ch = 06 : 1-26 : 1-28 ;
   ch = 07 : 1-30 : 1-32 ;
   ch = 08 : 1-03 : 1-05 ;
   ch = 09 : 1-11 : 1-13 ;
   ch = 10 : 1-15 : 1-17 ;
   ch = 11 : 1-19 : 1-21 ;
   ch = 12 : 1-23 : 1-25 ;
   ch = 13 : 1-27 : 1-29 ;
   ch = 14 : 1-31 : 1-33 ;
   ch = 15 : 1-06 : 1-08 ;
   ch = 16 : 1-07 : 1-09 ;
$DATAMODE;   * Mark-V data format
   parity = 0 ;  * non-parity
   nrzm   = 0 ;  * NRZL encoding
   format = Mark-IV ; * Mark-III or IV format
   ntrack = 32 ; * # of tracks (bits/word)
```

```
$BITPOS;      * bit position versus track information
**
**      bb    => bit position #
**      h-tt => h: head stack #, tt: track #
** bitpos = bb : h-tt
   bitpos = 00 : 1-02 ;
   bitpos = 01 : 1-03 ;
   bitpos = 02 : 1-04 ;
   bitpos = 03 : 1-05 ;
   bitpos = 04 : 1-06 ;
   bitpos = 05 : 1-07 ;
   bitpos = 06 : 1-08 ;
   bitpos = 07 : 1-09 ;;
   bitpos = 08 : 1-10
   bitpos = 08 : 1-10 ;
   bitpos = 09 : 1-11 ;
   bitpos = 10 : 1-12 ;
   bitpos = 11 : 1-13 ;
   bitpos = 12 : 1-14 ;
   bitpos = 13 : 1-15 ;
   bitpos = 14 : 1-16 ;
   bitpos = 15 : 1-17 ;
   bitpos = 16 : 1-18 ;
   bitpos = 17 : 1-19 ;
   bitpos = 18 : 1-20 ;
   bitpos = 19 : 1-21 ;
   bitpos = 20 : 1-22 ;
   bitpos = 21 : 1-23 ;
   bitpos = 22 : 1-24 ;
   bitpos = 23 : 1-25 ;
   bitpos = 24 : 1-26 ;
   bitpos = 25 : 1-27 ;
   bitpos = 26 : 1-28 ;
   bitpos = 27 : 1-29 ;
   bitpos = 28 : 1-30 ;
   bitpos = 29 : 1-31 ;
   bitpos = 30 : 1-32 ;
   bitpos = 31 : 1-33 ;
$GROUP;   * group # versus channel # table
*****************************************
**Please edit this table as you like     **
*****************************************
**      g    => group #
**      ch1 => 1st channel # in this group
**      ch2 => 2nd channel # in this group
**      ch3 => 3rd channel # in this group
**      ch4 => 4th channel # in this group
** group = g : ch1 : ch2 : ch3 : ch4 ;
   group = 1 :  1 :  2 :  3 :  4 ;
   group = 2 :  5 :  6 :  7 :  8 ;
   group = 3 :  9 : 10 : 11 : 12 ;
   group = 4 : 13 : 14 : 15 : 16 ;
```

d

Appnedix-D Example of Info file for 'k5tom5'

```
*** k5tom5 information file created by k5tom5 (Ver 1.32  2008-02-19)
***       on Thu May 08 09:08:03 2008
***   (head stack number included in track info)
***   analyzed VEX file     : d:/temp/rd0803.vex
***   station : TSUKUB32     (Ts)
***   mode (for scan # 1)   : GEOSX8N-8F
***
$CHANNEL;  * channel-track info block
   adbit   = 2 ;  *   A/D resolution
   sample = 16000000.000000 ;  *   Sampling frequency
   fanout = 1 ;  *   Fanout
** default pass = A
**
**      nn    => channel #
**      h-ss => h: head stack #, ss: sign bit track #
**      h-mm => h: head stack #, mm: magnitude bit track #
**      bbc# => BBC#
**      sb    => sideband   L(SB) or U(SB)
** ch = nn : h-ss : h-mm : bbc# : sb
   ch = 01 : 1-02 : 1-04 :    1 : U ;
   ch = 02 : 1-06 : 1-08 :    1 : L ;
   ch = 03 : 1-10 : 1-12 :    2 : U ;
   ch = 04 : 1-14 : 1-16 :    3 : U ;
   ch = 05 : 1-18 : 1-20 :    4 : U ;
   ch = 06 : 1-22 : 1-24 :    5 : U ;
   ch = 07 : 1-26 : 1-28 :    6 : U ;
   ch = 08 : 1-30 : 1-32 :    7 : U ;
   ch = 09 : 1-03 : 1-05 :    8 : U ;
   ch = 10 : 1-07 : 1-09 :    8 : L ;
   ch = 11 : 1-11 : 1-13 :    9 : U ;
   ch = 12 : 1-15 : 1-17 :   10 : U ;
   ch = 13 : 1-19 : 1-21 :   11 : U ;
   ch = 14 : 1-23 : 1-25 :   12 : U ;
   ch = 15 : 1-27 : 1-29 :   13 : U ;
   ch = 16 : 1-31 : 1-33 :   14 : U ;
**
$DATAMODE;   * Mark-V data format
*********************************************
**      Please edit this table as you like     **
*********************************************
**
   parity = 1 ;  * non-parity
   nrzm    = 1 ;  * NRZL encoding
   format = VLBA ; * Mark-IV format
   ntrack = 32 ; * # of tracks (bits/word)
   modulation = ON
```


```
$DATAMODE;   * Mark-V data format
*********************************************
**      Please edit this table as you like     **
*********************************************
**
   parity = 1 ;  * non-parity
   nrzm    = 1 ;  * NRZL encoding
   format = VLBA ; * Mark-IV format
   ntrack = 32 ; * # of tracks (bits/word)
   modulation = ON
```


```
$BITPOS;
** bit position versus track information
*************************************
**   Please edit this table as you like     **
*************************************
**
**      bb    => bit position #
**      h-tt => h: head stack #, tt: track #
** bitpos = bb : h-tt
   bitpos = 00 : 1-02 ;
   bitpos = 01 : 1-03 ;
   bitpos = 02 : 1-04 ;
   bitpos = 03 : 1-05 ;
   bitpos = 04 : 1-06 ;
   bitpos = 05 : 1-07 ;
   bitpos = 06 : 1-08 ;
   bitpos = 07 : 1-09 ;
   bitpos = 08 : 1-10 ;
   bitpos = 09 : 1-11 ;
   bitpos = 10 : 1-12 ;
   bitpos = 11 : 1-13 ;
   bitpos = 12 : 1-14 ;
   bitpos = 13 : 1-15 ;
   bitpos = 14 : 1-16 ;
   bitpos = 15 : 1-17 ;
   bitpos = 16 : 1-18 ;
   bitpos = 17 : 1-19 ;
   bitpos = 18 : 1-20 ;
   bitpos = 19 : 1-21 ;
   bitpos = 20 : 1-22 ;
   bitpos = 21 : 1-23 ;
   bitpos = 22 : 1-24 ;
   bitpos = 23 : 1-25 ;
   bitpos = 24 : 1-26 ;
   bitpos = 25 : 1-27 ;
   bitpos = 26 : 1-28 ;
   bitpos = 27 : 1-29 ;
   bitpos = 28 : 1-30 ;
   bitpos = 29 : 1-31 ;
   bitpos = 30 : 1-32 ;
   bitpos = 31 : 1-33 ;
**
```

References

[1] Whitney, A., VLBI Standard Hardware Interface Specification -- VSI-H, 2002. http://www.haystack.mit.edu/tech/vlbi/vsi/docs/2002_12_12_vsi-h_draft_rev_1.1.pdf

[2] VLBI Data Interchange Format (VDIF) specification,

[3] Kimura, M., J. Nakajima, H. Takeuchi, T. Kondo, High Performance PC Based Gigabit VLBI System, IVS-NICT TDC News No. 25, pp.64-66, 2004. http://www2.nict.go.jp/w/w114/stsi/ivstdc/news_25/pdf/tdcnews_25.pdf

[4] Kondo, T., Y. Koyama, H. Takeuchi, and M. Kimura, Development of a new VLBI sampler unit (K5/VSSP32) equipped with a USB 2.0 interface, *IVS 2006 General Meeting*, Jan. 2006.

[5] http://www2.nict.go.jp/w/w114/stsi/K5/VSSP/index-e.html