LO/ IF /Frequency Handling

R. Prestage

19th October 2000

Contents

1 Introduction

The purpose of this document is to provide reference information for how to configure the IF/LO system prior to observing, and how the frequency and polarization information is handled during and after observations. The intention is that this whole document will become part of the "GBT Observing System Specification" document being editted by Dana Balser. I have kept it separate for now so that others can comment on these sections in isolation.

NOTE: this is not intended to be a user document as such. It is intended to be a reference document describing precisely how things are/should be done. For example a glish procedure or the aips++ filler or whatever will probably perform most of the steps described here, and the user will not need to know the details.

2 IF/LO setup

The GBT LO/IF system is described in detail in:

http://www.gb.nrao.edu/GBT/EL/gbt_if.html.

The intention of this section is *not* to replicate that document, but to abstract out those sections related to configuration. For example, I would expect there to be sections on:

1. how to configure the IF router
2. how to choose LO1, and optionally LO2 settings
3. how to ensure the selected pass band matches the input pass band of the backend
4. etc.

And it would include tables of pass bands, centre frequencies, etc. If anyone believes this is *NOT* necessary, or such documentation exists elsewhere, please let me know before I start writing!

3 Collating Feed and Frequency Information - Background

Once the IF/LO system has been configured as described in Section Two above, the appropriate RF pass-band will be sampled by the backend, and either a continuum signal (DCR) or spectrum (Spectometer, Spectral Processor) will be obtained. This and the next section describe precisely how the frequency values associated with these data are calculated, and how the backend data are tagged with the correct receiver (feed, polarization) information.

Frequency related information is stored during a scan in four separate FITS files:

- the User Interface FITS file, which contains setup information for the scan (I haven't looked at this yet)
- the LO1 FITS file, which stores the values of the first LO during the scan
- the IF Manager FITS file, which describes the IF configuration, pass-bands and so on and,
- the backend FITS file which for the spectrometers contains information on the backend resolution elements.

These FITS files are described in detail in Appendices B through F

The IF Manager performs has a key role in tracking frequency and feed information. In particular, it:

- keeps track of which backend inputs have been selected.
- keeps track of the IF routing from frontend outputs to backend inputs
- calculates the effective center IF frequency and IF pass band as a result of the LO conversions and filters in the IF chain
- generates the coefficients to the "sky frequency formula", which allows the corresponding RF frequency to any backend input frequency to be calculated.

All of the above information is stored in the IF Manager FITS files on a per-scan basis. The format of the IF Manager fits file is described in Appendix B. The IF Manager creates one row in the binary table extension for each active IF CHANNEL (as described below). Note that the IF Manager will store information for channels which have been selected for a backend, even if that backend has not been selected to participate in a scan.

The combination of this information into the aips++ Measurement Set is conceptually a two-stage process. In the first stage, the information in the FITS files is combined to generate a topocentric sky frequency for each backend resolution element. In the second stage, this frequency information is stored in the aips++ Measurement Set in a way which is convenient for subsequent processing.

The topocentric sky frequency for any given frequency input to the backend is given by:

$$F_{sky} = SFF_{sideband} \times F_{backend} + SFF_{multiplier} \times LO1 + SFF_{offset} \tag{1}$$

where the "sky frequency formula" values $SFF_{sideband}$, $SFF_{multipler}$ and SFF_{offset} come from the IF manager FITS files, and represent the combined effects of the intermediate IF up/down conversion. LO1 is taken from the $LO1$ FITS file, and $F_{backend}$ is calculated as appropriate for each backend resolution element.

3.1 Nomenclature

For each backend, the IF Manager divides the "input ports" into BANK and CHANNELS. For example, the DCR has two input BANKs, A and B, each of which has 16 possible input CHANNELS. The division of "input ports" into BANKs reflects the reality of the hardware, but is essentially irrelevant to the remainder of the discussion. In the case where the GBT Spectrometer is configured to use more than one BANK simultaneously, the data from each BANK is stored to a separate FITS file.

For both the Spectral Processor and the GBT Spectrometer, a single CHANNEL is split into a number of "resolution elements". These are also often referred to as "channels". I will use the terminology "IF CHANNEL" or "backend channel" to distinguish between the two. From the point of view of aips++, each IF CHANNEL maps on to one SPECTRAL_WINDOW.

A CHANNEL refers to one IF path from a receiver to a backend. It is possible for the IF output from one polarization of one beam of a receiver to be split and fed into more than one backend input. In this case, each IF path is considered a separate CHANNEL.

3.1.1 Scans, Integrations and Phases

For the purposes of this discussion, a **SCAN** can be defined as a period of time with a well defined starting point and duration, during which all participating devices step through a well-defined sequence of phases. Each SCAN results in a separate FITS file for each device, and the names of these files are stored in the "Scan Log" FITS file for that scan. A typical SCAN might last from a few seconds to tens of minutes.

Within the backend, a SCAN may be broken down into a set of INTEGRATIONS. For example, a sixty second SCAN might consist of 10 six-second integrations. Each INTEGRATION consists of an integral multiple of PERIODs. Each PERIOD is composed of a number of PHASES, and the details of each PHASE is stored in the STATE binary tables extension in the backend FITS file. If an INTEGRATION consists of more than one PERIOD, the data for the same PHASE of successive PERIODs are combined together in the backend Manager before being written to the FITS file.

4 Collating Feed and Frequency Information - Details

In order to collate all of the above information, the process is as follows. First, the desired backend is chosen for processing. The name of the backend, and the backend channel of interest are used as the key to the appropriate row in the IF Manager binary table. From that row, the frontend (feed, polarization) and IF frequency information may be obtained. These are then combined with the appropriate value of LO1 to generate a topocentric sky frequency for each backend resolution element. Finally, these are written to the aips++ Measurement Set. The following sections describe this in more detail, without specifying which value of LO1 from the LO1 FITS file to use. This question is then addressed in Section N.M.

4.1 DCR

As noted above, the DCR has two BANKS, A and B. Each BANK has 16 possible input ports (1 through 16), referred to as "Channels" in the CLEO Interface, and labelled CHANNELID in the FITS file.

Using the DCR values BANK and CHANNELID, find the IF Manager ROW with corresponding values for DCR BANK and CHANNEL. This selects the appropriate row in the IF Manager. Extract the appropriate values of $SFF_{sideband}$, $SFF_{multipler}$ and SFF_{offset} from this row.

Extract the appropriate value of LO1 from the LO1 FITS file.

The DCR performs no further frequency conversions, pass-band filtering or spectral dispersion. Therefore, there is only a single backend resolution element. The centre frequency of this element is given by the value of center_IF in the IF FITS file. Use this for the value of $F_{backend}$, and apply the sky frequency formula. This provides the topocentric sky frequency for use in the aips++ Measurement Set. Use the IF Manager FITS file value "bandwidth" to provide the width of this element.

4.2 Spectral Processor

The Spectral Processor has two banks, A and B, which may be used individually, but are usually combined. The bank label is stored in the Primary Header keyword INSTRUME, and may take the values SPA, SPB or SPAB. Depending upon the choice of banks, the Spectral Processor may have 1, 2, 4 or 8 channels. This number is stored in the STATE binary table keyword RCVRS, and the actual values are stored in the RECEIVER binary table column labelled RCVRID.

Using the Spectral Processor values INSTRUME and RCVRID, find the IF Manager ROW with corresponding values for SpectralProcessor BANK and CHANNEL. This selects the appropriate row in the IF Manager. Extract the appropriate values of $SFF_{sideband}$, $SFF_{multipler}$ and SFF_{offset} from this row.

Extract the appropriate value of LO1 from the LO1 FITS file.

The SpectralProcessor may perform further IF conversions, but all of these are included in the IF Manager SFF values. Each IF CHANNEL is divided into N resolution elements (backend channels). I cannot find this value stored in the FITS file; it may be calculated by dividing BANDWD by FREQRES, or implicitly from the size of the DATA array.

For a DATA array with N frequency elements, counting from 1, the value of "center_IF" in the IF FITS file corresponds to the centre of the channel given by (N/2)+1. This corresponds to $F_{backend}$ for that channel. The backend channel spacing is given by the SpectralProcessor FITS file FREQRES quantity in the RECEIVER binary table. This allows the center IF frequency of the remaining channels to be calculated. Using each of these values for $F_{backend}$, the value of LO1 and the sky frequency formula allows the topocentric sky frequency for each backend channel to be calculated.

4.3 Spectrometer

The spectrometer has four BANKS: A,B,C,D plus PULSAR. Data from each BANK is written to a separate FITS file. Pulsar processing is not yet implemented, and will be discussed separately.

The following discussion is tentative, may be incorrect, and applies to cross-correlation only!

The Spectrometer has 40 SAMPLERs, each of which may accept an IF input. These are labelled SAMPLERA and SAMPLERB to allow for cross-correlation; in auto-correlation mode they will always be the same.

Using the value of BANK and SAMPLERA, find the IF Manager ROW with corresponding values for Spectrometer BANK and CHANNEL. This selects the appropriate row in the IF Manager. Extract the appropriate values of $SFF_{sideband}$, $SFF_{multipler}$ and SFF_{offset} from this row.

Extract the appropriate value of LO1 from the LO1 FITS file.

In the case of the Spectrometer, the various LO and IF filter setting ensure that the RF pass-band is correctly matched to the Spectrometer input passband, but the Spectrometer samplers determine the precise values of $F_{backend}$ which are sampled. The value for $F_{backend}$ for some reference channel, and the channel spacing is derived from the Spectrometer FITS file - details still to be clarified.

Using $F_{backend}$ for the reference channel, the channel number, the value of LO1 and the sky frequency formula allows the topocentric sky frequency for each backend channel to be calculated.

4.4 Holography Backend

To be written

4.5 Handling LO1

This section describes how to determine which value of LO1 to apply to each individual integration in a backend FITS file. The values of LO1 are stored in a binary table in the LO1 FITS file with columns DMJD, frequency, phase_number and sig_ref_state.

For each backend, it is necessary to determine how many times the switching PERIOD is repeated per INTEGRATION (I haven't sorted all of this out yet.) Given this, it will be possible to select from the LO1 FITS file all values of LO1 in a given phase which correspond to the same INTEGRATION in the backend data file. If Doppler tracking has not been selected, these should be identical for a given phase, and the single value for each phase can be used for each phase in the INTEGRATION.

Doppler tracking is covered in the next section.

5 Storing topocentric sky frequencies in the aips++ Measurement Set

A Nomenclature

This section describes how the various receiver feeds and backend input ports are labelled, and how this information is stored in the IF Manager FITS file.

A.1 Receivers

In GBT nomenclature, a **receiver** is the physical instrument mounted at prime or Gregorian focus. Each receiver may have one or more **beams** on the sky, and each beam has dual **polarizations**. For some receivers, the beams may be switched between linear and circular polarization.

Polarizations are labelled as L,R,X,Y and U for left and right-hand circular polarization, the two linear polarizations, and U if the polarization is unknown. Beams are labelled numerically. For example the Ku band receiver has two beams, each of which generate dual circular polarization. Hence the four output IFs are labelled L1, L2, R1, R2.

In the aips++ Measurement Set terminology, *a FEED is a collecting element on an antenna, such as a single horn, that shares joint physical properties and makes sense to calibrate as a single entity. It is an abstraction of a generic antenna feed, and is considered to have one or more RECEPTORs that respond to different polarization states.... Feeds are numbered from 0 on each separate antenna for each SPECTRAL_WINDOW_ID. Consequently, FEED_ID should be non-zero only in the case of feed arrays, i.e. multiple, simultaneous beams on the sky at the same frequency and polarization.*

For the above example, the mapping would be as follows:

GBT label	MS2 FEED_ID	MS2 RECEPTOR	MS2 POLARIZATION_TYPE
L1	0	0	L
R1	0	1	R
L2	1	0	L
R2	1	1	R

A.2 Backends

Each backend has a number of “IF input ports”, which are connected to one polarization of one beam of the (same) receiver. The input ports of each backend may be separated into a number of “banks”. The GBT backends all produce one FITS file per bank.

In the IF Manager FITS file, these two quantities are referred to BANK as CHANNEL. The use of CHANNEL is potentially confusing, since “channel” may also be used to refer to spectometer resolution elements. Hopefully however the use of the term CHANNEL is obvious from the context.

The IF Manager FITS file Binary Table Extension contains one row for every CHANNEL that has been selected (whether or not that backend has been selected for the scan), and provides the receiver and backend information associated with this channel.

The backend FITS files use a variety of terms for the quantities BANK and CHANNEL, as summarised in the table below:

NB: need to check indexing

IF Manager	BANK	CHANNEL
DCR Label	INPBNK	CHANNELID
DCR Values	A, B	1 - 16
SpectralProcessor Label	BACKEND	RCVRID
SpectralProcessor Values	SPAB ??	1 - 3
GBT Spectrometer Label	BANK	SAMPLERA, SAMPLERB
GBT Spectrometer Values	A,B,C,D, PULSAR	1 - 40

B LO1

```
LO1.fits - One Primary Header Unit, one Binary Tables Extension

Main Header:

-------------------------------------------------------------------------------
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT   Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT   Contact the NASA Science Office of Standards and Technology for the
COMMENT   FITS Definition document #100 and other FITS information.
INSTRUME= 'LO1_LO1A'           / instrument
DEVICE  = 'LO1     '           / Telescope's device
MANAGER = 'LO1A    '           / Device's manager
SAMPLER = 'phaseState'         / Manager's sampler
COMMENT   Phase and Frequency
DELTA   =         0.000000E+00 / Minimum time between writing samples in seconds
UTSTART = 5.18066537363426E+04 / DMJD when started writing FITS file
STRTTIME= '51806 15:41:22'     / expected start time
SCAN    = '2632    '           / integer scan identifier
PROJID  = 'MHC_IFLO'           / string project identifier
SOURCE  = 'test    '           / string source identifier
SCANID  = 'test    '           / string scan identifier
COMMENT   Green Bank Telescope
END
-------------------------------------------------------------------------------

PhaseState Binary Table Extension Header:
-------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   24 / width of table in bytes
NAXIS2  =                   21 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
```

```
TFIELDS =                    4 / number of fields in each row
TTYPE1  = 'DMJD    '           / label for field   1
TFORM1  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT1  = 'd       '           / physical unit of field
TTYPE2  = 'frequency'          / label for field   2
TFORM2  = 'D       '           / data format of field: 8-byte DOUBLE
TUNIT2  = 'MegaHertz'          / physical unit of field
TTYPE3  = 'phase_number'       / label for field   3
TFORM3  = 'J       '           / data format of field: 4-byte INTEGER
TUNIT3  = 'None    '           / physical unit of field
TTYPE4  = 'sig_ref_state'      / label for field   4
TFORM4  = 'J       '           / data format of field: 4-byte INTEGER
TUNIT4  = 'None    '           / physical unit of field
EXTNAME = 'phaseState'         / name of this binary table extension
COMMENT   DMJD: time sample taken;
COMMENT   frequency: The estimated frequency as commanded to the Synthesizer dev
COMMENT   ice;
COMMENT   phase_number: The phase number from 1..N;
COMMENT   sig_ref_state: State of the sig/ref phase for each switching phase; Si
COMMENT   gRefState (  Sig, Ref )
END
---------------------------------------------------------------------------
```

ROW	DMJD	frequency	phase_number	sig_ref_state
1	5.1806653737E+04	1.1000E+10	1	0
2	5.1806653742E+04	1.1000E+10	2	1
3	5.1806653748E+04	1.1000E+10	3	0
4	5.1806653754E+04	1.1000E+10	4	1
5	5.1806653759E+04	1.1000E+10	1	0
6	5.1806653765E+04	1.1000E+10	2	1
7	5.1806653771E+04	1.1000E+10	3	0
8	5.1806653777E+04	1.1000E+10	4	1
9	5.1806653783E+04	1.1000E+10	1	0
10	5.1806653788E+04	1.1000E+10	2	1
11	5.1806653794E+04	1.1000E+10	3	0
12	5.1806653800E+04	1.1000E+10	4	1
13	5.1806653806E+04	1.1000E+10	1	0
14	5.1806653812E+04	1.1000E+10	2	1
15	5.1806653817E+04	1.1000E+10	3	0
16	5.1806653823E+04	1.1000E+10	4	1
17	5.1806653829E+04	1.1000E+10	1	0
18	5.1806653835E+04	1.1000E+10	2	1
19	5.1806653841E+04	1.1000E+10	3	0
20	5.1806653846E+04	1.1000E+10	4	1
21	5.1806653852E+04	1.1000E+10	1	0

C IF Manager

IF.fits - One Primary Header Unit, one Binary Tables Extension

Main Header:

```
-------------------------------------------------------------------------------
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT   Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT   Contact the NASA Science Office of Standards and Technology for the
COMMENT   FITS Definition document #100 and other FITS information.
STRTTIME= '51806 15:41:22'     / expected start time
SCAN    = '2632    '           / integer scan identifier
PROJID  = 'MHC_IFLO'           / string project identifier
SOURCE  = 'test    '           / string source identifier
SCANID  = 'test    '           / string scan identifier
COMMENT   Green Bank Telescope
COMMENT   Sky Frequency Formula:
COMMENT       sky = SFF_sideband*IF + SFF_multiplier*LO1 + SFF_offset
COMMENT   backend: name of the terminating backend
COMMENT   bank: name of the terminating set of channels
COMMENT   channel: terminating channel index
COMMENT   receiver: name of the receiver of origin
COMMENT   feed: name of the feed of origin
COMMENT   sideband: resulting sideband: upper or lower
COMMENT   polarize: resulting polarization
COMMENT   center_IF: approximate physical center frequency
COMMENT   center_sky: approximate center frequency on the sky
COMMENT   bandwidth: approximate resulting bandwidth
COMMENT   test_tone_IF: approximate physical test tone frequency, if any
COMMENT   test_tone_sky: approximate test tone frequency on the sky, if any
COMMENT   test_tone_circuit: circuit producing the test tone, if any
COMMENT   test_tone_component: component producing the test tone, if any
COMMENT   Sky Frequency Formula multiplier coefficient
COMMENT   Sky Frequency Formula sideband coefficient
COMMENT   Sky Frequency Formula offset coefficient
COMMENT   transform_count: number of transform
COMMENT   transforms: matrix of transform descriptions (frequencies in MHz)
END
-------------------------------------------------------------------------------
```

Un-named binary tables extension (header):

```
-------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                 4282 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
```

```
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                   19 / number of fields in each row
TTYPE1  = 'backend '           / label for field   1
TFORM1  = '32A     '           / data format of field: ASCII Character
TUNIT1  = 'none    '           / physical unit of field
TTYPE2  = 'bank    '           / label for field   2
TFORM2  = '2A      '           / data format of field: ASCII Character
TUNIT2  = 'none    '           / physical unit of field
TTYPE3  = 'channel '           / label for field   3
TFORM3  = 'J       '           / data format of field: 4-byte INTEGER
TUNIT3  = 'none    '           / physical unit of field
TTYPE4  = 'receiver'           / label for field   4
TFORM4  = '32A     '           / data format of field: ASCII Character
TUNIT4  = 'none    '           / physical unit of field
TTYPE5  = 'feed    '           / label for field   5
TFORM5  = '2A      '           / data format of field: ASCII Character
TUNIT5  = 'none    '           / physical unit of field
TTYPE6  = 'sideband'           / label for field   6
TFORM6  = '1A      '           / data format of field: ASCII Character
TUNIT6  = 'none    '           / physical unit of field
TTYPE7  = 'polarize'           / label for field   7
TFORM7  = '1A      '           / data format of field: ASCII Character
TUNIT7  = 'none    '           / physical unit of field
TTYPE8  = 'center_IF'          / label for field   8
TFORM8  = 'E       '           / data format of field: 4-byte REAL
TUNIT8  = 'Hertz   '           / physical unit of field
TTYPE9  = 'center_sky'         / label for field   9
TFORM9  = 'E       '           / data format of field: 4-byte REAL
TUNIT9  = 'Hertz   '           / physical unit of field
TTYPE10 = 'bandwidth'          / label for field  10
TFORM10 = 'E       '           / data format of field: 4-byte REAL
TUNIT10 = 'Hertz   '           / physical unit of field
TTYPE11 = 'test_tone_IF'       / label for field  11
TFORM11 = 'E       '           / data format of field: 4-byte REAL
TUNIT11 = 'Hertz   '           / physical unit of field
TTYPE12 = 'test_tone_sky'      / label for field  12
TFORM12 = 'E       '           / data format of field: 4-byte REAL
TUNIT12 = 'Hertz   '           / physical unit of field
TTYPE13 = 'test_tone_circuit'  / label for field  13
TFORM13 = '32A     '           / data format of field: ASCII Character
TUNIT13 = 'none    '           / physical unit of field
TTYPE14 = 'test_tone_component' / label for field  14
TFORM14 = '32A     '           / data format of field: ASCII Character
TUNIT14 = 'none    '           / physical unit of field
TTYPE15 = 'SFF_multiplier'     / label for field  15
TFORM15 = 'D       '           / data format of field: 8-byte DOUBLE
TUNIT15 = 'none    '           / physical unit of field
TTYPE16 = 'SFF_sideband'       / label for field  16
TFORM16 = 'D       '           / data format of field: 8-byte DOUBLE
TUNIT16 = 'none    '           / physical unit of field
TTYPE17 = 'SFF_offset'         / label for field  17
TFORM17 = 'D       '           / data format of field: 8-byte DOUBLE
TUNIT17 = 'Hertz   '           / physical unit of field
TTYPE18 = 'transform_count'    / label for field  18
```

```
TFORM18 = 'J       '           / data format of field: 4-byte INTEGER
TUNIT18 = 'none    '           / physical unit of field
TTYPE19 = 'transforms'         / label for field  19
TFORM19 = '4096A:SSTR256/059'  / data format of field: ASCII Character
TUNIT19 = 'none    '           / physical unit of field
END
-------------------------------------------------------------------------------

Un-named binary tables extension data (example):
-------------------------------------------------------------------------------
ROW  BACKEND  BANK  CHANNEL   RECEIVER  FEED SIDEBAND POLARIZE center_IF ....
1      DCR      A      2      Rcvr12_18  L1     U        X     2.825E+09
2      DCR      A      4      Rcvr12_18  R1     U        Y     2.825E+09
3      DCR      A      6      Rcvr12_18  L2     U        X     2.825E+09
4      DCR      A      8      Rcvr12_18  R2     U        Y     2.825E+09

ROW center_sky bandwidth  test_tone_IF test_tone_sky test_tone_circuit....
1   1.3825E+10 3.1500E+09     NULL         NULL        NULL
2   1.3825E+10 3.1500E+09     NULL         NULL        NULL
3   1.3825E+10 3.1500E+09     NULL         NULL        NULL
4   1.3825E+10 3.1500E+09     NULL         NULL        NULL

ROW test_tone_component SFF_multiplier SFF_sideband SFF_offset....
1         NULL         -1.000000E+00  1.000000E+00 0.00000E+00
2         NULL         -1.000000E+00  1.000000E+00 0.00000E+00
3         NULL         -1.000000E+00  1.000000E+00 0.00000E+00
4         NULL         -1.000000E+00  1.000000E+00 0.00000E+00

ROW transform_count  transforms
1         6          as below
2         6
3         6
4         6


Example "transforms" (line breaks inserted for formatting):

feed(Rcvr12_18:L1): 12000 to 15400, linear_x polarization;
tone(Rcvr12_18:C1L): 0 x1;
filter(Rcvr12_18:FL1L): 11800 to 15600;
mixer(Rcvr12_18:MX1L): LO: 11000 (9200 to 15600) (x1) from LO1A:synthesizer,
                                                           IFo =  IFi - 11000;
filter(Rcvr12_18:FL4L): 1250 to 4750;
attenuator(OpticalDriver2:attenuator): ;
```

D DCR

DCR.fits - One Primary Header Unit, three Binary Tables Extensions

Main Header:

```
--------------------------------------------------------------------------------
SIMPLE  =                    T / File conforms to FITS standards.
BITPIX  =                    8 /
NAXIS   =                    0 / No image data array present.
EXTEND  =                    T / Standard table extensions follow.
SCAN    =                 2632 / Scan number
SCANID  = 'test    '           / Scan ID
PROJECT = 'MHC_IFLO'           / Project ID
OBJECT  = 'test    '           / Source
UTDATE  =                51806 / MJD of start time
UTCSTART=   5.648282000000E+04 / Start time
BACKEND = 'DCR     '           / Digital Continuum Receiver
CYCLETIM=   1.000000000000E+00 / Cycle time in seconds
CYCLES  =                    1 / Cycles per Integration
DURATION=   1.000000000000E+00 / Integration time
NPHASES =                    4 / Number of Phases
INPBNK  = 'A       '           / IF bank A or B
NRCVRS  =                    4 / Number of IFs
ADVSIG1 = 'CAL     '           / Adv Timing signal one
ADVSIG2 = 'SIGREF  '           / Adv Timing signal two
ADVTIME =   2.000000000000E-06 / Time advance in seconds
COMMENT    Green Bank Telescope Project
END
--------------------------------------------------------------------------------
```

STATE Binary Tables Extension: Header

```
--------------------------------------------------------------------------------
TENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                   23 / Width of table in bytes.
NAXIS2  =                    4 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                    9 / Number of fields in each row.
SCAN    =                 2632 / Scan number
UTDATE  =                51806 / MJD of start time
UTCSTART=   5.648282000000E+04 / Start time
TTYPE1  = 'BLANKTIM'           /
TUNIT1  = 's       '           /
TFORM1  = '1D      '           /
TTYPE2  = 'PHASETIM'           /
TUNIT2  = 's       '           /
TFORM2  = '1D      '           /
TTYPE3  = 'SIGREF  '           /
TFORM3  = '1B      '           /
TTYPE4  = 'CAL     '           /
```

```
TFORM4  = '1B      '           /
TTYPE5  = 'SWSIG1  '           /
TFORM5  = '1B      '           /
TTYPE6  = 'SWSIG2  '           /
TFORM6  = '1B      '           /
TTYPE7  = 'SWSIG3  '           /
TFORM7  = '1B      '           /
TTYPE8  = 'SWSIG4  '           /
TFORM8  = '1B      '           /
TTYPE9  = 'SWSIG5  '           /
TFORM9  = '1B      '           /
EXTNAME = 'STATE             ' / extension name
END
-------------------------------------------------------------------------------

STATE Binary Tables Extension: Data (complete)
-------------------------------------------------------------------------------
ROW BLANKTIM PHASETIM SIGREF CAL SWSIG1 SWSIG2 SWSIG3 SWSIG4 SWSIG5
1   5.00E-02 2.50E-01   0     0    0      0      0      0      1
2   5.00E-02 2.50E-01   0     1    0      0      0      0      0
3   5.00E-02 2.50E-01   1     0    0      0      0      0      1
4   5.00E-02 2.50E-01   1     1    0      0      0      0      0
-------------------------------------------------------------------------------

RECEIVER Binary Tables Extension: Header
-------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                    3 / Width of table in bytes.
NAXIS2  =                    4 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                    2 / Number of fields in each row.
SCAN    =                 2632 / Scan number
UTDATE  =                51806 / MJD of start time
UTCSTART=   5.648282000000E+04 / start time in sec since midnight
TTYPE1  = 'CHANNELID'          /
TFORM1  = '1I      '           /
TTYPE2  = 'TESTDATA'           /
TFORM2  = '1B      '           /
EXTNAME = 'RECEIVER          ' / extension name
END
-------------------------------------------------------------------------------

RECEIVER Binary Tables Extension: Data (complete)
-------------------------------------------------------------------------------
ROW  CHANNELID  TESTDATA
 1       1          0
 2       3          0
 3       5          0
 4       7          0
-------------------------------------------------------------------------------
```

```
DATA Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                   78 / Width of table in bytes.
NAXIS2  =                   10 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                    4 / Number of fields in each row.
SCAN    =                 2632 / Scan number
UTDATE  =                51806 / MJD start of integration
UTCSTART=   5.648282000000E+04 / start time seconds
BACKEND = 'DCR     '           / Digital Continuum Receiver
CTYPE1  = 'STATE   '           / First data axis is State
CTYPE2  = 'RECEIVER'           / Second data axis is Receiver
TTYPE1  = 'IFFLAG  '           /
TUNIT1  = 'CODE    '           /
TFORM1  = '1I      '           /
TTYPE2  = 'SUBSCAN '           /
TUNIT2  = 'CODE    '           /
TFORM2  = '1J      '           /
TTYPE3  = 'TIMETAG '           /
TUNIT3  = 'DMJD    '           /
TFORM3  = '1D      '           /
TTYPE4  = 'DATA    '           /
TUNIT4  = 'COUNTS  '           /
TFORM4  = '16J     '           /
TDIM4   = '(4,4)   '           / data dimension of the field
EXTNAME = 'DATA              ' / extension name
END
--------------------------------------------------------------------------------
```

```
DATA Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
ROW   IFFLAG   SUBSCAN     TIMETAG         COUNTS (4,4)
1        0        1    5.1806653736E+04      3622
2        0        2    5.1806653748E+04      3621
3        0        3    5.1806653759E+04      3622
4        0        4    5.1806653771E+04      3626
5        0        5    5.1806653783E+04      3624
6        0        6    5.1806653794E+04      3619
7        0        7    5.1806653806E+04      3619
8        0        8    5.1806653817E+04      3623
9        0        9    5.1806653829E+04      3615
10       0       10    5.1806653841E+04      3622
--------------------------------------------------------------------------------
```

E Spectrometer

GBTSpectrometer.fits - One Primary Header Unit, four Binary Tables Extensions

Main Header:

```
--------------------------------------------------------------------------------
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
TELESCOP= 'NRAO_GBT'           / Green Bank Telescope (Byrd)
OBJECT  = 'test    '           / Observer Object Name
SOURCE  = 'test    '           / Observer Object Name
PROJID  = 'test    '           / Observer Project Id Name
INSTRUME= 'spectrometer'       / GBT Sub-system
SCAN    =                    1 / Telescope Scan Number
SCANID  = 'test    '           / Observer's description of scan
DATE-OBS= '2000-10-25T13:57:17.659' / Date and Time of Scan Start
UTCSTART=            5.024E+04 / Start Time in Seconds Since Midnight
UTDATE  =            5.184E+04 / MJD of Start Time
BANDWIDT=            5.000E+01 / Bandwidth Selected
COMMENT   Green Bank Telescope
END
--------------------------------------------------------------------------------
```

STATE Binary Tables Extension: Header

```
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   28 / width of table in bytes
NAXIS2  =                    2 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    5 / number of fields in each row
TTYPE1  = 'BLANKTIM'           / label for field   1
TFORM1  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT1  = 'SECONDS '           / physical unit of field
TTYPE2  = 'PHASETIM'           / label for field   2
TFORM2  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT2  = 'SECONDS '           / physical unit of field
TTYPE3  = 'PHASESRT'           / label for field   3
TFORM3  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT3  = 'NONE    '           / physical unit of field
TTYPE4  = 'SIGREF  '           / label for field   4
TFORM4  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT4  = 'T/F     '           / physical unit of field
TTYPE5  = 'CAL     '           / label for field   5
TFORM5  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT5  = 'T/F     '           / physical unit of field
EXTNAME = 'STATE   '           / name of this binary table extension
PROJID  = 'test    '           / Observer Project Id Name
```

```
SCAN    =                    1 / Telescope Scan Number
DATE-OBS= '2000-10-25T13:57:17.659' / Date and Time of Scan Start
NUMPHASE=                    2 / Number of Phases if only Internal Switching Sig
MASTER  = 'SpectralProcessor'  / Switching Signals Master
END
--------------------------------------------------------------------------------

STATE Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
ROW   BLANKTIM     PHASETIM      PHASESRT    SIGREF CAL
1   1.31072E-03 4.548198400E-01 0.0000E+00     0     0
2   0.00000E+00 5.010882456E+00 8.3213E-02     0     1
--------------------------------------------------------------------------------

SAMPLER Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                  118 / width of table in bytes
NAXIS2  =                    2 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                   10 / number of fields in each row
TTYPE1  = 'BANK    '           / label for field   1
TFORM1  = '32A     '           / data format of field: ASCII Character
TUNIT1  = 'STRING  '           / physical unit of field
TTYPE2  = 'QUADRANT'           / label for field   2
TFORM2  = '32A     '           / data format of field: ASCII Character
TUNIT2  = 'STRING  '           / physical unit of field
TTYPE3  = 'SAMPLERA'           / label for field   3
TFORM3  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT3  = 'INDEX   '           / physical unit of field
TTYPE4  = 'SAMPLERB'           / label for field   4
TFORM4  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT4  = 'INDEX   '           / physical unit of field
TTYPE5  = 'LAG     '           / label for field   5
TFORM5  = '1J      '           / data format of field: 4-byte INTEGER
TUNIT5  = 'COUNT   '           / physical unit of field
TTYPE6  = 'LEVEL   '           / label for field   6
TFORM6  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT6  = 'COUNT   '           / physical unit of field
TTYPE7  = 'SPEED   '           / label for field   7
TFORM7  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT7  = 'HZ      '           / physical unit of field
TTYPE8  = 'FFT     '           / label for field   8
TFORM8  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT8  = 'T/F     '           / physical unit of field
TTYPE9  = 'VANVLECK'           / label for field   9
TFORM9  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT9  = 'T/F     '           / physical unit of field
TTYPE10 = 'MODE    '           / label for field  10
TFORM10 = '32A     '           / data format of field: ASCII Character
TUNIT10 = 'STRING  '           / physical unit of field
```

```
EXTNAME = 'SAMPLER '           / name of this binary table extension
PROJID  = 'test    '           / Observer Project Id Name
SCAN    =                    1 / Telescope Scan Number
DATE-OBS= '2000-10-25T13:57:17.659' / Date and Time of Scan Start
NUMSAMP =                    2 / Number of Samplers Used
SWPERIOD=            5.466E+00 / Switching period
SAMPUSED= '0 4     '           / Samplers used
END
-------------------------------------------------------------------------------

SAMPLER Binary Tables Extension: Data (complete)
-------------------------------------------------------------------------------
ROW BANK QUADRANT SAMPLERA SAMPLERB  LAG   LEVEL SPEED   FFT VANVLECK MODE
1    A       1       2        2     32768   3    1.0E+08  0      0  1N2-0A-50-3
1    A       1       7        7     32768   3    1.0E+08  0      0  1N2-0A-50-3
-------------------------------------------------------------------------------

CYCLE Binary Tables Extension: Header
-------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   16 / width of table in bytes
NAXIS2  =                    1 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    8 / number of fields in each row
TTYPE1  = 'ISIGREF '           / label for field   1
TFORM1  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT1  = 'T/F     '           / physical unit of field
TTYPE2  = 'IASIGREF'           / label for field   2
TFORM2  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT2  = 'T/F     '           / physical unit of field
TTYPE3  = 'ICAL    '           / label for field   3
TFORM3  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT3  = 'T/F     '           / physical unit of field
TTYPE4  = 'IBLANK  '           / label for field   4
TFORM4  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT4  = 'T/F     '           / physical unit of field
TTYPE5  = 'ESIGREF '           / label for field   5
TFORM5  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT5  = 'T/F     '           / physical unit of field
TTYPE6  = 'EASIGREF'           / label for field   6
TFORM6  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT6  = 'T/F     '           / physical unit of field
TTYPE7  = 'ECAL    '           / label for field   7
TFORM7  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT7  = 'T/F     '           / physical unit of field
TTYPE8  = 'EBLANK  '           / label for field   8
TFORM8  = '1I      '           / data format of field: 2-byte INTEGER
TUNIT8  = 'T/F     '           / physical unit of field
EXTNAME = 'CYCLE   '           / name of this binary table extension
PROJID  = 'test    '           / Observer Project Id Name
SCAN    =                    1 / Telescope Scan Number
```

```
DATE-OBS= '2000-10-25T13:57:17.659' / Date and Time of Scan Start
MASTER  = 'SpectralProcessor'  / Switching Signals Master
END
--------------------------------------------------------------------------------

CYCLE Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
ROW ISIGREF IASIGREF ICAL IBLANK ESIGREF EASIGREF ECAL EBLANK
1       1       0     0      0      0      0        0    0
--------------------------------------------------------------------------------

LAGS Binary Tables Extension: Header
--------------------------------------------------------------------------------
TENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =               262160 / width of table in bytes
NAXIS2  =                   11 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    3 / number of fields in each row
TTYPE1  = 'TIME-MID'           / label for field   1
TFORM1  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT1  = 'dMjd    '           / physical unit of field
TTYPE2  = 'INTEGRAT'           / label for field   2
TFORM2  = '1D      '           / data format of field: 8-byte DOUBLE
TUNIT2  = 'sec     '           / physical unit of field
TTYPE3  = 'DATA    '           / label for field   3
TFORM3  = '65536E  '           / data format of field: 4-byte REAL
TUNIT3  = 'correl  '           / physical unit of field
EXTNAME = 'LAGS    '           / name of this binary table extension
CTYPE1  = 'LAGS    '           / Axis 1 of DATA
CTYPE2  = 'MODE    '           / Axis 2 of DATA
CTYPE3  = 'STATE   '           / Axis 3 of DATA
TDIM3   = '(32768,2,1)'        / Structure of 3D item (LAG,MODE,STATE)
CRPIX1  =               1E+00 / Reference Pixel of Lag data
CRVAL1  =               0E+00 / Reference Value of Ref. Pixel
CDELT1  =               0E+00 / Time offset between Lags
PROJID  = 'test    '           / Observer Project Id Name
SCAN    =                    1 / Telescope Scan Number
DATE-OBS= '2000-10-25T13:57:17.659' / Date and Time of Scan Start
INTEGCYC=                 4170 / Number of Memory Cycles Per Integration
NYQUIST =                    0 / Twice Nyquist Sampling Flag (0 = FALSE, 1 = TRU
END
--------------------------------------------------------------------------------

LAGS Binary Tables Extension: Data (1 row)
--------------------------------------------------------------------------------
ROW         TIME-MID          INTEGRAT     DATA(63336)
1    5.4575432654E+04  5.465702024000E+00  1.2000E+05
--------------------------------------------------------------------------------
```

F Spectal Processor

SpectralProcessor.fits - One Primary Header Unit, three Binary Tables Ext.

Main Header:

```
--------------------------------------------------------------------------------
SIMPLE  =                    T / File conforms to FITS standards.
BITPIX  =                    8 /
NAXIS   =                    0 / No image data array present.
EXTEND  =                    T / Standard table extensions follow.
FORMATID= 'GBSDD007'           / SDD_FORMAT_ID
PROJECT = 'test    '           / Project Id
SCAN    =                 1434 / Scan number
SCANID  = 'test    '           / Scan Id
BACKEND = 'SPAB    '           / Spectral processor
INSTRUME= 'SPAB    '           / Spectral processor
DATE-OBS= '2000-10-17T00:29:56.171' / Date of Observation
OBJECT  = 'test    '           / Source name
SPMODE  = 'SSL StdSpectLine'   / SPECTRAL PROCESSOR MODE
UTDATE  =                51834 / MJD of start time
UTCSTART=   1.796171086975E+03 / start time seconds
UTCSTOP =   1.818000000000E+03 / stop time seconds
COMMENT   Green Bank Telescope Project
END
--------------------------------------------------------------------------------
```

STATE Binary Tables Extension: Header

```
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                   72 / Width of table in bytes.
NAXIS2  =                    4 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                    6 / Number of fields in each row.
FORMATID= 'GBSDD007'           / SDD_FORMAT_ID
SCAN    =                 1434 / Scan number
SUBSCAN =                    1 / Scan record number
UTDATE  =                51834 / MJD of start time
UTCSTART=   1.796171086975E+03 / UTC start time seconds
UTCSTOP =   1.818000000000E+03 / stop time seconds
RCVRS   =                    4 / each item is index by RCVRS
TTYPE1  = 'BLANKTIM'           /
TUNIT1  = 's       '           /
TFORM1  = '4E      '           /
TTYPE2  = 'PHASETIM'           /
TUNIT2  = 's       '           /
TFORM2  = '4E      '           /
TTYPE3  = 'SIGREF  '           /
TFORM3  = '4B      '           /
TTYPE4  = 'CAL     '           /
```

```
TFORM4  = '4B      '           /
TTYPE5  = 'FFTS    '           /
TFORM5  = '4J      '           /
TTYPE6  = 'DELETED '           /
TFORM6  = '4J      '           /
EXTNAME = 'STATE              ' / extension name
END
--------------------------------------------------------------------------------
```

STATE Binary Tables Extension: Data (complete)

```
--------------------------------------------------------------------------------
ROW BLANKTIM(4) PHASETIM(4) SIGREF(4) CAL(4) FFTS(4)  DELETED(4)
1    0.00E+00   9.5002E-01     0         0    92775      0
2    0.00E+00   9.5002E-01     0         1    92775      0
3    0.00E+00   9.5002E-01     0         0    92775      0
4    0.00E+00   9.5002E-01     0         1    92775      0
--------------------------------------------------------------------------------
```

RECEIVER Binary Tables Extension: Header

```
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                   74 / Width of table in bytes.
NAXIS2  =                    4 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                   22 / Number of fields in each row.
SCAN    =                 1434 / Scan number
UTDATE  =                51834 / MJD of start time
UTCSTART=   1.796171086975E+03 / start time seconds
UTCSTOP =   1.818000000000E+03 / stop time seconds
TTYPE1  = 'RCVRID  '           /
TFORM1  = '1J      '           /
TTYPE2  = 'TAPER   '           /
TFORM2  = '8A      '           /
TTYPE3  = 'OBSFREQ '           /
TUNIT3  = 'HZ      '           /
TFORM3  = '1D      '           /
TTYPE4  = 'IFF     '           /
TUNIT4  = 'HZ      '           /
TFORM4  = '1D      '           /
TTYPE5  = 'FREQRES '           /
TUNIT5  = 'HZ      '           /
TFORM5  = '1D      '           /
TTYPE6  = 'BANDWD  '           /
TUNIT6  = 'HZ      '           /
TFORM6  = '1E      '           /
TTYPE7  = 'TCAL    '           /
TUNIT7  = 'DEGREES '           /
TFORM7  = '1E      '           /
TTYPE8  = 'TPLEVEL '           /
TFORM8  = '1E      '           /
TTYPE9  = 'FASTTIM '           /
```

```
TUNIT9  = 'SECONDS '           /
TFORM9  = '1E      '           /
TTYPE10 = 'SLOWTIM '           /
TUNIT10 = 'SECONDS '           /
TFORM10 = '1E      '           /
TTYPE11 = 'CLIP    '           /
TUNIT11 = 'SECONDS '           /
TFORM11 = '1E      '           /
TTYPE12 = 'THRESH  '           /
TUNIT12 = 'SECONDS '           /
TFORM12 = '1E      '           /
TTYPE13 = 'SYNTHL  '           /
TUNIT13 = 'CODE    '           /
TFORM13 = '1B      '           /
TTYPE14 = 'OVERL   '           /
TUNIT14 = 'CODE    '           /
TFORM14 = '1B      '           /
TTYPE15 = 'IMODF   '           /
TUNIT15 = 'CODE    '           /
TFORM15 = '1B      '           /
TTYPE16 = 'IFSYNTH '           /
TUNIT16 = 'CODE    '           /
TFORM16 = '1B      '           /
TTYPE17 = 'TAPEROFF'           /
TUNIT17 = 'CODE    '           /
TFORM17 = '1B      '           /
TTYPE18 = 'RFIEXC  '           /
TUNIT18 = 'CODE    '           /
TFORM18 = '1B      '           /
TTYPE19 = 'CLKSRC  '           /
TUNIT19 = 'CODE    '           /
TFORM19 = '1B      '           /
TTYPE20 = 'IFLO    '           /
TUNIT20 = 'CODE    '           /
TFORM20 = '1B      '           /
TTYPE21 = 'IFSIDE  '           /
TUNIT21 = 'CODE    '           /
TFORM21 = '1B      '           /
TTYPE22 = 'RFSIDE  '           /
TUNIT22 = 'CODE    '           /
TFORM22 = '1B      '           /
EXTNAME = 'RECEIVER          ' / extension name
END
--------------------------------------------------------------------------------

RECEIVER Binary Tables Extension: Data (1 Row)
--------------------------------------------------------------------------------
ROW  RCVRID TAPER OBSFREQ     IFF    FREQRES         BANDWD   TCAL   TPLEVEL...
 1     0     Box  1.42E+09 2.50E=08   1.95312E+04  1.00E+07 1.00E+00 6.3E+01

ROW FASTIM  SLOWTIM   CLIP     THRESH   SYNTHL OVERL IMODF IFSYNTH...
 1  1.0E-02 1.0E+00 9.995E+00 9.995E+00   0      0     0      0

ROW TAPEROFF RFIEXC CLKSRC IFLO IFSIDE RFSIDE
```

```
 1     0       0      0     0     0     0

--------------------------------------------------------------------------------

DATA Binary Tables Extension: Header
--------------------------------------------------------------------------------
XTENSION= 'BINTABLE'           / FITS binary table.
BITPIX  =                    8 / Binary data.
NAXIS   =                    2 / Table is a Matrix.
NAXIS1  =                32792 / Width of table in bytes.
NAXIS2  =                    1 / Number of entries (rows) in Table.
PCOUNT  =                    0 / Pointer Count
GCOUNT  =                    1 / Only one group.
TFIELDS =                    5 / Number of fields in each row.
OBJECT  = 'test    '           / Source_name
SCAN    =                 1434 / Scan number
SUBSCAN =                    1 / Scan record number
BACKEND = 'SPAB    '           / Spectral processor
UTDATE  =                51834 / MJD start of integration
UTCSTART=   1.796171086975E+03 / start time seconds
UTCSTOP =   1.818000000000E+03 / stop time seconds
INTTIME =   2.000076800000E+01 / sample time seconds
CTYPE1  = 'FREQUENCY'          / First data axis is Frequency
CTYPE2  = 'STATE   '           / Second data axis is State
CTYPE3  = 'RECEIVER'           / Third data axis is Receiver
TTYPE1  = 'SUBSCAN '           /
TFORM1  = '1J      '           /
TTYPE2  = 'UTDATE  '           /
TUNIT2  = 'MJD     '           /
TFORM2  = '1J      '           /
TTYPE3  = 'UTCSTART'           /
TUNIT3  = 'SECONDS '           /
TFORM3  = '1D      '           /
TTYPE4  = 'PSRPER  '           /
TUNIT4  = 'PULSAR PERIOD'      /
TFORM4  = '1D      '           /
TTYPE5  = 'DATA    '           /
TUNIT5  = 'COUNTS  '           /
TFORM5  = '8192E   '           /
TDIM5   = '(512,4,4)   '       / data dimension of the field
EXTNAME = 'DATA               ' / extension name
END
--------------------------------------------------------------------------------

DATA Binary Tables Extension: Data (complete)
--------------------------------------------------------------------------------
ROW   SUBSCAN UTDATE     UTCSTART          PSRPER    DATA(8192)
 1       1    51834    1.7961710870E+03    1.000E+00  4.6439E+07
--------------------------------------------------------------------------------
```