Decoder preliminary design

Randy Hudson*

20 July 1992

1 Introduction

This document outlines design requirements and considerations for the Green Bank OVLBI Earth Station Data Decoder. The Decoder is part of the high-speed data subsystem which receives QPSK modulated digital data from an earth-orbiting satellite. The remainder of the earth station as well as preliminary considerations for the decoder are detailed in [5].

The Decoder accepts two bitstreams from the demodulator. The bitstreams, designated 'I' and 'Q' both have exactly the same bit rates, which may be rates of 18, 36, 64, or 72 Mbits/sec each. Both bitstreams have a common simultaneous clock. The streams contain synchronous binary data. The data is differentially encoded, so there is no ambiguity about which bitstream is I and which is Q [3]

The bitstreams contain digitized astronomical data, spacecraft operating information, frame sync, and various forms of error checking. The format of the 64 Mbit stream is completely different from the others.

The primary purpose of the decoder is to detect frame sync. No other function is possible until proper framing has been performed.

The decoder must send the astronomical data to the recording system data formatter in a fashion suitable for the formatter. The spacecraft operating information must be readable, at will, by the station computer. The recorder requires a continuous bitstream, since the tape can not be started and stopped randomly. Pseudo-random data will be sent to the recorder during the time the spacecraft is sending information other than astronomical data.

Radioastron includes parity bits in its transmissions; the parity bits must be stripped from the data before the data is sent to the formatter. Additional use may be made of the parity. Radioastron also includes longitudinal control bytes in the header block. This can be used for further error correction and performance estimation.

The decoder supplies clock signals to the tape recorder synchronous with the spacecraft bitstream.

The decoder must include some method of determining the UTC time of decoding of some spacecraft data frame designated the 'initial' frame. For the sake of dropout recovery, the decoder should be able to record the UTC time of decoding of subsequent frames. This means, in essence, that after sync is first acquired any subsequent chagnes in sync timing must be known. Therefore, the decoder must know when any subsequent syncwords are expected and make known to the station computer any time that sync arrrives at an unexpected time. The decoder should keep track of the total accumulated sync error.

The decoder must also have the ability to perform a 'confidence' self test.

2 Frame Sync

This is the most important function of the decoder, since no other function is possible without it.

Essentially, frame sync involves finding a specific pattern in the bitstream. A timing marker is produced when this pattern is found and all other data is found relative to this timing marker.

Radioastron transmits a sync pattern composed of seven 9-bit bytes. The 9th bit is a parity bit and the 7 bytes are all different. The sync bytes are transmitted with even parity. All other bytes are transmitted with odd parity.

*National Radio Astronomy Observatory, Green Bank WV

VSOP transmits a 32 bit syncword with no parity. The complete VSOP sync word is split between the the I & Q channels.

It should be noted that none of the frame sync algorithms described below is optimum. An optimum algorithm would operate on the received signal, which is analog. All the following algorithms operate on digital data which was produced by a maximum likelihood estimate of the transmission. Some [redacted] lost in this estimation process which could have been used to improve the sync det[redacted] sort of estimate [9, 8, 10].

It would be extremely difficult to perform an optimum estimate of frame sync on data transmitted at most of the data rates envisioned for this project. Commodity parts are available to perform optimum estimation at data rates up to 20 Msym/sec, but it would be a significant challenge to do the same at 72 Msym/sec, as the task involves digitizing the incoming analog data at twice the symbol rate and then performing a significant amount of processing on the digitized information. The signal-to-noise ratios envisioned for the data links [5] do not warrant the trouble.

2.1 Method 1: Parity Detection

The simplest, though least robust, method of detecting sync with the Radioastron format is to connect 7 parity checkers to the parallel outputs of a serial input shift register. When all parity checkers report no error for even parity, the sync word has been found. If the parity-check outputs are arithmetically summed (each output having a value of 1), then a threshold can be set allowing sync to be declared with one parity checker reporting an error - or more if desired. Details can be found in [1].

Since VSOP does not transmit parity, this scheme won't work for the Japanese satellite.

2.2 Method 2: Digital Correlation

2.2.1 Serial

The most direct method is a digital correlator. In this method, the incoming bitstream is shifted into a serial-in, parallel-out shift register. Each bit of the shift-register output is compared to a holding register. This is done with an XOR gate; one input to the shift register and one input to the comparable bit of the holding register. The expected sync pattern is loaded into the holding register. For Radioastron, this is a 63-bit long pattern, so a 63-bit shift register and holding register, along with 63 XORs are needed.

The 63 XOR outputs are summed. Each output has a weight of 1, so the output sum is from 0 to 63 - the sum counts the number of mismatched bits. A magnitude comparator can be used to compare this output to a threshold and when the sum is less than the threshold, the sync pattern is found.

The hard part is the 63 bit adder. This operation can not be performed in one bit interval without the highest speed GaAs logic. Of course, it doesn't have to be if the operation can be pipelined. It is reasonable for the 'sync found' signal to mean 'sync was found 6 bits ago'. A 6-stage pipelined adder can be constructed to determine the correlation score which resulted from the 63-bit correlator 6 clock periods previous. This can be done in 10kH ECL or perhaps with FAST logic at the required speeds.

Of course, if it is not necessary to set a threshold where sync is accepted despite errors, then the adder isn't needed. The adder is replaced with a 63-bit OR gate and when its output goes low, sync has been found. This simplifies the implementation but reduces the flexibility.

A single correlator scheme can also work for VSOP, but there is a problem.

Radioastron transmits a sync code on POLARIZATION I and the same code a fixed time later on POLARIZATION II. It is only necessary to search for sync on one data stream. Both streams could be searched for added confidence of finding sync, if desired, but only one needs to be. This is not the case with VSOP, which splits its 32 bit sync pattern between the two data channels. Therefore, for direct serial correlation, it is necessary to combine both VSOP channels and shift bits through the correlator at 128 MHz.

2.2.2 Parallel

Suppose, however, that we stubbornly shift VSOP's I channel into one 16 bit correlator and its Q channel into another 16 bit correlator and combine the scores from each correlator. Now we can corrleate the two streams

in parallel if we put the right sync pattern into each correlator. Now the data rate is back to Radioastron's — a little lower, in fact.

But why not do the same thing with Radioastron's POL I channel? Instead of shifting bits into a 63-bit register at 72 MHz, why not shift alternate bits into two 32 bit registers at 36 MHz, or even four 16 bit registers at 18 MHz?

The problem with this parallel correlation scheme is that we might choose the wrong bits. Suppose we shift the 'odd' bits into one correlator and the 'even' bits into another. We must load the first correlator with the odd bits of the synchword and the second with the even bits. Since we start off with no knowledge of where we are in the bit stream, we choose 'odd' randonly. It has a .5 probability of being wrong, hence we will never find the sync pattern, since we might be comparing the odd bits with the even pattern.

There's a fix for this, however. If sync is not found in a reasonable time, say in two frame lengths, we can change the assignment of odd and even bits into the two correlators. Now we should be able to find a match. Finding sync from scratch is half as efficient, since we'll waste half our time looking for the wrong thing, and it might take twice as long. This is probably not a problem for initial acquisition of the spacecraft signal. It may be a problem if sync is lost in one or more frames during tracking - it would increase the length of time to reacquire sync and more data would be lost. Still, only a few extra milliseconds of data would be lost.

There is a practical reason for proposing a parallel scheme: It turns out that TRW sells just the correlator we need on a chip. It's comparatively cheap ($150), but it only runs at 20 MHz. Actually, the chip has four 32-bit correlators on the chip. Running in parallel, the four correlators can find sync in an 80 MHz bitstream. That's fast enough for Radioastron. The chip also has all the logic necessary to combine the correlation scores from the 4 correlators. Some extra logic is necessary to compare this score to a threshold to produce the sync pulse.

The Green Bank Decoder could use 2 of the TRW chips and search for sync on both Radioastron's POLARIZATIONS for robustness. The two chips could then be used with 8 correlators in parallel for VSOP. This is a reasonably inexpensive, robust solution.

Of course, it is possible to build a single high-speed correlator in ECL or maybe FAST logic for Radioastron. It would also be possible to do the same with 2 correlators (one for I, one for Q) for VSOP. It would be highly risky to attempt a correlator design in the XILINX FPGAs. It might be possible, but it requires creating a shift register which would operate near the toggle rates of the array and certain other combinatorial logic would also be stressed. This approach is not recommended.

2.2.3 Parallel Parallel

The parallel correlation scheme described above can also be run in parallel with itself. If four slow correlators are required to find sync, then there are 4 ways the incoming serial bitstream can be apportioned among the correlators. Four of these correlator sets can operate in parallel, each one looking for one of the four possible ways of apportioning the incoming serial stream. One of the sets will find sync each time the syncword is transmitted. A relatively small amount of additional logic is required to keep track of which correlator set finds sync. The major disadvantage to this scheme is cost. It would require 8 TRW chips to find sync this way for VSOP, and four to find sync on one bitstream from Radioastron. This is a quantity purchase for this part, however, and the cost should be under $1000 for 10 parts.

A block diagram of this method for one channel of Radioastron data is presented in fig. 1. Two such circuits would find sync for VSOP and would provide redundancy for Radioastron as described above.

The incoming serial data stream is shifted into a shift register at the full clock rate. Every 4th clock pulse, the most recently received 7 bits are latched. Each of the TMC 2220 correlator chips contains 4 complete correlators, each 32 bits in length – longer than necessary for our purposes. Let us suppose that the reference register for the bottom correlator chip (as shown in fig. 1) has been loaded such that the bottommost of its 4 registers matches every 4th bit of the syncword starting with the 4th bit. Its next register is set to match every fourth bit starting with the 3rd bit, the next starting with the 2nd, and the topmost correlator register in the bottommost chip has been programmed to match every 4th bit in the syncword starting with the first bit in the syncword.

Obviously, this bottommost correlator chip will find the syncword if the latch clock (at one fourth the bit rate) happens to coincide with the beginning of the sync word, but the phase of this clock is not determined so it might coincide with the 2nd, 3rd, or 4th bit of the syncword. The other 3 correlator chips take care of

these possibilities. The 4 correlator reference registers in each of these chips are loaded wit the same patterns as the first chip, but their data registers are presented with different bits from the latch. While the first correlator received the first four bits in the latch (corresponding to the most recently received 4 bits), the second correlator chip is presented with the 2nd through 5th bits in the latch. This correlator will find sync if the latch clock coincides with the 2nd bit of the syncword.

Similarly, the third and fourth correlators will find sync if the latch clock corresponds to the third or fourth bit of the syncword. Of course, if the latch clock is synchronous with the 5th bit of the syncword, it also coincides with the first bit, and the first correlator will find sync. All possibilities have been covered.

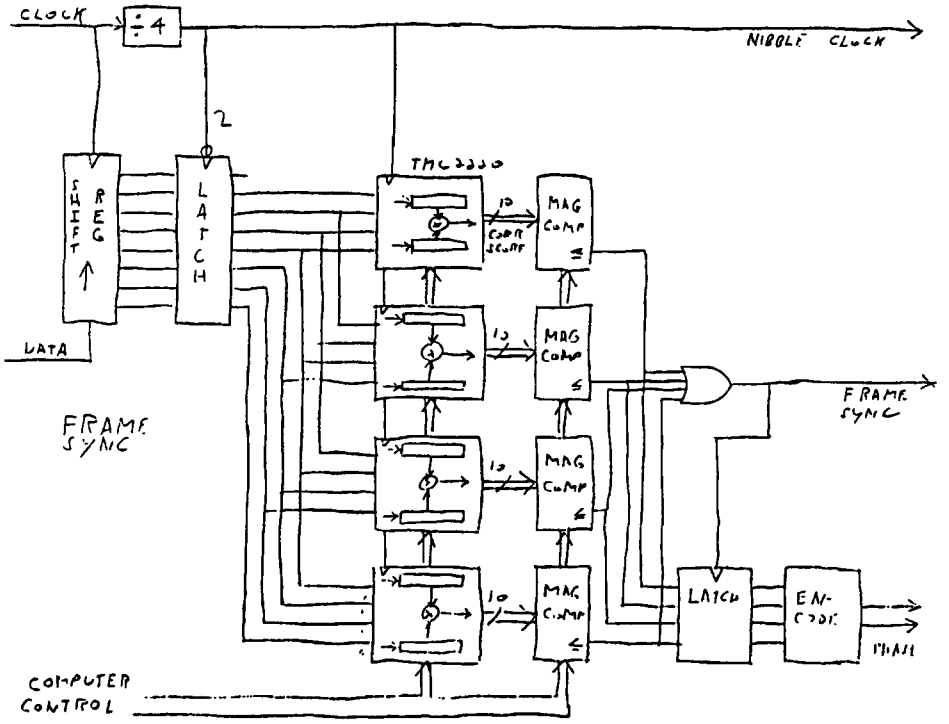

Figure 1: Frame sync circuit with correlators in parallel.

2.3 Method 3: One bit at a time.

There is another method for finding sync. It does not allow an error threshold to be set. It is not actually necessary to compare a long (63-bits in the case of Radioastron) bitstream to the complete syncword. The comparison can be made 1 bit at a time.

Compare a bit of the incoming data with the first bit of the sync pattern. If it matches, compare the next incoming data bit with the second bit of the syncword, if it matches, continue with the next syncword bit, if not, compare the next incoming bit with the first sync bit. Return to the first syncword bit whenever the incoming bit fails to match, otherwise match the next incoming bit to the next sync bit.

This operation can be done either with a parallel loadable shift register to hold the sync pattern and which is reloaded with the sync pattern whenever a bit doesn't match, or it might be done with a state machine which produces the sync pattern. Note that the shift register is parallel-in serial-out in this case.

This is a very simple system and can work very fast, it just will not accept errors in the sync.

2.4 Method 4: Fuzzy Logic

'Fuzzy Logic' is logic where each bit is not merely on or off, it has a defined probability of being on. This is precisely the sort of thing that we deal with in attempting to find a (possibly corrupted) syncword. Work is being done using fuzzy logic to find sync in serial communications. Motorola, in particular has developed applications in this area.

A device called a 'Fuzzy Data Correlator' has become available from NeuraLogix Inc. during the preparation of this document. It appears to be the best solution to the sync problem.

Although the part is called 'fuzzy' it appears to work similarly to the TRW correlator chips described above. In addition, the chip contains a threshold comparator to compare the correlation score to a predefined threshold and assert a sync signal when that threshold is reached or exceeded. This comprises a good bit of logic which would need to be added to the TRW chips.

The NeuraLogix NLX113 part operates at clock signals up to 50 MHz, rather than the TRW's 20 MHz. This allows the chip to be run at half the incoming 72MHz clock rate instead of one fourth the clock rate as necessitated by the TRW chips. This reduces the number of chips required from 8 down to 2. (see the explanation above in 2.2.3) In addition, the NeuraLogix chips are considerably less expensive, bringing the cost down from $1000 to $56.50.

3 Sync Control

The previous section discussed at extensive length methods of finding a syncword in a binary data stream. This section addresses the questions: What if sync isn't found? What if it's found in the wrong place?

3.1 Sync Qualification

VSOP transmits one 32-bit syncword in 640,000 bits. There is a relatively high probability that the syncword may be found by accident within the data, especially if sync is declared on an imperfect match. Therefore, it will be necessary to 'window' the sync. That is, sync will be accepted only if it occurs in a window around the time it is expected. The width of this window should be programmable.

3.2 Missed Sync

It is also possible that a syncword will be missed due to sudden fading. If sync fails to appear in the expected window on more than one occasion, then timing may have changed and it may be occuring outside the window, or maybe not. In this case, the decoder should begin searching for sync anywhere. With luck, sync may be found several frames later in the expected place, but if the timing has severly slipped, then it must find sync wherever it occurs.

3.3 Keeping Track

The decoder needs to report the initial acquisition of sync to the station computer. Once sync is initially acquired, the decoder needs to report any subsequent errors, such as missed sync or changes in sync timing (sync not occuring exactly when expected). In addition, the decoder needs to keep track of the accumulated changes in sync timing and make this information available to the station computer. The information will be used to correct timing when the astronomical data is correlated with data from other radiotelescopes.

4 Data Format Translation

The data transmitted from the spacecraft is divided into 'frames'. Frames may be further subdivided into 'blocks'. Each frame contains a syncword and some auxilliary information which forms a 'header'. The remainder of the frame is digitized astronomical observations. The observations are digitized to several different accuracies at several different rates from the IF signals from one or more receivers.

Radioastron transmits 180,000 bits per frame; 270 of them are in the header. Radioastron data is grouped into 9-bit bytes; the 9th bit being a parity check bit for the first 8 bits in the byte. Therefore, Rasioastron

transmits 20,000 bytes per frame. The frame is further subdivided into 11 'blocks'. There is one header (also called 'sync') block consisting of 30 bytes, and 10 data blocks of 1997 bytes each.

Radioastron computes a Longitudinal Control Byte (LCB) for each of the blocks. Each bit in the LCB is the modulo 2 sum of the corresponding bits from a previous block. For example, the first bit in the LCB is the modulo 2 sum of the first bit of every byte in a block, while the ninth bit in the LCB is the sum of all the parity bits, which are the ninth bit in each byte, for the block.

The LCBs for the 10 data blocks are transmitted as the first 10 bytes in the sync block (the header). The last byte in the sync block is the LCB for the sync block.

While the original Radioastron specifications proposed LCBs for all blocks, as just described, current expectations are that only the sync block LCB will be computed. There is some question about what will actually be transmitted as the first 10 bytes in the sync block.

VSOP transmits 640,000 bits per frame, 96 of which are header information. At least that appears to be the latest format. VSOP data is more-or-less organized into 8-bit bytes with no parity. No other error-checking transmission is proposed.

Once frame sync has been found, the meaning of any subsequent bit depends on the number of bits (or data clock cycles) that have passed between the end of the sync word and the current bit. For certain applications, the decoder must understand the actual meaning of bits in the header.

Certainly the first operation to be performed on the bitstreams after frame sync has been found (and perhaps as part of the sync search process) is a serial-to-parallel conversion — partition groups of temporally adjacent bits into groups which can be subsequently handled in parallel at proportionally lower clock rates. A reasonable size for these groups would seem to be bytes, although larger groups are also possible. In the case of Radioastron, 9-bit bytes would be a good starting point.

4.1 Parity Removal

This section applies only to Radioastron, since VSOP data is transmitted with no error checking.

Every 9th bit transmitted by Radioastron is a parity bit for the immediately preceding byte. Parity is included in the syncword, although the syncword is transmitted with even parity while everything else is transmitted with odd parity. Parity is also transmitted as part of the other error check transmission, the LCBs. Parity must be removed from the data stream before it is passed to the VLBA formatter.

Parity errors should be detected and a count kept of the number of errors found. Again, it is a matter of discussion how the decoder might respond to errors other than to simply count them and allow the station computer to read the accumulated result over some interval.

Parity may be removed from the Radioastron data simply by ignoring every ninth bit in the data stream (counting from the bit after the end of the frame syncword). Depending on the width of the serial-to-parallel conversion, one or more parity generator chips can be used to check for parity and increment a counter when errors are detected. The counter can be read and reset by the station computer.

Radioastron also transmits a 9-bit long Longitudinal Control Byte (LCB) for the short (30 byte) sync block, which includes the syncword and header information.

More advanced use can be made of the parity and LCB transmitted in the sync block if the decoder logic is sophisticated enough to do so. A single-bit error in the syncword can be detected and corrected using both parity and the LCB. If the decoder misses frame sync, it can be subsequently determined whether a timing error has occurred in many cases, using this error correction information. Making full use of this ability requires buffering the data and a fairly fast and flexible microprocessor to examine the received data and make decisions.

4.2 Spacecraft Data Formats

We now present a short description of the astronomical data formats from both spacecraft. At some point in the processing of this data, it will be necessary to know the meaning of all data bits so they can be correlated with data received at other groundstations.

4.2.1 Radioastron

Radioastron has 13 operational 'modes'. These are different combinations of number of video channels, data rate and number of bits per sample. Data rate does not affect format, so there are actually 6 different data formats transmitted by Radioastron.

There are a maximum of 8 data channels which may be digitized: 4 receiver frequencies, each with upper and lower sidebands. A data sample may be one or two bits. One bit data is essentially only a 'sign' bit, while two-bit data is coded as one sign bit and one 'amplitude' bit. Therefore, there are only 16 different 'meanings' to a bit. They are shown in Table 1. In most modes, the serial bitstream designated POLARIZATION I (effectively the I channel of the QPSK transmission) carries data received by the astronomical antenna with one polarization while the stream designated POLARIZATION II (essentially the Q channel of the QPSK transmission) carries digitized astronomical data from the other electromagnetic polarization. The different receivers are assigned to the two polarizations.

Bit #	Receiver Channel	Sideband	sign/ mag	desig
1	1	Upper	Sign bit	(1Us)
2	1	Upper	Ampl bit	(1Ua)
3	1	Lower	Sign bit	(1Ls)
4	1	Lower	Ampl bit	(1La)
5	2	Upper	Sign bit	(2Us)
6	2	Upper	Ampl bit	(2Ua)
7	2	Lower	Sign bit	(2Ls)
8	2	Lower	Ampl bit	(2La)
9	3	Upper	Sign bit	(3Us)
10	3	Upper	Ampl bit	(3Ua)
11	3	Lower	Sign bit	(3Ls)
12	3	Lower	Ampl bit	(3La)
13	4	Upper	Sign bit	(4Us)
14	4	Upper	Ampl bit	(4Ua)
15	4	Lower	Sign bit	(4Ls)
16	4	Lower	Ampl bit	(4La)

Table 1: Radioastron bits

The Russian decoder specifications call, in essence, for one data output line for each of these 16 bit meanings. If the bit is absent, its data line is unused. For instance, if only channels 1 & 3 were in operation, then bits 5-8 and 13-16 would be unused and these lines to the data recorder would be unused. If, in addition, only single bit digitizing were used, then lines 2&4, and 10&12 would be unused (because they carry the amplitude bits), leaving only line 1 to carry the sign bit for channel 1 upper sideband, 3 for lower sideband, 9 for channel 3 upper sideband, and 11 for channel 3 lower sideband. This is, in fact, a valid Radioastron operating mode: I-8-1 (also I-4-1).

The order in which the different bits detailed in Table 1 appear in the serial bitstream depends on the operating mode. Some bits are not transmitted in some modes. For details, see [1].

4.2.2 VSOP

VSOP appears to have only 3 operating modes: 2 channels / 1 bit, 2 channels / 2 bits, and 1 channel / 2 bits. For VSOP there appear to be only 2 sampling rates: 32 Million samples per second (Msps) for the 2-bit/2-chan option, and 64 Msps for the other options. The bit rate is always 128 Mbps [5, 2]. The resulting bit 'meanings' are shown in Table 2.

No error checking has been specified for VSOP, either parity or CRC. It is assumed that no error correction needs to be done.

Bit #	Receiver Channel	sign/ mag
1	1	Sign bit
2	1	Ampl bit
3	2	Sign bit
4	2	Ampl bit

Table 2: VSOP bits

4.3 Data Output Format

Data bits must be sent to the VLBA formatter hardware in a fashion that the formatter can handle and put on tape. The formatter has inputs for 32 serial bitstreams, each with a maximum bit rate of 32 Mbit/sec. The recorder uses 36 tracks to record the data plus control information. The maximum aggregate data rate for the recorder is 256 Mbit/sec. Formatter input bitstreams can be split onto multiple tracks if the data rate requires it, or combined onto single tracks if the data rate allows.

The recorded tapes will be played back, along with tapes recorded at other sites from other receivers, at the VLBA correlator where interferometric 'fringes' will be produced. It must be possible for the correlator playback equipment to 'make sense' of the bits on the tape, or at least make the same sense of them as the bits on a tape from a different site. This is the only time that any meaning is derived from the astronomical data previously transmitted serially from the spacecraft.

4.4 Demultiplexing Astronomical Data

The spacecraft astronomical data must be separated from header information in the serial bitstream, any parity or other error control bits must be removed from the data, and the data must be presented to the tape formatter in a fashion it can handle. How can this be done; more important, how can it be done simply?

For both Radioastron and VSOP, the header bits must first be separated from the astronomical data bits. The header data must be stored to await access by the station computer. The astronomical data may require further processing, such as the removal of parity, or the parity can be removed prior to the header separation.

Once the header data and parity information is removed from the data stream, how can the data be arranged to satisfy the formatter, tape recorder, and VLBA correlator hardware and software?

4.4.1 Radioastron

In spite of the many data formats transmitted by Radioastron, it is relatively simple to unscramble the astronomical data bits (See sec. 4.2.1).

The Russian decoder specifications suggest a simple serial-to-parallel shift-register which shifts 8 bits from each serial bitstream into a shift register, latches them, and then assigns the resulting 16 bits to formatter output lines with a fairly simple multiplexer circuit, whose input-to-output assignments remain the same from byte to byte. Its purpose is to assure that spacecraft IF channels always appear on the same output lines. The multiplexer routes the bits in accord with the spacecraft operating mode. If a spacecraft channel is unused, then its corresponding output line is unused. In this scheme, each formatter input line receives a continuous stream of bits, all of which have the same meaning as defined in Table 1. 16 MHz becomes the highest rate for any given bitstream, so it is within the Formatter's capability.

The multiplexer suggested in [1] may be dispensed with. If the decoder contains only a byte-wide serial-to-parallel converter, no other demultiplexing is required. In every byte, each bit always has the same 'meaning' and the maximum data rate is 8 Mbyte/sec for each channel. Therefore, simply assigning each bit from the serial-to-parallel converter (for a total of 16 lines) to a Formatter input line is feasible. The Formatter can be configured to split each input line between two tracks and record at 4.5 MHz each track and 135 ips.

4.4.2 VSOP

VSOP only transmits three different astronomical data formats.

It should be noted that a bit with any given 'meaning' (i.e. sign bit, channel 1) will arrive at a higher rate than the formatter can accept it (32 MHz) for two of the three data formats. Therefore, it is impossible to assign a single formatter input line to a given bit 'meaning'. It is inevitable that one bit will appear alternately on separate Formatter input lines.

Since the formatter has the flexibility mentioned in sec. 4.4.1, this is no problem. VSOP astronomical data can be handled exactly as Radioastron data — with just a serial-to-parallel conversion which puts the data on a sufficient number of formatter input lines to satisfy its speed constraint. The decoder then needs no knowledge of the organization of the astronomical data other than knowledge of the header bits.

4.4.3 Simplest Approach

The demultiplexing for Radioastron can be done with simple hardware, or it might be accomplished with no help from the decoder other than to present bytes from POLARIZATION I to 8 Formatter data lines and bytes from POLARIZATION II to another 8 Formatter data lines. The Formatter can select the individual lines to record.

Since the Formatter has sufficient flexibility, the simplest possible decoding scheme can be used in the decoder. Sixteen of the serial input lines to the formatter will be used. If each line is presented with every 8th bit from one of the spacecraft serial channels (I or Q), then it is assumed that the tape can be written in a format supported by the VLBA correlator.

4.5 Managing Header Information

Relatively simple logic can be devised to determine when the received data is header information. The simplest possible system is to store each header in a small memory and have a microprocessor retrieve the header data once each frame. Radioastron has the simpler requirements for header processing and produces one header every 2.5 msec. VSOP has a somewhat more complex header design, as the meaning of the bytes in the header changes from frame to frame, but the frame is repeated every 5 msec and the same header information is transmitted for 25 consecutive frames so this should give the processor plenty of time to retrieve each header and process it as necessary.

5 Recorder Clock & Timekeeping

The VLBA Formatter is designed to accept an uninterrupted bit stream that is synchronous with a 32 MHz clock (although the data rate can be a sub-multiple of the clock). This same clock is used by the Formatter to determine the time. The Formatter breaks the incoming data into blocks and inserts a header, which includes the time, into each block. The time resolution is limited. The Formatter design assumption was that its clock was derived from a stable source and therefore that the time recorded in the header is the exact UTC at the beginning of the tape frame.

The data rate from Radioastron is not synchronous with a UTC clock, indeed, its rate varies because of Doppler shift. Therefore, the time an astronomical sample was taken is not directly related to UTC time.

The recorder can run off the data clock from the downlink bitstream, since it does not vary much from the nominal required clock. The recorder internal clock will then no longer track UTC and the time headers written on each data block will no longer be the correct time.

The decoder needs to supply nominal 32 MHz and 1 Hz clocks to the formatter. All these signals can be derived from the incoming serial data clock from the demodulator. This can be done fairly simply. See fig. 2.

Some kind of time marker is necessary for the eventual correlation of the recorded data. Most of the time resolution is handled by the two-way timing subsystem [5], but the decoder is assigned the task of producing an initial time reference.

The simplest way to do this is to set the recorder clock to some arbitrary time, say 0, at the start of a run. The recorder time is determined by a tic derived from the data clock at nominal 1-second intervals. These tics will not arrive on the UTC 1-second interval; indeed, they won't even arive exactly every second.

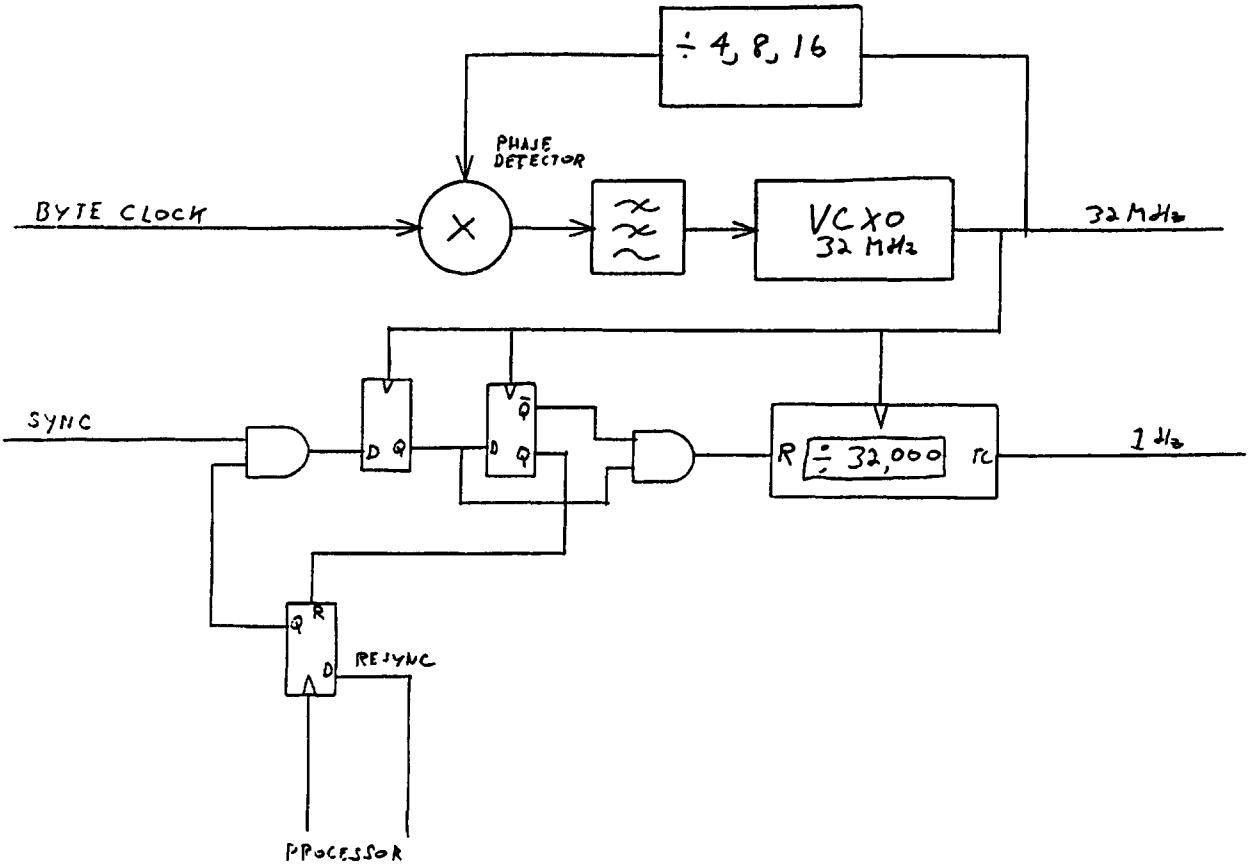

Figure 2: Formatter clock generator - block diagram

Arrange for the 1-sec tics to be coincident with data from the start of a satellite downlink data frame. Record the actual UTC time that one 'initialization' tic arrived and store this time separately in a file managed by the station computer. This is the correction factor. When the tape is played back, it will be known that the sample with tape clock value 367, say, was actually recorded at 2:57.00000003 UTC aug 23 1995, or whatever that is in Julian date. The two-way timing subsystem can then determine when the sample was actually taken on the spacecraft.

This scheme is fairly easy to accomplish. It merely requires a counter initialized to UTC time whose value can be latched in a register at the satellite 1-sec tic. This scheme is outlined in the original OVLBI preliminary design document.

The Russian decoder specifications designate satelite data frame indices which are divisible by 400 as the triggers for the 1-sec tic from their decoder. In fact, any frame can be arbitrarily chosen. If the satellite frame index is stored along with the UTC time in the station computer files, then no ambiguity will result if contact is lost and then reestablished with the satellite. The Russian convention may be used for consistency in addition.

Since the satellite data frame index is a part of the downlink header, it will be available to the station computer in any case. The special decoder function is just to keep a UTC clock run by the H-maser reference. The clock needs to be settable by computer and it needs to hold its value at the last satellite 1-sec tic so that value may be read by the computer.

6 Additional Capabilities

The Russian decoder specification for Radioastron [1] has a considerable number of capabilities beyond those mentioned elsewhere in this document.

6.1 Error Detection & Correction

The Radioastron decoder specification prefers the ability to correct errors in the data stream. Every byte transmitted by Radioastron (including the frame sync bytes) includes an additional parity bit. The original Radioastron decoder specifications [1] detailed a 'Longitudinal Control Byte' (LCB) to be computed for each 1997 byte data block and for the 30-byte block containing sync and auxiliary data. Only the LCB for the header block is currently envisioned (See sec. 4.1).

Considerable use can be made of the header LCB if the Green Bank decoder includes the capability. This capability is fairly complex, requires operations to be performed on stored data, and should be performed by a microprocessor and implemented in software.

A sufficiently powerful processor and full use of the header LCB would allow single bit errors in the syncword to be detected and corrected, thus allowing for frame sync to be unambiguously confirmed. If sync was detected at a time other than the expected time, certain causes could be detected and the correct timing confirmed before the next sync. Some data lost due to timing errors could be recovered.

Had the LCBs for the data blocks been implemented, a considerable amount of memory would have been required to make use of them, since an entire frame would have to be stored. Those LCBs would have been transmitted as many as 17,973 bytes after the data block they refer to. The capability to handle one LCB is essentially the same as the capability to handle the full set of LCBs, except for a matter of magnitude. A microprocessor would have difficulty making use of a full set of LCBs, because of the number of operations which would have to be performed each frame. With only one LCB, this limitation is removed. The memory and programmability must be retained to make use of a single LCB, but the processor should have the time to perform numerous operations on a small data block like the header, once per frame.

The Radioastron decoder also estimates data quality by counting parity errors for each data block. The total number of parity errors are reported for each frame (10 data blocks and 1 shorter sync block). If the calculated and received LCBs do not match and no parity error was detected for the corresponding block, then at least two errors must have occurred, so the Radioastron decoder specifications require that the frame error counter be incremented by 2 when this condition is observed.

6.2 Power Data

Radioastron Decoder specifications call for computations to determine the actual received astronomical power level. Specifications suggest using data from a frame near the beginning and near the end of a 1 second interval. If the station computer can access spacecraft operating data fast enough, then it can perform the computations. Otherwise, they must be performed with a microprocessor. Since the computations need be performed only once per second, there should be minimal load on the microprocessor, even though the calculations require several divisions.

6.3 Mode Information Processing

Radioastron specifications detail how spacecraft operating mode information is packed into the sync/mode block. In addition, the specifications explain how this data is formatted in messages to be sent to the control computer. The Radioastron uses a single level command structure.

7 Computer Communication

The Radioastron Decoder specification [1] defines a message-based protocol between the decoder and station computer. The messages are simple and require no parsing to determine what the message is, but the protocol is true variable-length messages. Considerable error checking is built in. The Radioastron decoder may initiate messages to the computer, and is required to initiate several 'information' messages every second.

NRAO's Monitor and Control Bus (MCB) is a sort of register-based protocol. The computer sends a value to a given address. If the exchange is a request for information, then 2 bytes may return on a separate line. The computer initiates all communication and requests for data. The responses are not variable length.

The recently introduced VSOP multiple header requirements and requests for near real-time satellite operating information increase the required data transfer rate from the decoder to the station computter to

one that may be uncomfortable for the MCB. Space is available in the station computer VME card cage, and it appears as though the decoder can be placed there and communication with the staion computer occur over the VME backplane.

The decoder itself can use its own VME computer board to communicate with the station computer and with the several sub boards needed for the decoder logic.

Each part of the decoder which needs communication with this processor will appear at an address within its card address space. Since very few registers need appear on a card, the address decoding for each memory-mapped register need be only 3 or 4 bits, all other bits being decoded as part of the card address. VME backplane interface chips are available to perform much of the task. The sync detect logic will need to produce an interupt for the decoder processor to synchronize its tasks with the frame rate. No other interrupts are necessary.

The backplane communication between the decoder computer board and its other cards need be only single-address transfers. More complex data transfers will not be necessary. The processor board itself would be determined by the number of extra features as discussed in sec. 6. If no additional capabilities are required, a bare-bones 68020 board would be more than adequate. It would be desireable to have a suitable single-tasking operating system for the board for the sake of its simple I/O and interrupt handling and timekeeping routines.

8 Design

It is recommended that custom logic design be kept to a minimum to shorten design time and reduce risk. Frame sync and parity decimation need to be custom logic, the fastest part will need to be made from MSI ECL gates. As curently envisioned, a few circuits must run at the incoming clock rate, up to 72 MHz and these will need to be constructed using ECL msi or ECL gate arrays. It appears that all functions can be comfortably built with 10kH ECL or 2.5 nsec ECL arrays.

Certain other logic involved in the sync qualification circuit would need to run at 36 MHz (see 2.4) and this can be constructed with FAST TTL and 7.5 nsec gate arrays where necessary.

Wherever speed requirements allow and other random logic is required, LSTTL should be used for compatibility with the FAST TTL. If there were not such a compatibility requirement, HCMOS would be preferable as it is the current technology, but its logic levels (except for a few HCT parts) are not compatible with TTL. Few enough parts should be rrequired that this should not be any problem.

The amount of hardware that responds to sequences in the data stream should be kept to a minimum and as much as possible done with software.

A proposed block diagram for the decoder appears in fig. 3. The INPUT section breaks the incoming data stream up into 8 or 9 bit bytes and produces a clock at the byte rate and at the sync rate. It presents bits to the SYNC DETECT block at the sync rate. No correlators have been found to operate at the maximum bit rate, but it appears that there are inexpensive parts which work at half this bit rate. See sec. 2.4 and sec. 2.2.3.

The SYNC CONTROL block takes raw sync detects and qualifies them as to whether they are in window. This block also keeps track of missed sync and 'opens' the window if sync has been missed a number of times. The STATE MEMORY & COUNTER provides control signals to the control logic. These signals depend on the relative position within the data frame. The memory counter value is stored whenever the counter is reset; the processor can read this value. This allows the procesor to keep track of 'bit slips' – sync events which occur at other than the expected time.

The PARITY block merely counts the number of parity error since it was last read by the processor. The HEADER/DATA switch inserts pseudo random bits from the PSEUDO-RANDOM SEQUENCE GENERATOR whenever the header portion of the frame is being received. The HEADER STATE MEMORY is driven by the byte clock and knows when header is expected.

The SYNTHESIZER is driven from the byte clock and its 1Hz output can be set to be coincident with the next sync pulse on command from the processor. The UTC clock is set to a value programmed by the processor. The clock will begin counting on the occurance of the next 1Hz pulse after it is set. On command from the processor, the clock will store the UTC time coincident with the next sync pulse and this time can then be read by the processor.

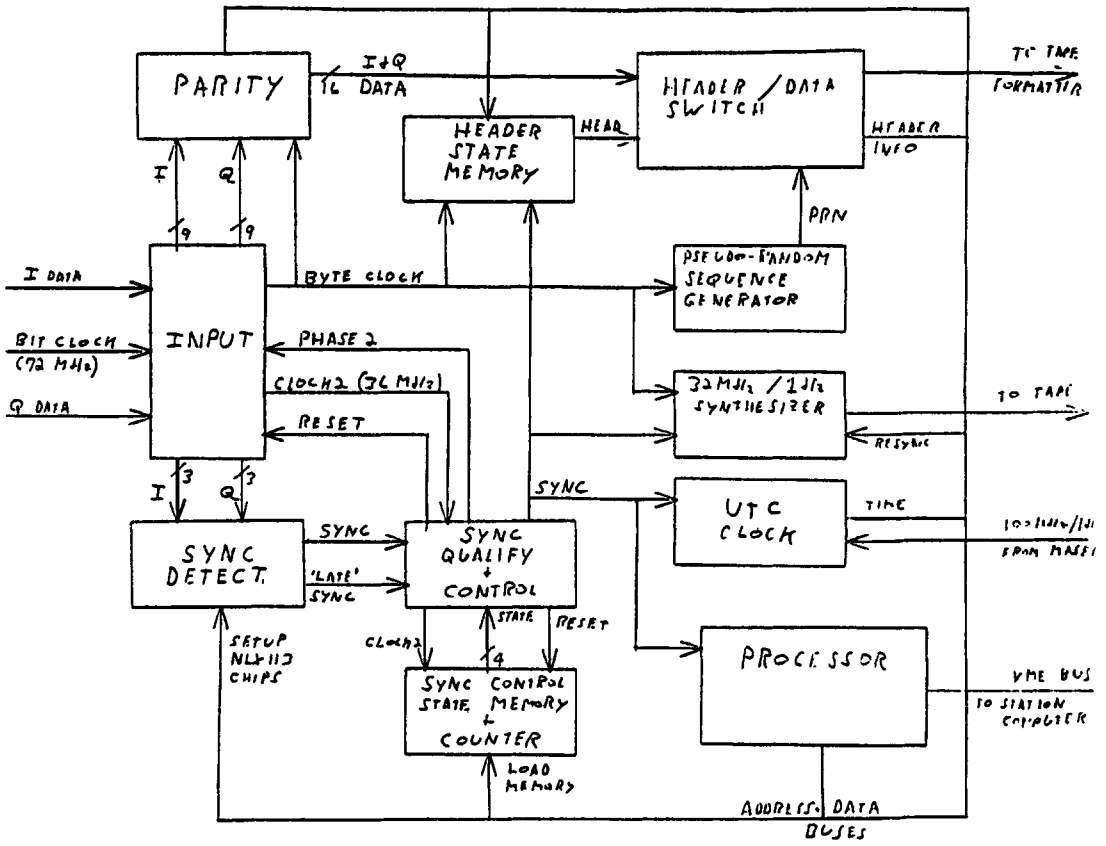

Figure 3: Decoder Block Diagram

8.1 Frame Sync

The preferred method of sync is that described in sec. 2.2.3. The sync detector can be implemented using the NeuraLogix chips described in sec. 2.4.

The sync control circuit, together with its state memory will qualify the output of the sync detect chips.

If sync has never been found, it will be accepted at any time. Once a sync pulse has been detected, sync will only be accepted if it is in a programmable window about the next expected sync point. If sync fails to appear in two consecutive windows, the window will be opened and sync will be accepted at any time. Whenever sync appears the sync memory counte will be reset and its previous value stored so that it may be read by the microprocessor. This allows bit slips to be accounted for.

This scheme will be implemented with a state memory addressed by a counter and by a small amount of combinatorial logic which can be programmd into a TTL PAL.

8.2 Parity & Error

Parity may be checked with MSI logic chips and the checking and parity removal can be implemented as shown in fig. 4. The circuit simply counts parity errors. It is the task of the microprocessor to read and reset the counters at appropriate times. If it is felt that parity count must be kept precisely on frame boundaries, then this feature may be added with some increase in complexity.

8.3 Astronomical Data Demultiplexing

It is believed that the serial-to-parallel conversion and parity decimation is all that is needed. The Formatter electronics should prove flexible enough to perform whatever demultiplexing is required.

During the intervals where pseudo-random data must be substituted for spacecraft header information, a simple circuit as shown in fig. 5 can provide data of any word size.

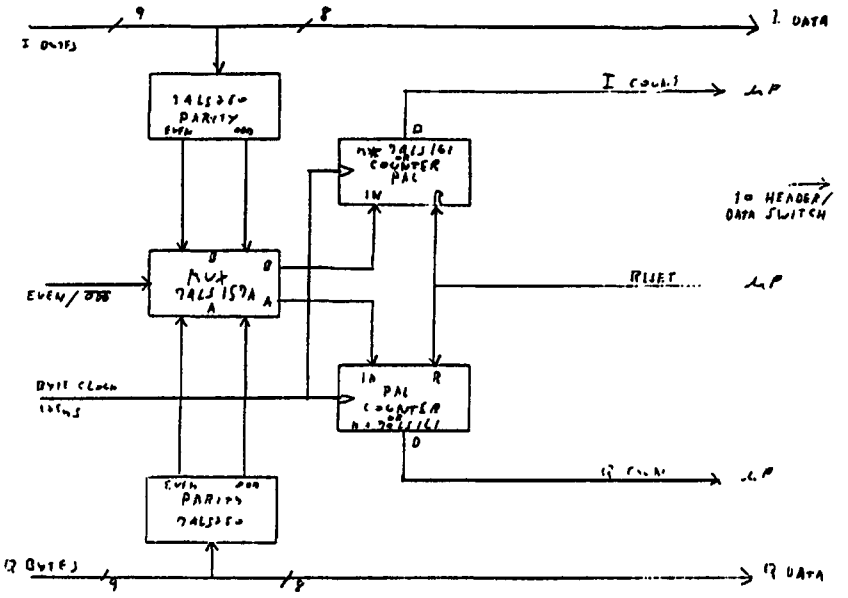

Figure 4: Parity check

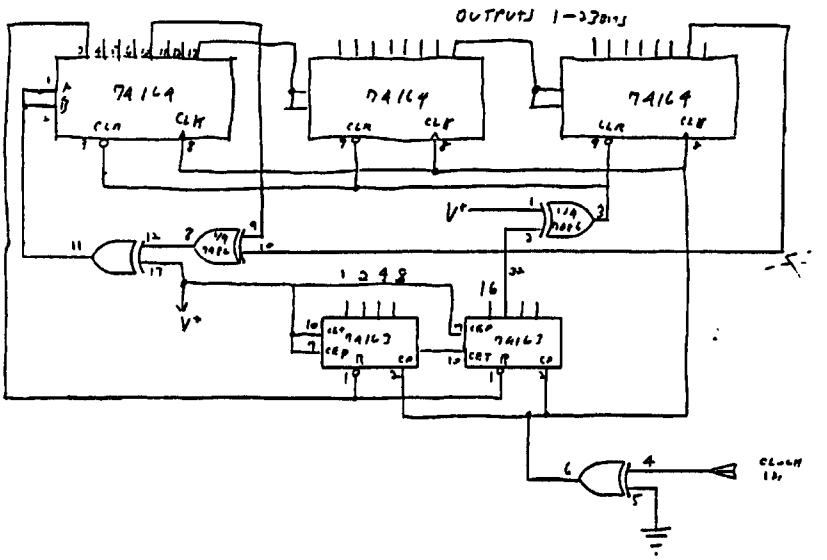

Figure 5: Pseudo-random word generator

8.4 Header Data

A ROM (actually RAM) driven state machine similar to the one used for sync control is used to control the header/data switch. Since this data is byte-oriented, the memory address counter is stepped by the byte clock.

When header data is being received, the output of the PRN generator is sent to the tape recorders instead of the header data.

An additional small memory and address counter receives the header data so that the processor can read the header information from the memory sometime during the frame.

8.5 Computer Interface

The decoder will reside in the same VME card cage as the station computer. Therefore, it can communicate with the station computer over the VME backplane.

8.6 Self Test

Self test involves creating a bitstream which appears the same as the spacecraft output and using it as input to the decoder. The decoder is then operated normally and it is determined whether it finds sync, removes parity properly, performs the correct functions on the header data, etc.

The pseudo-random astronomical data can be generated with the circuit already shown in fig. 5. Additional circuitry is neded to generate parity. The header can be inserted merely by storing its bit pattern in a long register or memory and replacing the pseudo-random sequence with the data, or by storing an entire frame in a memory.

The task is not too difficult, though not trivial, as long as the header data remains fixed and no attempt is made to determine the dynamic error performance of the decoder. It is even reasonable to load the header memory with a partially erroneous syncword and determining if the decoder finds sync given an appropriate threshold.

8.7 Additional Capabilities

The additional capabilities detailed in sec. 6 should not be implemented.

References

[1] A. Novikov. *Radioastron Downlink Decoder Specifications for USSR Earth Stations.* Astro Space Center, Lebedev Physical Institute, Moscow. April 1991.

[2] Zen-Ichi Yamamoto *VSOP Ku-band Data Transmission System*

[3] N. Kardashev, V. Andreyanov, V. Rogalskiy, V. Altunin. *Radioastron Radiolinks and Tracking Station Specification* Part 2: "Wide Band Down Radiolink"

[4] USSR Academy of Sciences: Spave VLBI. *Radioastron Tracking & Data Acquisition Requirements.* 1990.

[5] L.D'Addario, B. Shillue, & D. Varney. *The Green Bank OVLBI Earth Station: Preliminary Design.* National Radio Astronomy Observatory, Green Bank WV. 2.July.1991.

[6] L. D'Addario. *OVLBI Timekeeping with the VLBA Formatter.* OVLBI-ES Memo No. 3, 1.Aug.90.

[7] L. D'Addario *Recovery from Communication Dropouts.* OVLBI-ES Memo No. 10, 9.April.91.

[8] J. L. Massey, "Optimum Frame Synchronization," *IEEE Trans. Commun.,* vol. COM-20, pp. 115-119, Apr. 1972.

[9] Robert A. Scholtz, "Frame Synchronization Techniques," *IEEE Trans. Commun.,* vol. COM-28, pp. 1204-12, Aug. 1980.

[10] P. Tolstrup Nielsen "Some Optimum and Suboptimum Frame Synchronizers for Binary Data in Gaussian Noise," *IEEE Trans. Commun.*, vol. COM-21, pp. 770-2, June 1973.