The EVN/Mark IV Station Unit: Straw-Man Design

MARK IV MEMO #141

B. Anderson

Revision C
930302

Contents

1 Introduction

The EVN/Mark IV Station Unit (SU) is part of a VLBI system being developed by the EVN and the Haystack Observatory of M.I.T. in the U.S.A. A companion *Requirements* document [**?**] gives the requirements for the SU. This *straw-man* design attempts to show how the requirements might be met. The straw-man design is by no means exhaustive nor is any bidder for the SU development contract required to use it as the basis of their design. It is intended to be helpful and illustrative.

Some compromises have been made in the design. For example, it is assumed that the SU will be used with either 2 headstacks for 16 output channels or with 4 headstacks for 32 output channels. The expansion to 4 headstacks is achieved by using twice as many circuit boards. No provisions are made in this straw-man design for cross-connections between these two sets of circuit boards so it is not possible to get 32-channel, 1-bit output from a single headstack. Mark IIIA compatibility is thus achieved for Mode B and C recordings (14 channels) in a single pass but Mode A (28 channels) is only possible in two passes albeit with each at twice recording speed. Apart from this, it is believed, but not guaranteed, that the design is Mark IV, VLBA and Mark IIIA formats compatible.

The *suggestion* in the Requirements that the SU should be divided into recorder-dependent and non-recorder-dependent parts is only partially followed in this straw-man design. There are however, in Section 2.8, some suggestions of the modifications that would be required for this separation to be implemented. It is desirable that the extra costs associated with these changes be costed.

The process of defining what is required for the EVN/Mark IV system has relied heavily on the Mark IV Proposal and several Mark IV, Mark IIIA and VLBA memos.

2 Straw-Man Design

2.1 Introduction

Figure 2 is a schematic diagram of the straw-man Station Unit. The SU has been divided into 5 different modules which perform the functions shown by the text below the boxes in the diagram. The divisions are not necessarily representative of what might fit on a circuit card. The straw-man design has to cope with VLBA format tapes with $32 + 4$ tracks simultaneously from one headstack, with Mark IV format tapes with 64 tracks from 2 headstacks, and with an extended Mark IV with 128 tracks from 4 headstacks. Extended Mark IV operations can be achieved by doubling the number of track/channel-dependent cards.

A VLBA headstack with 36 heads is a mechanical and electrical superset of a Mark IV headstack with 32 heads. It is assumed that VLBA headstacks will be used with the EVN/Haystack SU. Since the VLBA uses only 36 tracks at a time compared with 64 for Mark IV, it is possible to handle both requirements by processing simultaneously a maximum of only 64 tracks. A cross-bar configuration switch in the Clock Recovery Module (CkRM) selects which 64 of the 72 tracks are processed. When handling VLBA

tapes, the 32 data tracks are handled by the top Track Recovery Module (TRM) in Figure 2, and the 4 cross-track parity tracks are handled by the bottom TRM. Cross-track parity corrections are performed in the Channel Recovery Module (ChRM) from these $32 + 4$ tracks. The ChRM also reverses any barrel rolling applied during recording and de-multiplexes the data channels from the tracks.

Some of the tasks performed by the SU are accomplished fairly easily. This cannot be said of the means by which the data are re-framed and by which relative delays between telescopes are achieved. In the straw-man design, the TRMs re-frame the data, decode the track-frame headers, isolate the rest of the system from tape skew, wow and flutter, and output demodulated track-based bit streams, a common Trk_Hdr waveform which marks the extents of track-frame headers in the data and a common Trk_Frm waveform which is asserted during data transfers.

The *Delay Module*, (DM), accepts channel data in the form of sign, magnitude and validity bits and buffers the data in RAM. Data transfer in blocks corresponding to several track frames is managed asynchronously by the SUCC which initialises the write address counter in the DM, the read address counter in the TRM, the frame count, the Trk_Hdr generator and then starts the transfer. Data are written to RAM in the DM if Trk_Frm is asserted and, in VLBA mode, if Trk_Hdr is de-asserted. The transfer rate is in excess of the transfer rate out the the DM RAM so that positive and negative delay rates and transfer overhead can be accommodated.

The straw-man design thus has 3 distinct areas of timing coherence which are not reflected in the module structure but which are delimited by the RAMs in the TRM and the DM. The data outputs from the DM are synchronous with the correlator system clock. The data moving between the TRM RAM and the DM RAM are track and channel synchronous with each other but are phase-independent of system timing. The data moving from the recorder heads to the TRM RAM are track asynchronous.

Control of a SU is quite complex. It is envisaged that, wherever possible, the SUCC will be used only to pass on control information from the C^3. For example, configuring the cross-bar switch in the CkRM requires data from the telescope logs which are not directly available to the SUCC. It makes sense for the SUCC software not to need to know why a particular configuration is required, only that it is required.

2.2 Clock Recovery Module (CkRM)

The straw man CkRM is shown in Figure 3. Apart from the selection of 64 from a maximum of 72 tracks, the main function of the CkRM is to recover an individual clock signal for each track. This can be accomplished economically through the use of an integrated circuit such as the AT&T T7032 Clock Recovery Circuit.

2.3 Track Recovery Module (TRM)

2.3.1 Requirements

In the straw-man design, the **Track Recovery Module** is required to

- convert from NRZM to NRZL format.
- re-frame the data.
- decode Mark III/IV or VLBA headers.
- provide track-based error statistics
- de-skew data across recorder tracks and remove effects of tape transport wow and flutter.
- replace parity bits by error bits.
- demodulate the data fields.
- supply a header marker, Trk_Hdr, which is asserted whenever a header is present in the output data and a track-frame marker, Trk_Frm, which is asserted when a frame is being sent.

2.3.2 Functional Description

Figure 4 is a schematic diagram showing the elements of the Track Recovery Module. The data from each track are converted from NRZM to NRZL format. The data are then assembled into *track frames* in RAM from which they can be read in parallel with tracks aligned and at a smooth rate independent of tape transport characteristics. A microcontroller unit (MCU) for every 8 tracks is used to manage track recovery and to communicate with the SUCC. (The figure shows the MCU as an 8751 but a more powerful processor may be required to handle the work load.) The means by which the requirements are satisfied are described in the following sub-sections.

a) Re-framing and Synchronisation

The bit stream from a track is converted into a *track frame* in RAM consisting of a number, N, of 9-bit bytes. N is programmable between 2000 and 4000 depending on the format being decoded. 9-bit bytes composed of 8 data bits plus the corresponding parity bit are separated from the bit stream using a 9-stage serial-in-parallel-out shift register (SIPO). A parity-check circuit working serially on the track bit stream checks the parity of each byte of the field. The 8 data bits in the SIPO register and the parity-error bit are written into the track-buffer RAM.

The framing process is controlled by the MCU and by 9-bit, N-bit and write address (15-bit) counters which need to be in the correct phases. The frame is generated from the apparently structureless bit stream by detecting an embedded 36-bit *sync* pattern. What happens when the pattern is detected depends on the state of the synchroniser.

If the system is in the un-synchronised state (SYNC_EN is asserted by the MCU), the detection of a sync pattern causes the 9-bit and N-bit counters to be set to the zero state, the write address counter is loaded with a starting address provided by the MCU and SYNC_EN is de-asserted.

If the next sync pattern is detected at the end of the current track frame, a SYNC_OK bit is asserted and essentially nothing else happens – the 9-bit, N-bit and write address counters must be correctly synchronised. If the pattern is detected but not at the expected position, a SYNC_NOK bit is asserted.

The MCU reads the SYNC_OK and SYNC_NOK bits and controls the SYNC_EN bit. Synchronisation is attempted by setting the SYNC_EN bit to let the next sync-pattern detection re-phase the frame counters. Synchronisation is confirmed by the coincidence of another sync pattern at the proper interval after a previous detection. Loss of synchronisation is detected by repeated failure to detect a sync pattern where it is expected to be. The SYNC_NOK bit helps to detect slippage due to missing or extraneous clock transitions and so speeds up detection of loss of sync.

The states of the framing counters are decoded to provide gating waveforms Datf, Hdrf and Tcf which are used to enable parity-fail counts in the data and header fields of a frame, and to activate the cyclic-redundancy check of the header respectively.

b) Error Detection and Statistics

Bad parities in the data and header fields of each frame are counted separately. The MCU can access both the CRC and the parity fail counters and decide whether to flag the whole frame as being bad because of lack of sync, too many errors, etc.

c) Data Handling and Demodulation

Different recorded formats are decoded by setting the track-frame length appropriately. Details of the frame header are established by the firmware. The MCU can read the headers from RAM, take majority votes across and along tape tracks to determine the time associated with the header and any other information therein.

Data are automatically de-skewed in this system. Each RAM is triple-ported: once for writing and twice for reading. Data are written into RAM independently for each track. Data reads from common addresses are always performed simultaneously for all tracks as requested by the common clock, TClk/9 (at 2.1 MHz maximum), and take priority over other accesses. Write operations are fitted in around the read operations as required for particular tracks. MCU read accesses, which have lowest priority, allow the MCU to examine track-frame headers. With access times of about 100 nS, there is plenty of margin to allow all 3 sorts of access within the overall 450 nS data cycle time.

The transfer mechanism from the RAM is outlined in Sections 2.1 and 2.5.3. Trk_Frm is asserted during the transfer from the RAM of a frame of data. Non-data replacement is not implemented in the TRM but is made possible by it. A Trk_Hdr waveform is generated by counting from the start of the data transfer for a programmable number of clock cycles. These waveforms are used in the DM to control writing into RAM: in VLBA mode, only data fields are stored in the RAM and used subsequently.

Data from the RAM are loaded into the PISO register so that the parity-error bit for the

byte is sent before the data bits. Track recovery through cross-track parity can thus have access to the error flag before any data bits are handled. In addition, the MCU can force, for each track, the error flag for each byte of a frame to be set. Finally, the data fields can be demodulated by a sequence generated in the read address counter module.

The firmware must ensure that the speed of the recorder is controlled well enough for data read and write addresses never to overlap. The mean data read and write rates must be identical.

2.4 Channel Recovery Module (ChRM)

The straw man ChRM is shown in Figure 5. The function of the ChRM is to take properly-framed bytes, apply cross-track parity corrections in VLBA mode if necessary, and then undo any barrel rolling and track/channel multiplexing. Error flags are combined to form an overall validity bit for each sample. Several clock waveforms used on the TRMs and DM are also generated here.

Track recovery from cross-track parity is performed in the DM as follows. If there is a parity-error flag in a data byte on only one track, that byte is sent with the error flag reset and the data bits corrected by applying cross-track parity. Otherwise, the byte is sent complete with error.

The **64 x 64 cross-bar switch** and the **cross-bar controller** undo barrel rolling and also route active tracks to the correct places. The controller relieves the SUCC of the burden of having to service the barrel roller on potentially every track frame. Enough storage capacity is required in the controller to perform, once initialised, at least several barrel-roll sequences. It is assumed that the cross-bar switch permits the whole of the *next* configuration to be loaded at any time and then implemented by a single transition of Trk_Hdr.

The **rate change** circuitry converts each serial stream of 9-bit bytes, parity-error flag plus 8 data bits, into 2 parallel streams of serial 8-bit data bytes and of 8-times repeated parity-error flags.

The **Mux/Demux** circuitry reverses any multiplexing of channels to tracks and vice versa. A mux element accepts up to 4 data and parity-error-flag bit-stream pairs and outputs 4, 2 or 1 bit-stream pairs at 1, 2 or 4 times the input rate respectively. The demux element accepts 1, 2 or 4 bit-stream pairs from a mux and outputs up to 4 output pairs at 0.25, 0.5 or 1 times the input bit rate. A particular recording requires track signals to be either muxed, demuxed or not changed and only some of the inputs and outputs will be valid for a particular mode.

The **clock generator** circuitry provides for the TRM, ChRM and DM clocks which can be selected in octave steps according to SYSCLK and the mux/demux requirements. In particular, TClk and TClk/9 at 18.9 and 2.1 MHz maximum respectively are required for the TRMs, and DClk and DClk/6 at 33.6 and 5.6 MHz maximum respectively for the DM.

The **1 or 2-bit data selectors** in 1-bit mode set the magnitude output bits to 1 and generate validity bits from the parity-error flag for the sign bit. In 2-bit mode, the validity

bit is formed from the sign and magnitude parity-error flags.

2.5 Delay Module (DM)

2.5.1 Requirements

The **Delay Module** is required to

- output 3-bit signal data at the SYSCLK rate of up to 32 Msample s^{-1} in correlator frames with durations of at least 10 millisec.
- start a correlator frame by generating a signal, BOCF, and overwriting 224 consecutive **validity** bits by headers of about 224 bits consisting of a frame serial number, the delay error, the delay-error rate, and individually for each channel, a phase, a phase rate and a phase acceleration for the **next** correlator frame, and finally a checksum for the header. These delay and phase parameters must be computed by the SUCC from the delay polynomial and from a phase polynomial for each channel. The period for which BOCF is asserted must be long enough to cover the time for which the correlator frame header is transmitted.
- delay to within ± 0.5 SYSCLK periods, according to the polynomial model, the signal data with respect to wall-clock time of the BOCF and time recorded in track-frame headers.
- accept channel data from the TRMs and the ChRM.
- permit VLBA astronomical data to be unsqueezed.
- extract quadrature components of at least 2 tone frequencies from an output channel selected from the 16 available.

2.5.2 Functional Description

Figure 6 is a schematic diagram of the straw-man DM. The DM acts as a buffer for 3-bit channel data: it isolates the correlator from the rest of the SU. Channel data organised as track frames are written into the DM RAMs when Trk_Frm is asserted and, in VLBA mode, Trk_Hdr is not asserted. The channel data are first serial-parallel converted from 3 bits to 18 bits in order to match the speed of the RAMs to the data rate. The data-transfer process between the TRM and the DM is outlined in Section 2.1 and is described in a more detail in Section 2.5.3.

The read process from the RAMs is initiated by BOCF at a starting address loaded from the SUCC to achieve the initial delay goals. Delay is achieved in two ways: coarsely by selecting the address of the data in RAM and finely by adjusting the phasing of the SYSCLK/6 clock (5.33 MHz maximum) which loads data from the RAMs into the parallel-in, serial-out registers. 18-bit registers on the inputs and outputs of the RAMs allow some latitude in the timing of writes to and reads from RAM. The delay is tracked in hardware by the *delay generator*.

The delay generator is basically a *digital accumulator* which is started by BOCF and which increments on every clock cycle thereafter. It is loaded with an initial delay-error (32 bits) and a delay-error rate (18 bits). Overflow from the digital accumulator causes the read address to be either not incremented or to be double incremented depending on the sign of the delay rate. The most-significant bits from the accumulator control the phasing of the PISO parallel-load clock, a re-phased SYSCLK/6.

The *Correlator Header Inserter* does just that. It is loaded with parameters by the SUCC which it uses to overwrite 224 validity bits of data on receipt of BOCF.

The similarities between the TRM and the DM are obvious. Some consideration has been given to subsuming the functions of the DM between the TRMs and the ChRM. The inclusion of a DM does however allow a significant simplification and keeps different logical functions separate.

It is envisaged that either the Phase Cal module (PCM) designed by Alan Rogers [**?**, **?**] or the one suggested by Sergei Progrebenko [**?**] will be modified for use in the EVN/Mark IV station unit. The PCM must be capable of extracting at least 2 tone frequencies simultaneously from a channel selected from the 16 available.

2.5.3 Recorder Control and Data Transfer

The SUCC needs to to be interrupted possibly a hundred times each second to control the recorder, manage data transfer between RAMs and to initiate correlator frames. The SUCC has to ensure that the SU delivers data at the appropriate wall-clock times. This is achieved with margins for tape speed fluctuations, etc. by controlling the tape speed and by using the buffering capacity of the RAMs in the TRMs and the DM. The TRM RAMs hold at least 16 milliseconds of data (12.8 track frames) and the DM RAMs hold 6 milliseconds of data (4.8 equivalent track frames).

The first stage of the process is to advance or retard the tape so that the time codes of the latest track frames are between several and about twenty milliseconds ahead of delay-adjusted wall-clock time. The SUCC must read these times codes at regular intervals and issue advance or retard commands to the recorder to ensure that this is true. The data can then be transferred between the TRM and DM RAMs up to several milliseconds before they need to be read from the DM RAM.

Data transfer between the RAMs of the TRMs and the DM is managed asynchronously by the SUCC. The SUCC initialises the TRM RAM read address, the DM RAM write address and the DM RAM read address once the data from tape are within range and start data transfer between RAMs. Data flow from the DM RAMs is more-or-less a continuous flow at a rate somewhat less than the rate at which data are written into it. The TRMs have to dispatch track frames of data to the DM at a rate the same, on average, as they arrive from the tape. The model for data flow from the TRMs to the DM is therefore one of continuous flow punctuated now and then by pauses which prevent the DM RAM from overflowing. The SUCC inspects the gap between the write and read addresses of the DM RAM: if the margin becomes too wide, the TRMs must be instructed to halt the data flow for, say, a track frame's worth of data.

2.6 Test and Performance Verification

A very important aspect of the design of the SU is how it can be tested. Data rates and formats change as the data pass through the SU making it extremely difficult to test by conventional means using logic analysers and oscilloscopes. It seems that some sort of functional testing offers the best means to verify how well various elements of a SU are working.

One obvious functional test is to pass known data through a SU and capture or even correlate what comes out. The major difficulties in this approach are how the known data can be generated, how any failure can be localised to a particular module/submodule and how the test facilities can be constructed without adding appreciably to the overall costs. It is possible that a reference tape could be written solely for performing functional tests. Another, more convenient, possibility is the construction of a computer-controlled data generator that could deliver known *track* signals to a SU.

An appropriately-controlled TRM could act as the source of known track signals. Fake track frames with headers and data could be generated and loaded into the test TRM RAM from which they could be read as if from tape. With minor interface electronics, the fake frames could be fed into the SU. These fake frames could be captured in and read from the TRM RAM of the SU being tested or, after format conversions, in the DM RAM. These RAMs offer alternative places in which other fake data could be injected into the system. By these means it seems possible to make realistic functional tests of the whole system from the SU through to the correlator; the SU up to the DM; the CkRM and TRM; or the TRM, ChRM and DM. It would be worth considering minor additions to the TRM and DM to facilitate these tests e.g. being able to inhibit track data writes to RAM when the RAM is sourcing fake data.

2.7 Control

The control of the SU and indeed the whole system is very complex. Some of the tasks that need to be performed are independent of the details of the implementation and others are not. Tasks in the former category are outlined in the requirements document [**?**] and some of the others are outlined here. The straw-man specific tasks fall into 3 broad categories: configuration, processing and test. Almost all the details of configuration can be dealt with by C^3 with the SUCC restricted to accepting data from C^3 and passing a table of values on to the appropriate element of hardware. Suggestions on how a SU may be tested can be found in the previous subsection.

- configure
  - track select (CkRM)
  - mode select e.g Mark IV/VLBA (TRM)
  - set frame size (TRM)
  - modulation ON/OFF (TRM)
  - modulation pattern (TRM)

  - track recovery ON/OFF (ChRM)
  - barrel roll ON/OFF (ChRM)
  - mux/demux mode (ChRM)
  - 1 or 2-bit data select (ChRM)

- process
  - check sync (TRM: MCUs)
  - check headers (TRM: MCUs)
  - check CRC (TRM: MCUs)
  - check parity errors (TRM: MCUs)
  - check frame times (TRM: MCUs)
  - error statistics (TRM: MCUs)
  - flag bad frames (TRM)
  - wall-clock time (SUCC)
  - transfer data to DM (TRM and DM)
  - slip tape (SUCC)
  - interpolate polynomials (SUCC)
  - load correlator frame headers (DM)
  - load delay generator (DM)
  - load read address (DM)
  - read phase calibrations (PCM)

- test
  - load test data to DM RAM and correlate
  - load test data to TRM RAM and correlate
  - load test data to TRM RAM and read from DM RAM
  - load test data to dummy track signal generator
  - read dummy track signals from TRM and DM RAMs

2.8 Non-Recorder-Dependent Parts Separation

It is desirable but not a requirement that the recorder-dependent parts of the SU are kept logically and physically separate from the non-recorder-dependent parts i.e. the delay system. This would enable the non-recorder-dependent parts of the SU to be incorporated in alternative SUs capable of handling "foreign" recorders and tape formats, e.g. the S2 system. The *re-useable* parts of the straw-man design would be the Delay Module, the SUCC and the firmware that performs the delay function. Some additional separation of the tasks in the firmware might be necessary for this to be possible. For the delay to be completely implemented in the DM RAM, it may be desirable to increase the DM size by

a factor of four to 128 k x 18 bits so that delays greater than an Earth-radius equivalent can be achieved within the RAM.

It is envisaged that the electrical interface to the DM for input signals and control is via a connector to the backplane. It would be possible to connect a DM to any system that conformed to the interface specification. The "foreign" system would have to obey *VLBA Recorder Control Commands* and deliver channel data and clock, and with, at least, a marker pulse serving the same function as Trk_Hdr in the straw-man design.

References

[1] Mark IV Memo #140 "The EVN/Mark IV Station Unit Requirements" Revision D.

[2] VLBA Acquisition Memo #248 (910328) "Proposed phase cal extractor for upgraded formatter A/D board" A.E.E. Rogers.

[3] VLBA Acquisition Memo #249 (910401) "Proposed redesign of the formatter A/D buffer module" A.E.E. Rogers and E.F. Nesman.

[4] EVN (930118) "Proposal for a Multiband & Multitone Phase Calibration Signal Extractor for the EVN Upgrade Project" S.V. Pogrebenko.