NATIONAL RADIO ASTRONOMY OBSERVATORY

Charlottesville, Virginia

June 7, 1974

VLA COMPUTER MEMORANDUM #110

# VLA DATA REDUCTION ROUTINES--INTAKE AND RED--VERSION A PRELIMINARY SPECIFICATION

Barry Clark

In the VLA the "natural" interferometer fringe is removed from the system by the combined action of the lobe rotators and delay lines. The lobe rotation is applied in the C band local oscillator. As protection against the bad things that can happen in analog circuits, the natural fringe is replaced by a Walsh function phase modulation (see VLA Electronics Memo #122 by Dick Thompson). This phase modulation is removed by the digital hardware immediately after the sampler. Having noted this fact, we procede to ignore it. After passing through the delay lines, the three level digital data streams are multiplied in all useful combinations and accumulated. These accumulations are a direct measure of the complex correlation function and require very little additional processing to convert them into the final interferometer output.

There are a number of factors of two associated with deciding how many components of a given type there are in the system. Because it is confusing, I shall go into this in obnoxious detail, despite the fact that it is well discussed elsewhere. First, in an interferometer array of N antennas, each antenna may be correlated with every other, leading to a square matrix of correlators with antennas 1, 2, 3.....N feeding into each of two sides. Such a matrix has, of course, $N^2$ intersection points, or correlators. However, the (m,n) correlator produces identical results with the (n,m) correlator, so we do not bother to build it. Also, for rather quaint technical reasons, the data from the (n,n) correlators is not useful. After discarding these, the number of interesting correlators remaining is $\frac{N(N-1)}{2}$, or , for 27 antennas, 351 correlators.

This triangular matrix picture is an appropriate way to regard the relationships of baselines. However, there is a third dimension to the picture. First, there are two complete sets of IF's, delays, and correlators which interact very little. They share the use of the antenna reflector, feeds, front ends, and parametric amplifier preamps. Then, except for shared use of the

milimeter waveguide modulator/demodulator system, they remain independent until the final stages of data processing, where the two maps may be combined. Each system has its separate C band LO, IF amplifier system, sampler, delay system, correlator set and correlator data reduction computer.

The second complication is that there are two preamplifiers at each antenna, connected to orthogonal senses of polarization. The orthogonal polarizations are not completely independent, so that it is informative to consider the product of the oposite sense (crossed) polarizations as well as the same sense products. For instance if the orthogonal polarizations are circular, right (R) and left (L), from the $n^{th}$ and $m^{th}$ antennas, the products may be expressed in terms of the traditional Stokes parameters describing the mean polarization properties of the incoming radiation as follows:

$$R_n R_m = (I + Q)/2$$
$$L_n L_m = (I - Q)/2$$
$$R_n L_m = (U + i\,V)$$
$$L_n R_m = (U - i\,V)$$

Therefore, at each intersection of our triangular matrix is not a simple correlator, but what we may call a "tensor correlator" which calculates all four of the products (RR, LL, RL, LR) shown above.

Third, even these are not simple correlators. Each IF, before being sampled, is split into two parts. One, the Cosine (or C) part, is sampled directly. The other, the Sine (or S) part, is run through a 90° phase shifter before being sampled. Each component of the tensor correlator discussed above, which we may call a "complex correlator", consists of two simple correlators, for instance:

$$R_n R_m = RC_n RC_m + i\, RC_n RS_m$$

The unit imaginary is introduced because the 90° phase shift may be represented as $e^{i\pi/2} = i$. Calculating $RS_n RS_m$ and $RS_n RC_m$ contributes nothing additional since $i^2 = -1$.

The simple correlator multiplies the digital representations of the voltages present in the IF's and averages over its readout interval ("integrates" and "accumulates" are, in this usage, synonyms for "averages").

To summarize, there are:

Two sets of correlators.
351 tensor correlators per set, 702 in all

Four complex correlators per tensor correlator, 1404 per correlator set, 2808 in all.
Two simple correlators per complex correlator, eight per tensor correlator, 2808 per correlator set, 5616 in all.
Each simple correlator output occupies two computer words, making four computer words per complex correlator, sixteen per tensor correlator, 5616 per correlator set, 11232 in all.

The situation with delay lines is nearly as complicated. To put the discussions together (though it is not relevant to the subject of this memo) I include here a similar sort of summary:

There are two delay line sets.

27 single antenna lines per set.

There are two polarizations (with independently settable delays), making 54 IF delay lines per set, 108 in all.

As mentioned above, the separation into real and imaginary parts occurs before the sampling and delaying. Therefore, each of these IF delay lines consists of two scalar delay lines, making 108 scalar delay lines per set, 216 in all.

The digital representation of the IF voltage is logically one trinary digit, but is physically represented by two binary digits. Therefore, there are two one bit delay lines per scalar delay line, four per complex delay line, 216 per delay line set, 432 in all.

For reasons which are no longer applicable in detail, it was decided that at intervals of 0.3125 seconds (3.2 Hz rate) the hardware correlators will dump their accumulations into the computer system. This is a much higher rate than we would ever wish to output data from the VLA, but is done so that various internal consistency checks may be made on the data going into each output. This memo concerns itself with the programs which accept these 3.2 Hz inputs, sum them to form the final estimate of correlation coefficient, make an internal consistency check (at the moment a check on rms deviations from the mean) and take the first steps toward converting from correlation coefficient to correlated flux density.

I anticipate the need for several versions of this program, of which more than one may be current at once. I do not think a program can be written which is all things to all men, because of CPU cycle and memory size limitations. This memo specifies version A. It will be as simple as possible, but will perform as much checking as we can conveniently do. I anticipate the need for the following versions: Version B - same as version A except with

a reduced rate of checking to save memory and CPU time. Version C - same as version B but reducing the amount, but increasing the relevance, of data passed to other parts of the system when the array is operated as two or more subarrays. Version D - short output time to increase array output data rate to map large fields of view.

Version A accepts the correlator data from an array of N antennas (where N and possibly N(N-1)/2 are assembly parameters) at a 3.2 Hz rate and produces an output every 10 seconds. It consists of two tasks: INTAKE, which runs at the 3.2 Hz rate and is concerned with accepting data from the hardware correlators, and RED, which runs at the 0.1 Hz rate, and is concerned with making the data over into a final output.

## SPECIFICATION FOR INTAKE

INTAKE will be invoked with two calling parameters. One will be the location of the buffer that it will use, and the other is an absolute number, NSHIFT, guaranteed by the invoking task to be between zero and six, whose usage is described below.

The correlator microcontroler sends its data to the computer by a high speed serial line. It is received in a special purpose device mounted in the Modcomp rack, and sent parallel to a 4805 dual unit data terminal mounted in the Modcomp PCI chassis. I only discuss here how the 4805 will appear to the programmer. The device which mediates between the 4805 and the serial output of the correlator proper will be specified elsewhere.

INTAKE will begin with a transfer initiate command (tentatively, to Group B device 8). When it is ready to input correlator data, it will input status, and wait for device ready (bit fifteen). Thereafter, the device can be assumed to always be ready. It is unnecessary to input status and test before inputting data. The input sequence IDB,r+1,9 IDB,r,8 fetches the next correlator output word in sequence. If a buffer fault occurs (that is, if the device really wasn't ready) or if a serial line parity error occurred, a service interupt will be generated. I really don't know what to do if this occurs - the alternatives are to mutter "all right, but don't do it again" or to crash the system - I tend to favor the latter. INTAKE will end with an EOB command to make the controller not busy.

The significance of the various words coming from the correlator are described here according to the preliminary specification of Art Shalloway, dated January 23, 1974. As used below, "word" will mean a 32 bit double word unless noted otherwise.

Words 0, 1 and 2 contain discretes which will be passed to the monitor programs. The right sixteen bits should be OR'd into a buffer location. Words 3-7 contain values to be passed to the monitor programs. They should be added to a buffer location.

The rightmost five bits of word 8 will be used to address four word blocks within a 128 word (16 bit words, that is) memory buffer area.

Words 9, 10, and 11 are spare and may be ignored.

Words 12-15 are shifted right 3 places to make them single precision and are stored into a buffer area as addressed by the last five bits of word 8. These words monitor the orthogonality of the sine and cosine samplers.

Words 16 to 231 contain the self-multiplier information which will be passed to the monitor routines. Each word will be accumulated in its own buffer area.

Words 232 to 231 + N(N-1)/2*8 contain the correlator data and are treated as described below.

It is understood that there will be additional monitor words sent after the data, but they are not at present defined.

Each complex correlator output, for instance the correlator words $RS_1RS_2$ and $RS_1RC_2$, will be assigned a buffer location of 6 computer words - the double precision fixed expressions for the real and imaginary parts of the complex output, and the sum square deviation (also double precision fixed). As each correlator word is input, it is added to its buffer location. It is then shifted right NSHIFT bits (which will depend on the strength of the source being observed), squared, and the resulting double precision square deviation is added to the buffer location (the same location is used for both the square of the real part and the square of the imaginary part, so that it represents the absolute square of the complex correlator output).

## SPECIFICATION FOR RED

The calling parameters for RED are first, a pointer to a buffer which has been filled by 32 applications of INTAKE, and second, the subsidary quantities NSHIFT (as used by INTAKE), NGAIN (an integer between 0 and 8 inclusive), and the IF gain buffer NORM with dimension 2*N.

The sine*cosine correlators, coming from correlator words 12 to 15, should be handled as follows: The double word arising from correlator word 3 should be shifted right eight places to make it single precision. Then the 128 words arising from correlator words 12-15 are divided by this word to normalize them, and the resulting 128 single precision words are stored in the output buffer.

The self-multiplier information arising from correlator words 16-231 will be shifted right eight places to make them single precision, and divided by the same normalization factor. The single word outputs will be placed in the output record.

The data portion of the output record will be so arranged that it is appropriately addressed by the following DO loops.

```
K = 0
DO I = 1 TO N;
DO J = I + 1 TO N;
K = K + 1;
(Input addresses, this tensor correlator, 24*(K-1)
   to 24*K - 1)
(Output addresses, this tensor correlator, 12*(K-1)
   to 12*K - 1)
(Gain factor addresses, 2*I - 2, 2*I - 1,
   2*J - 2, and 2*J - 1)
```

The components of the tensor correlator are arranged in the order $RC_mRC_n$, $RC_mRS_n$, $LC_mLC_n$, $LC_mLS_n$, $RC_mLC_n$, $RC_mLS_n$, $LC_mRC_n$, $LC_mRS_n$. The IF gain factors will be ordered $R_m$, $L_m$. The processing necessary, then, is to pick up the input double word, shift it right NGAIN places, multiply it by the appropriate two IF gain factors, and store it in the single word output position.

In order to derive the variance from the sum squared outputs, the following procedure is necessary. The output data double words are shifted right NSHIFT + 3 places, the result is squared and shifted left one place (making in the end (data right shifted by NSHIFT)**2/32). This circuitous procedure is required to avoid both overflows and loss of significance. The two (real and imaginary) parts of the squared data are then successively subtracted from the sum of squares, computing the variance by means of the formula

$$\mathrm{var}(z) = (\Sigma|z|^2 - N^* |\overline{z}|^2)/N$$

For the moment, we will pass this variance on as is. It seems likely that in the near future we will do a little further processing to construct a number which is a slightly more straightforward indication of interference or malfunction. The difficulty with the current expression is that the atmospheric phase wander will couple a few percent of the source power into the variance, so that large variances on strong sources are to be expected, and not regarded as serious errors.

RED should clear its input buffer, so that the buffer may be passed back to INTAKE without any further operation on the buffer itself, but only by passing the appropriate pointer.