NATIONAL RADIO ASTRONOMY OBSERVATORY

Charlottesville, Virginia

April 19, 1973

VLA COMPUTER MEMORANDUM #101

B. G. Clark

VLA SOFTWARE SUBSYSTEMS

The VLA software is sorted into subsystems below, and brief descriptions of each subsystem are given. These descriptions are not specifications. The detailed specifications are very dependent on the particular computer configuration chosen, as, to a slightly lesser degree, are the breakdown into subsystems, the external appearance of subsystem performance, and the order of implementation of the various tasks. The separation of software into subsystems apparently cannot correspond too closely with the separation of the computer hardware into CPU's, without the proliferation of communications between CPU's. Several subsystems as defined below may reside in one CPU, or a single subsystem may be spread over several CPU's.

1. The Synchronous system software.

   1.1 The data collection system. This program will accept parameters from the scheduling subsystem, and use it to collect data from the correlators, accumulate it for a major cycle, and buffer out the final answers. A "major cycle" is at the moment planned to be fixed at about 10 seconds duration. This cycle is planned to be an important scheduling cycle throughout the computer system.

   1.2 Receiver control system. This program will regularly inspect system elements (delay lines, lobe rotators, antenna positions, front end selection, etc.) to see if the output values should be changed. In the course of these computations this subsystem may examine parameters supplied by the scheduling system, or may make requests of it, such as informing it that a given observation is completed. It is anticipated that the following modes of operation will be recognized by this program:

      a) Observe source

      b) Observe calibrator (calculating effective system temperature for each antenna, etc.)

c) Track point without phase switching; take no correlator data. This mode would be used with a signal combiner for the IF's to use the array as a big collecting area, for example as a VLB element or to look at pulsars.

d) Check delay center values. Requires that one antenna be designated as reference; delay centers of other antennas in the subarray will be determined relative to it. I anticipate about 2 minutes observation per determination. Answers would be presented to operator for further action.

e) Point interferometer mode. Again, one antenna would be designated reference and other antennas measured relative to it.

f) Point single dish mode. It is anticipated that eventually only one of these pointing modes will prove useful.

g) Exercise receiver while slewing antennas. Receiver exercise functions are not yet fully designed. This mode will include running computer system diagnostic programs.

h) Move antennas of subarray under control panel direction, taking no data (this mode includes stop, stow, and disconnect modes).

The software will be written to divide the VLA into up to five subarrays, each consisting of one or more antennas, and each of which may operate in any of the above modes. Note, however, that hardware may restrict some of the independent operations of the subarrays. For instance, only four IF transmission system LO busses are provided for the whole array, so that observing two different lines (at odd frequencies) with two different subarrays may not be possible.

1.3 Scheduling system. This subsystem conveys information from an input source (console typewriter, digitswitches, or card reader) to an observation queue for the appropriate subarray, filling in default values of parameters as required, checking for consistency (e.g. that the observation point is above the horizon), and supplying parameters for use of later programs. The subsystem on request will supply a new set of parameters to the other subsystems. This subsystem would also control some aspects of the array connection, e.g. reconnection of an antenna to a different subarray.

1.4 Data flagging system. This system consists of two parts. One examines incoming monitor points to determine if a fault condition has occurred, the other concerns itself with definitions of fault conditions. Both operate in real time during

array operation. This calls for some care in writing definitions, since they become effective immediately and an inept definition may generate many nuisance error messages or may require a lengthy run on the assynchronous subsystem computer to remove erroneous flags. Further, a certain amount of care has to be excercised in managing such changes in order to avoid springing a new definition on an observer who is expecting something else, without prior warning. Definitions would be given in a FORTH-like language.

1.5 Connectivity system. On request from a control panel or terminal this subsystem would make a software connection between input devices (e.g. a multiplexor input on an ADC) and output devices (e.g. a D/A connected to a chart recorder, or a given printed line on a CRT terminal, or an endpoint of a vector on a graphic CRT).

1.6 Control panel system. Formats data for output to CRT screens, recognizes typed commands and routes them to appropriate subsystem, supplies default values for missing parameters, etc.

1.7 Logging system. Transfers data from electronic monitor points to disk and tape logs. Considerable data compression will be done before recording. In order to make this procedure as flexible as possible, it is anticipated that several types of compression will be implemented, and which one is applied will be selected by a table, so that it can be readily changed.

1.8 Log display system. Acts on terminal requests to read data from log maintained on moving head disk and place on graphic output devices or CRT terminals. Again expected to be programmed in a FORTH-like language, both definitions and executions being available during operation. One or two days of logging data would be accessible to this subsystem. Requests for longer past histories would go through the asynchronous subsystem.

1.9 Table maintenance system. There will be many tables associated with the system, (e.g. antenna locations, station locations, pointing connections, etc.) which must be maintained by a flexible and convenient program.

1.10 Library maintenance system. Manipulation of synchronous program libraries need not be done while the array is observing (it would be rather confusing to make this possible, I think). However, a system for modifying, saving, and manipulating programs and documenting the changes must exist and be easy to use. This system may partly reside in the asynchronous computers.

2. Asynchronous System Software.

2.1 Synchronous data acceptance and partial sorting task. Every 10 seconds, the program in execution is rolled out in its entirity, the latest data from the synchronous system is read in, sorted by u,v within itself, and merged with a partial string from disk. If the resultant string exceeds a limiting size, it would be discontinued after writing back to disk. At regular intervals the partial strings are merged to produce complete ordered data strings.

2.2 Map Immediate task. Takes, on request, the partial data strings from this scan, produces a completely sorted string, merges it (if requested) with those from other recent scans, executes the Fourier transform and activates a specified display task.

2.3 Map manipulation system. The WSRT system of making available such things at +, -, *, /, $\sqrt{}$, and CLEAN, whose operands are maps, seems to be the best system. This system would be tailored to terminal operation.

2.4 Map display programs. One program for each appropriate graphic device output (perhaps sharing subroutines) is needed for each of the following displays: radio photograph (gray-scale map), sky slices (alá the 5C presentations, or the beam presentations in the VLA proposals), contour maps, and polarization vector maps.

2.5 Data flagging program. This is an equivalent of the synchronous data flagging program, to be run when an observer must change the flags that were in force at observe time.

2.6 Log review programs. Various sumarization, report generation, and terminal access programs must be written to process the monitor data logs.

2.7 Sort-merge-map routines. The equivalent of the sort and map immediate tasks for reprocessed data.

2.8 Interferometer processing programs - programs to average data from each correlator, to display the averages, to check for a slightly resolved strong source, to evaluate the system noise of each antenna, to estimate baseline parameters, and to search for apparent system temperature variations with elevation or delay.