National Radio Astronomy Observatory
Socorro, New Mexico
Very Large Array Program

VLA COMPUTER MEMORANDUM 172

WHATEVER HAPPENED TO THE PIPELINE?

Robert T Duquet

August 1984

INTRODUCTION

This is a short review of the history and of the current status of the VLA computer system known as the "Pipeline". The combination of hardware and software referred to by that name was first suggested nearly a decade ago but has just recently come into limited productive use.

Despite its very long development period, the Pipeline may still justify the time and effort that went into its construction. No other system, available at the present or in the near future, can do as much to widen the class of spectral-line experiments that can be conducted at the VLA. The problem, of course, is the frustrating mismatch between the rate at which spectral-line data can be acquired at the VLA and the rate at which it can be processed.

Some important lessons can be learned from a review of the Pipeline history. They should be taken into account in the VLA long range computer plans.

GENESIS

As early as 1973, a planning committee decided that the VLA mapping (imaging) facility would consist of a network of machines rather than a single conventional computer. The nature of the network was not specified. The committee recognized that no digital machine available at that time could cope with the flood of data expected from the completed telescope. It hoped that, before the data inundation began, suitable analog optical devices would available to handle most of the imaging problem. An important assumption upon which the planning committee operated was that the data emerging from the on-line system would be flagged and calibrated (at least to first order). Although the CLEAN algorithm for removing instrumental effects from the data was known and in use elsewhere at that time, the committee assumed that it would not be needed for VLA data.

A DEC10 was purchased as an intermediate development facility which could handle, by itself, the mapping of data received from as many as ten antennae. (From 10 antennae one can derive 45 baselines compared with 351 obtainable from the full complement of 27 dishes.) Beyond the 10-antenna level, the DEC10 was expected to serve as a node in the anticipated (but unspecified) net. A PDP 11/40, supporting various display devices, was connected to the DEC10 through a terminal port and became, thereby, the first additional node on this net.

The notion that VLA data should be mapped by means of a digital pipeline synthesized from minicomputers was put forward in a memo written by Barry Clark in 1975 (VLA Computer Memorandum 127). During the next two years much effort went into evaluating alternatives, including the use of an optical device. A note by Ehnebuske et al. to the VLA planning committee (1976) proposed a 12-machine minicomputer net which they felt would be needed by the VLA even if the optical processor was successful. The function of each mini was clearly stated in the Ehnebuske proposal but there was no mention of the method of intercommunication.

In 1977 Clark issued a second memo (Computer Memorandum 137) recommending a revised form of pipeline which relied heavily upon array processors controlled by just a few minicomputers to achieve the required computational power at a relatively low cost. (The communication method was still unspecified.) This is the form in which the pipeline eventually materialized. Not until 1980, however, was the first Pipeline hardware purchased and the first Pipeline software written.

VLA Computer Memorandum 137 is remarkably brief; including block diagram, budget and action plan it is only 8 pages long. Its most important aspect, therefore, is the range of subjects that it does NOT discuss. For example, it offers no estimate of the programming man-years that would be required by the project. Likewise, it does not mention the relative role of the Pipeline and the AIPS project that was being planned concurrently (see Computer Memoranda 140 and 141, 1977).

All in all, it is clear that a very large amount of deliberation and investigation preceeded the construction of the Pipeline. But almost all of it went into systems that were never implimented (eg the optical processor). The array processor made the pipeline so much cheaper and simpler than other alternatives (such as the network proposed by Ehnebuske et al.) that design details were never committed to paper. In addition, the problems associated with communication between units appear to have been consistently overlooked.

PROTOTYPE

The Clark memorandum suggested that a prototype pipeline be constructed from a single minicomputer (PDP 11/70) and an array processor (AP120B). The lack of programming staff forced a delay of somewhat more than a year but in 1978 the prototype system, named MAPPER, was brought into production. Within the context of its objectives it was a complete success.

Approximately two and a half man-years of programming went into the initial MAPPER system. Table 1 shows the distribution of this effort by task. The central program was written in FORTRAN IV. The array processor was controlled by calls to subroutines, almost all of which were supplied by the vendor. Only a few special array processor routines were microcoded (by Clark himself) but one of those (CLEAN) was the most critical element in the system. The largest software effort was required for the communication elements whereby data and control information was transfered between the DEC10 and the PDP 11/70. This component, called HARVEY, had to be written in assembly language. HARVEY also provided communication with the PDP 11/40 dedicated to display tasks thereby incorporating that machine into the prototype system.

Table 1
Prototype Pipeline Software Effort 1978-1979

Person	Time	Task
Al Braun	9 months	Communications
Dave Ehnebuske	6 months	Communications (DEC10 interface)
Barry Clark	3 months	Design. Microcode. CLEAN Algorithm
Bob Payne	6 months	Principal programmer
Jim Torson	6 months	Graphic display
Total	30 months	(about 2 1/2 man-years)

DEMISE OF THE PROTOTYPE

For more than three years the prototype Pipeline was used as the primary means of producing images from VLA data. It became a crucial part of VLA computing partly because its hardware was inherently much faster than the DEC10 but mostly because Clark's CLEAN algorithm was much more efficient than the counterpart on the DEC10. In some ways the success of the prototype was responsible for delaying the full Pipeline. Because the prototype was so important as a production tool, pressures to improve it on a piecemeal basis diverted manpower from proper implementation of the full system.

The prototype Pipeline remained an essential part of VLA software until the installation of the first AIPS system, at which time it was essentially abandoned. The abruptness with which the MAPPER system was discarded foreshadowed problems that would be faced by the full Pipeline system. Astronomers preferred the AIPS system because:

1- Its communication with the DEC10 was simple yet reliable (magnetic tape carried by hand). The HARVEY communication software that linked the DEC10 and the PDP 11s in the MAPPER system had been plagued with bugs (arising mostly from the antiquated DEC10 operation system). More than anything else, the unreliablility of the MAPPER system alienated its users.

2- AIPS incorporated a new computer technique (SELFCAL) for extracting more information from a given set of data. In this respect, AIPS represented a logical progression from the MAPPER system whose initial popularity was due in part to the great improvement of its CLEAN algorithm over the counterpart on the DEC10. (An attempt had been made to incorporate SELFCAL into the MAPPER system but the results were never satisfactory.)

3- AIPS was an interactive single-user system. The MAPPER system was a batch system that queued requests from multiple users. Astronomers clearly prefered the interactive mode of operation.

4- The graphics facilities on the AIPS system were superior to those connected to the prototype Pipeline. In particular, it was easier to obtain larger maps (images) on AIPS. Furthermore, the display facility was an integral part of the AIPS system so that there were no communication problems involved as there were between the mapping and the display components of the prototype pipeline.

5- The AIPS system was exportable to other institutions so that an observer could begin processing the data for a quick "first look" at the VLA then carry it home for further processing without having to convert the data (or himself) to a new system.

In an ideal world, the full Pipeline would have been redesigned after the introduction of AIPS in order to merge its best features with the advantages (mostly throughput) that a pipeline could offer. There was, indeed, some review. The most important decision was to adopt DECNET software in place of the homemade communications software. Unfortunately, that decision was undercut by the failure of DEC to deliver a useable DECNET for the operating system (TOPS10) in the central pipeline node. Furthermore, the fundamental problems that arise from dealing with an inadequate operating system were not fully appreciated.

FULL IMPLIMENTATION

The main reason for the delay between the design of the pipeline (Memo 137, 1977) and the beginning of work on a full version was the lack of programmers to create the software. In 1981, Dr. Wim Brouw, Director of Netherlands Foundation for Radio Astronomy, devoted a year's leave of absence (spent at the VLA) to the sole purpose of writing the full Pipeline software. It was expected that, with some modest amount of help, he could complete the task within the time available.

Unfortunately Brouw had to return to his normal duties before the full Pipeline could be placed in production. At his departure, the only incomplete portion seemed to be the communications link with the synchronous computers. In order to debug his code, Brouw had put together a temporary scheme for reading data from the DEC-10 data base. It was also known that "a few trivial glitches" remained in the code. A member of the permanent VLA programming staff (Bob Payne - who had programmed a large part of the prototype pipeline) was assigned to complete the project.

As things turned out, the programming effort required to bring the full Pipeline into production has been greater than 10 man-years. The distribution by task is shown in Table 2. There were several reasons for this many-fold increase of actual over estimated programming time. The first reason was that the initial estimate was unrealistically low. The other reasons will be described in detail below.

Table 2
Pipeline Software Effort 1981-1984

Person	Time	Task
Al Braun	2 months	Operating system patchs
Wim Brouw	14 months	AP Microcode, Main FORTRAN framework
Barry Clark	3 months	Interface to synchronous system
Phil Dooley	12 months	Special Hardware Construction
Bob Duquet	18 months	Utility routines, Documentation
Miller Goss (et al)	3 months	Validation
Eric Graham	12 months	Task control, Queueing
Bob Kummerrer	1 month	DEC-10 interface
Bob Payne	36 months	Principal programmer (all aspects)
Jim Torson	20 months	Graphic display
Total	121 months	(more than 10 man-years)

The length of time required to bring the pipeline into routine use was bad enough; even worse was the apparent unpredictability of the project. Completion of at least a partial pipeline was repeatedly announced and repeatedly delayed. The result of such uncertainty was a widespread impression that the pipeline project was a monumental failure.

Many factors contributed to the difficulty of creating the full pipeline:

1- Intermittent faults in "home-brew", one-of-a-kind, hardware (the transpose memory) were mistaken for software errors. More than a man year of the most frustrating and demoralizing programming effort went into chasing non-existant "bugs".

2- A concurrent defect in a part of the conventional hardware (the AP's) greatly confused the issue. The problem was one of timing and its effects were data dependent. Furthermore, the effect could not be reproduced consistently even with a given data set. Not until the pipeline had already been severely delayed was this condition identified as a separate problem.

3- Crucial parts of the pipeline software involved microcode rather than high level language. Validating this massive and complex code was made abnormally difficult by the absence of the original programmer and by the lack of any flow chart, structure diagram, or almost any other system documentation.

4- In the microcode refered to above, a strong emphasis was placed on efficiency even at the cost of complexity. As it turned out, the efficiency could not be achieved because of the hardware failures described in item 2 but the complexity remained (and even worsened as the microcode was rewritten to bypass the hardware limitations).

5- Program development tools were inadequate. During the early days of the project the only version of FORTRAN that was available did not contain features to allow structured programming. Despite the presence of reasonable amounts of physical memory the software was restricted to grossly inadequate address space. This added to the system complexity by forcing numerous levels of program overlays.

The combination of these factors led to not only a painfully protracted development period for the pipeline but to a demoralizing waste of human resources.

The unfortunate problems with intermittent hardware that were encounterred in the pipeline can be construed as a piece of bad luck. It was certainly that, but it was more than that too. The pipeline was not the first computer project designed to achieve a major astronomical objective by being very clever with very small resources. About the time the pipeline was suggested, an astronomer in Groningen was finally solving a sporadic hardware problem that, for three years, had frustrated his attempt to CLEAN images by doing map arithmetic on an IIS device. (The problem turned out to be a ground loop on one of the IIS boards.)

The pertinence of this history to the NRAO long range computer plan has to do with the often heard claim that a "clever" approach to computational problems would eliminate the need for costly computational resources. What the claimants inevitably overlook is the indirect costs of such cleverness. The pipeline history clearly illustrates some of those costs:

1- Vulnerability to unforseen problems.

2- Uncertainty in scheduling.

3- Absorption of scarce human resources.

4- Inflexibility in the face of changing requirements.

The last point, inflexibility, is so very important that it is the subject of a separate section which follows next.

SHOOTING AT A MOVING TARGET

In six crucial areas, assumptions made in the design of the pipeline have been invalidated by later events. These six areas are:

The requirement for data flagging

The demand for user interaction

The need for sophisticated displays

The desirability of image enhancement through iteration

The need to interface with many disparate systems

The usefullness of Map Arithmetic

The most important difference between the pipeline described in Barry Clark's 1977 memo and the system presently in use is the "hands-on" involvement of the observer. As originally envisaged, the pipeline was an extension of the on-line data-acquisition system which automatically converted raw numeric data into maps (i.e. images). The observer, at a later time, might create modified and enhanced images using a post-processing system such as AIPS but the original pipeline maps would never be recomputed (much as the correlator output is currently considered sacrosanct).

This vision of the pipeline assumed that bad data (e.g. interference) could be detected automatically and discarded whenever appropriate. In fact, no algorithm for recognizing contaminated data has yet been developed which can satisfactorily replace an observer's trained eye. Furthermore, certain types of data flaws can only be detected AFTER that data has been transformed into an image and subjected to considerable processing ("CLEANED"). The pipeline has therefore been converted from an automatic batch system into one in which interaction with the user (and with other systems accessed by the user) is an important aspect.

The prototype pipeline failed to point out the importance of user access to the data. Since all the data (not just calibrators) resided in the DEC10, it was directly accessible to the user. In the full pipeline only calibrator data is accessible - the rest is isolated on the Pipeline's own disks.

The second major difference between the pipeline design and actuality is the existance of algorithms for extracting more information from the data than was previously possible. The pipeline was designed at a time when the dynamic range in VLA images was expected to be about 100:1; that figure is now 10,000:1. The improvement has been achieved through numerical processing techniques that increase the computing load for image formation by more than an order of magnitude. More important, the new algorithms involved a type of iteration for which the fundamental design of the pipeline is poorly suited.

The third unforseen circumstance that affects the pipeline is the existance of an alternate system (including both hardware and software) for creating images from raw correlator data. In some circumstances the most valuable current role of the pipeline is that of interface between the raw data and the alternate map-making package (AIPS).

The net effect of these three differences is that, in the current operational pipeline system, at least half of the software performs tasks that were not part of the initial design. An indirect consequence is that the pipeline is much more complex than initially intended. In fact, the programming effort for the main part of the full pipeline (Dr Brouw's work) was a bit more than 1 man-year but 3 additional man-years were required to debug and revise that task, and 5 additional man-years were required for new or auxiliary tasks. (Details were shown in Table 2).

COMPLEXITY

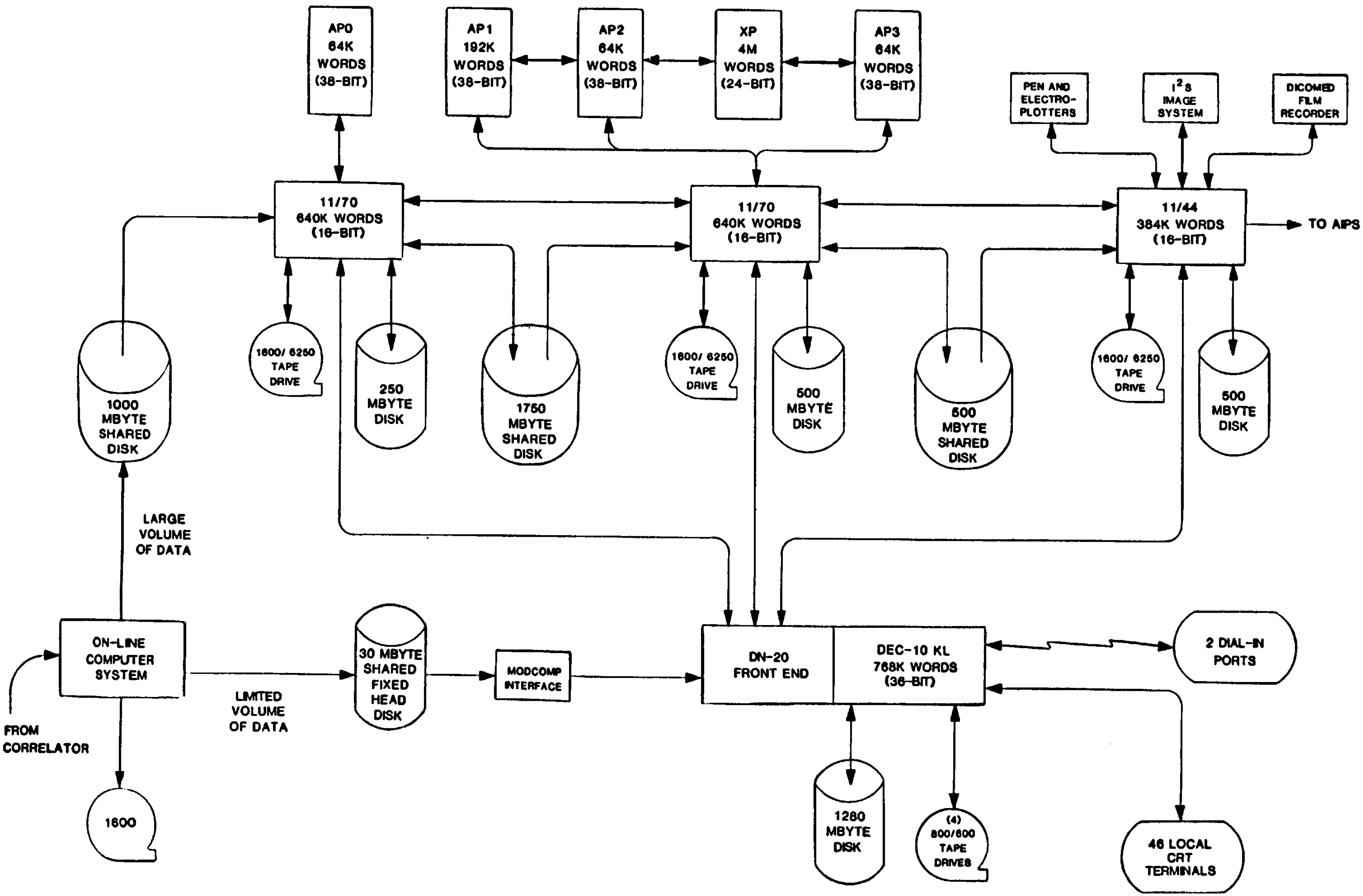
Data flowing from the VLA correlator through the pipeline to a final image in AIPS must pass through a bewildering mishmash of computer hardware. The principle components of the Pipeline and the data format in each are shown in Table 3. The interconnection between units is shown schematically in Figure 1.

Table 3
Pipeline Hardware and Data Format

Vendor	Device	Word Length
Modcomp	Modcomp	16 bit integer (*)
Century	Disks and Controller	
DEC	DEC-10	36 bit (packed 18-bit integer)
DEC	ANF-DECNET converter	
DEC	PDP-11	16 bit integer (*)
Century/Emulex	Shared disk/Controller	
FPS	AP-120B	38 bit floating point
Dataram	DATARAM	24 bit scaled integer
Telex/IPS	Tape drives/Controller	16 bit scaled integers
DEC	VAX 11/780	32 bit floating point

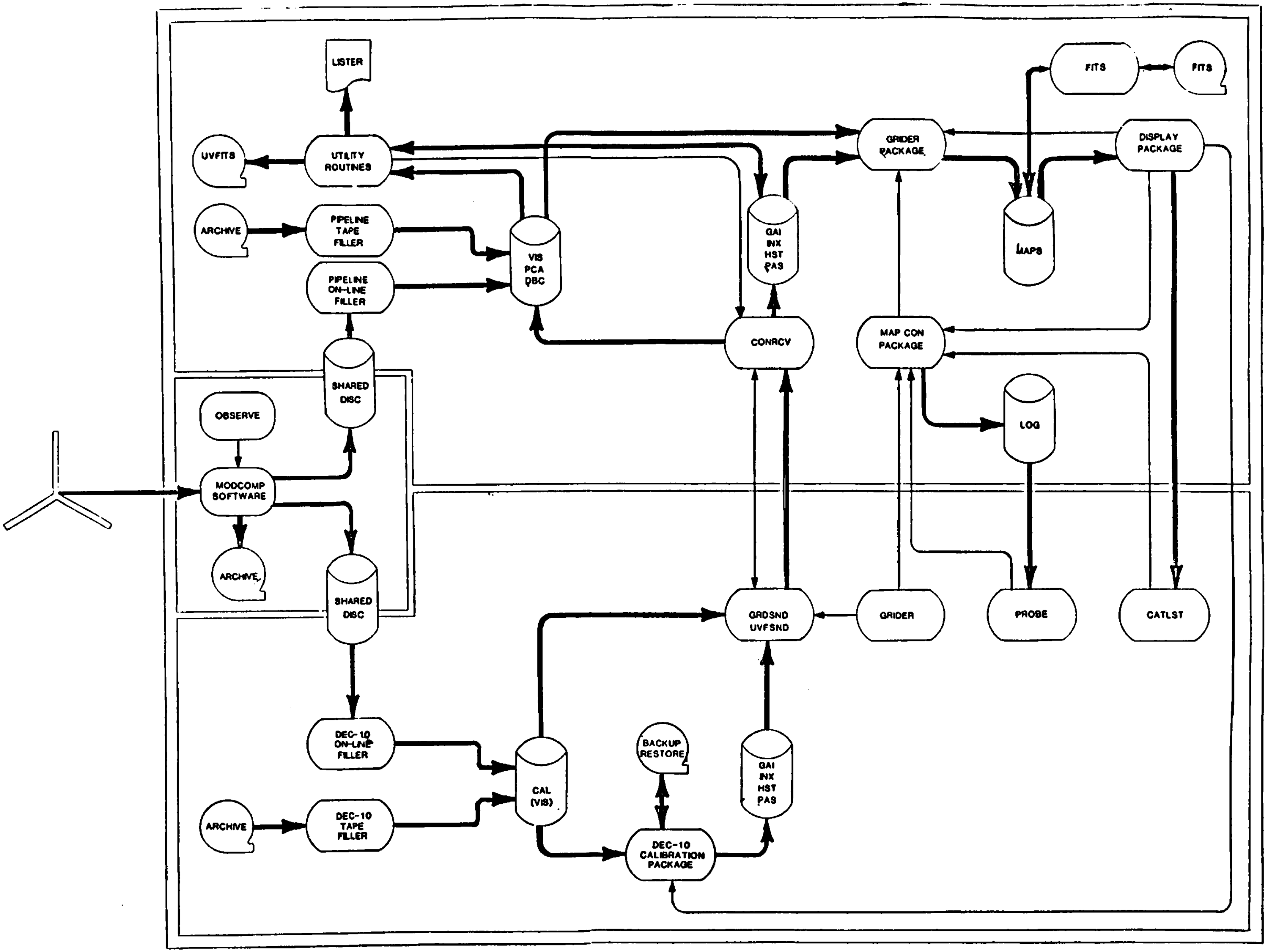
(*) Byte order is opposite in these two 16-bit machines

The contrast between the block diagram in Memorandum 137 and Figure 1 is fairly indicative of what has happened to the pipeline concept. While the earlier diagram contained 7 blocks, the current diagram contains 30.



AP0
64K
WORDS
(36-BIT)
AP1
192K
WORDS
(36-BIT)
AP2
64K
WORDS
(36-BIT)
XP
4M
WORDS
(24-BIT)
AP3
64K
WORDS
(36-BIT)
PEN AND
ELECTRO-
PLOTTERS
I²S
IMAGE
SYSTEM
DICOMED
FILM
RECORDER
11/70
640K WORDS
(16-BIT)
11/70
640K WORDS
(16-BIT)
11/44
384K WORDS
(16-BIT)
TO AIPS
1000
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
250
MBYTE
DISK
1750
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
500
MBYTE
DISK
500
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
500
MBYTE
DISK
LARGE
VOLUME
OF DATA
ON-LINE
COMPUTER
SYSTEM
FROM
CORRELATOR
LIMITED
VOLUME
OF DATA
30 MBYTE
SHARED
FIXED
HEAD
DISK
MODCOMP
INTERFACE
DN-20
FRONT END
DEC-10 KL
768K WORDS
(36-BIT)
2 DIAL-IN
PORTS
1600
1280
MBYTE
DISK
(4)
800/600
TAPE
DRIVES
46 LOCAL
CRT
TERMINALS

Figure 1



[redacted]gur[redacted]

The magnitude of the problem of maintaining Pipeline hardware can be inferred from the mix of vendors listed in Table 3. There are actually more vendors listed in Table 3 than there were blocks in the original diagram. All of the CPU's (except the Modcomps) are under contract for maintenance by DEC but this does little to solve the problem. Whenever anything goes wrong, a substantial amount of the VLA's own very limited resources (especially system programming time) must be expended on problem determination before the provisions of that contract can be invoked.

Appearances to the contrary, the Pipeline hardware is NOT the result of trying to see how many vendors could be represented in a given system. Each purchase was dictated by the unavailability of a counterpart from the original computer vendor (DEC) or by funding inadequate to the purpose. For example, DEC did not supply a 6250bpi tape drive for PDP 11 machines until quite recently and the disk space required by the pipeline would have cost twice as much had it been obtained from DEC.

The data and control paths through this hardware are represented by Figure 2. It is important to note that, with just one exception, DECNET is used only to pass control information through the system. The transfer of data between the online system and the pipeline, between the pipeline shared disks and their hosts, between minicomputers and array processors, between array processors and transpose memory, are all mediated by special "handlers" that had to be written as part of the pipeline software effort.

The transfer of data from the DEC10 to the PDP 11's (via DECNET) was not included in the original pipeline design - it grew out of the need to debug the Pipeline code before pipeline databases were available. Users discovered that they too could bypass construction of a pipeline database by inserting in the pipeline system data from the DEC10. Users found this mode of operation to be convenient because they could massage their data on the DEC10 before passing it to the pipeline for mapping or for export to the AIPS system. Thus a "temporary" kludge became an important part of the system. The price for this adaptation was a great increase in system complexity, a data storage and communications load that was never intended or provided for on the DEC10 and, because the kludge was available, a premature use of the pipeline which led to unwarranted disillusionment with the system. All of these costs seem to have gone unnoticed -certainly they never triggered a reevaluation of the pipeline system.

As originally conceived the pipeline had a single input (the online system) and a single output (the completed maps). As it is currently implimented the pipeline has three forms of input and four types of output (see Table 4). That means 12 different modes of use or, to put it another way, an order of magnitude greater complexity. Even that does not tell the full story because the organization of the input stream differs according to the data type. For example, spectral line data from the online system must undergo two stages of sorting whereas continuum data can be pigeonholed directly.

Table 4
Pipeline Input and Output

INPUT	OUTPUT
Online system in real time	Ordered raw data via shared disk
Online System buffered by tape	Calibrated data via UVFITS tapes
DEC10 via DECNET	Images to DISPLAY via shared disk
	Images to AIPS via FITS tapes

CURRENT STATUS

Early this June (1984) the Pipeline was made the default path through which VLA spectral line data would be processed in the absence of specific instructions to the contrary from the observer. This decision was prompted more by necessity than by desirability. It was anticipated that the observations scheduled for this summer would completely swamp all other computing facilities. In the sense that the current pipeline is filling an essential need, it is a success.

The pipeline has turned out to be very good at the task for which it was initially intended: making large numbers of images. It is an efficient mapping engine which can create maps 3 or 4 times faster than the only alternate system available (AIPS). It takes the pipeline approximately 8 minutes to produce a typical group of 8 spectral line maps, 512 pixels on a side, from approximately 100,000 visibility records. (This time is totally dominated by I/O.)

The ability to make maps quickly (if not in the greatest achievable refinement) is especially valuable for a "quick look" at spectral line data and snapshots where many maps are involved in a given experiment. Part of the value of the pipeline to spectral line observers comes from avoidance of cumbersome database conversions that are required on the DEC10 (eg the SPECTER program).

Many desirable features are still lacking in the pipeline. A "wishlist", compiled at the time the pipeline was made the default data path, contains four pages of one-line items some of which involve many man-months of effort. The other side of the coin is that the pipeline already provides some capability absent from other systems - for example a graphic presentation of raw data (BTMAP) in which errors can be detected more readily than by any other means. The pipeline is also the only available means of processing large (i.e. 4k by 4k) maps.

Quite apart from what the pipeline does or does not offer in its own right, it has been a success in decongesting the DEC10 computer which had become a painfully tight bottleneck for all VLA processing. This relief came in three forms: CPU requirements on the DEC10 were reduced (because spectral line data did not need to be transformed to meet the antiquated requirements of the DEC10 database structure), the crunch on disk space was reduced (because the pipeline provided additional space and used it more effectively), and I/O bottle necks were relieved (because the pipeline offers a more efficient gateway to the AIPS system and to the outside world in general).

Hindsight reveals, with its customary clarity, that the pipeline design was far too optimistic in the following expectations:

1- It overestimated the extent to which I/O between the Array Processors and their host machines could be overlapped with, hence hidden behind, computing time.

2- It did not take into account the practical limitations on massive data transfer between machines. Especially, it overlooked the effects of limited buffer space and the correspondingly small I/O block size.

3- It expected to utilize too great a percentage of the theoretical power of the Array Processor. It overlooked the problem of keeping the AP busy when only a small amount of memory was available to hold data.

As a result of these miscalculations the performance of the completed system is much less than anticipated. But loss of throughput is a relatively minor problem; a running system can be accelerated by simply adding more or faster hardware. The more significant problems are those that arise from system instability.

The fragility of the current pipeline system is a direct result of its complexity. There are too many hardware components which must all be running at the same time, there are too many elaborate database systems in which exceptional conditions must be recognized for proper data transfer, there are too many points of user access at which correct input must be supplied and there are too many user-controlled parameters which mesh into too many possible internal modes of operation. A very large part of the "wish list", referred to above, consists of measures to "ruggedize" the system but adding "fixes" to a system that is fundamentally over-complex can only increase that complexity.

LESSONS LEARNED

Three basic lessons that should be remembered while drawing up long range computer plans emerge from our examination of the pipeline experience. They are these:

1 - Requirements Grow

2 - Systems Degenerate

3 - Complexity Costs

ACKNOWLEDGEMENTS

In describing the background of the Pipeline system the writer, a relative newcommer to the VLA, has had to rely upon accounts from those who were present at the time. Their cooperation is appreciated.