VLB ARRAY MEMO No. 399

# VLBA Station Computer: Choice of CPU

B.G. Clark
October 1984

Listed below are what appeared to be the most attractive possibilities for an inexpensive station computer. Following the discussion of the individual vendors is a recomendation of the course we follow for the real-time developement of the VLBA.

The material from which this memo was compiled was gathered over the course of several months from July 1984 to the end of September 1984. In such a fast moving industry, some sections will be found to be out of date already. Rather than trying to keep the entire document current, I have not changed my initial draft of any section where the changes would not affect the conclusions drawn.

## 1. 68K based systems

For some years the Motorola 68000 series microprocessors have been hailed as the once and future king. Having no desire to spend effort enthroning this dauphin, I feel we should confine ourselves to systems that are purchasable, off the shelf, NOW. This is somewhat limiting. The currently available computers involve the 68000 and 68010 processors (the difference is primarily in memory handling, and does not impress me as being significant in our application). The 68020 is available as a processor chip (that is, orders are being accepted; deliveries are another matter), but not really as a computer. The big deal about the 68020 is that it comes with a matching chip, also by Motorola, which provides hardware floating point. This is a major plus although various systems houses sell 68K systems with other people's floating point hardware (Masscomp, Sun and Data Sud that I know of).

The reason that floating hardware is a big deal is as follows. The tasking structure of the on-line program is simplified if the full pointing calculation is done every pointing update. (The VLA, to save CPU time, does a full calculation every ten seconds, and runs on a linear extrapolation between). This appears to involve about 60 to 100 floating point calculations (if the calculation is done in floating point) and five times that number of fixed point ones. A software floating point operation will run between 100 and 200 microseconds, which is up to 20ms for the calculation, which is to be done at 50ms intervals. 40% CPU usage is unacceptable. The ways out are 1) adding an extra task (a la VLA), 2) doing the calculation in scaled binary in an assembler subroutine, or 3) doing the calculation in scaled binary in ADA (the only other high level language I know of that supports scaled binary is PL1).

68000 computers can be bought to interfaced either to a Versabus (Motorola, Force, Data Sud, and many others) or to a Multibus (MassComp, Sun, Omnibyte, and many others). Either provides access to a fairly comprehensive family of peripherals from a number of vendors.

Unlike the Intel computer, which tends to always run one of the Intel operating systems, or to run an IBM PC system, 68000s have a choice of a very large number of operating systems, ranging from real-time systems delivered on ROMs to at least a half dozen varieties of UNIX (at least two with real-time extensions). The 68000 and the operating system can be considered separately, which may lead to inclinations to buy a 68000 system from vendors other than Motorola, in order to get some other operating system well supported. There are also pure software houses that sell operating systems tailored to any of several systems houses 68000 based computers.

The vendors I have looked at to some extent are given below.

## 1.1. Masscomp

Masscomp sells a 68000 computer with a Multibus, and with their own operating system, which is Unix with real-time extensions. They do have a floating point unit, which is packaged on a separate board.

The operating system is large (about 250KBytes) and is heavily disk oriented. This makes it very difficult to meet our goals of being able to operate without a disk running, and to have a secondary means of system loading--the large size makes it undesireable to load over the telephone or to ROM the operating system (I haven't even inquired whether it is ROMable).

For the above reason I do not consider the Masscomp suitable for an on-line, unattended system. The same might extend to any Unix based system.

## 1.2. Force Sytems

Force makes Versabus systems running either the PSOS diskless real-time system or PDOS real-time system with disk. PDOS supports only Basic and Pascal as high level languages (some of the literature mentions Fortran, but it doesn't seem to be a current reality). Basic is, of course, unacceptable because of its lack of modularity (Force's Basic is a partially compiled version, which is, while inefficient, not as bad as an interpreter would be.) The Pascal is not particularly well integrated with the operating system. Although it is advertised as concurrent Pascal with sharing of objects as needed, the Pascal system includes only its own round-robin scheduler without priority, instead of interfacing to the nice, prioritized multitasking provided within the operating system. We would be forced to make use of the operating system services anyway, if necessary by writing assembler subroutines to make the system calls available to Pascal, so the "concurrent" feature of the Pascal is not useful to us.

Force does not sell a floating processor, so the discussion above implies that we would have to do the pointing routines in assembler.

All in all, Force provides a fairly nice real-time workstation, but not very rich in software tools.

### 1.3. Sun Microsystems

Sun makes a 68000 Multibus system running UNIX. The UNIX is the Berkeley 4.2 variety, which supports a full IPC networking facility. They support C and Fortran, and their UNIX supports windows, and they have a nice graphics library for monochrome or 8 color monitors that they sell. They have a floating point chip on the same board as the CPU. In sum, they appear to have everything except any real-time capability whatsoever.

### 1.4. Motorola

Motorola (the chip manufacturer) sells single board computers with either VMEbus or Versabus. They supply an operating system, VersaDOS, which seems to have reasonable real-time facilities, but rather primative multi-user facilities. The discription is very reminiscent of the Modcomp MAX operating systems. The impression I get is that the operating system may not provide everything you would like, but it is simple enough to be able to customize it without too much difficulty.

Especially, the 68020 and its recently announced single board system does make a very attractive possibility, but is likely to encounter problems·with delivery times. However, of all the 68K systems this is the most attractive possibility. The announced single board 68020 connects to a VersaBus. The VME bus is probably a more attractive alternative, however, because many manufacturers make modules that plug into the VME bus, whereas only Motorola and its liscencees make VersaBus modules. Also, the VMEbus connectors appear to be of better quality (and are recommended for hight vibration environments, for instance). However, it can be expected that a single board 68020 computer on the VME bus is not very far from announcement.

## 2. DEC systems

DEC offers several different possibilities for the station computer. In the VAX line, there is the microVAX and, to be available shortly, the microVAX 2. In the PDP 11 line, there are the two micro systems, packaged as the 11/03 for the slow one, and as 11/23, 11/24 for the fast one (there are also various board or box level items outside these packages). DEC also offers two series of "personal computers". The DEC Rainbow is a packaged PDP-8 (I do not think this curious 12 bit machine is sufficiently powerful to be a station computer, and will not discuss it further). The DEC Professional is yet another packaging of the good micro-11 processor.

## 2.1. VAX systems

The microVAX (and presumably the microVAX 2) can be bought with one of three operating systems--MicroVMS, ULTRIX (ie UNIX), or VAXELN (don't ask me how to pronounce the latter--I've never gotten a DEC person to actually say it out loud, nor have they protested any of the half dozen variations I've tried on them). Micro VMS appears to be essentially the full VMS system, a fantastic buy if you have a microVAX for data processing. A less than fantastic buy for a real-time unattended small system. It appears to contravene the requirement for being able to survive a disk crash.

VAXELN seems likely to be a good little real-time system in a few years. It does not seem to be a good one now. The features that lead me to say that are given below.

1). The only fully supported high level language is Pascal. Fortran and C can be used, but only if you either do not use, or link into the Pascal library to use, I/O or system services.

2). DECNET is supported only partially. In particular, there is no SET HOST equivalent, that is, there is no way to use the VAXELN system as a terminal on the VMS host system.

3). There is no 'load and go' facility. That is, to start a new program, one must link it into the system image, stop VAXELN, load the new system image, and start it up. The facility will be provided in the next release says DEC, in spring 85, says DEC.

Items 2 and 3 seem to me to be fatal. I therefore did not inquire what sort of real-time debugging tools are provided.

## 2.2. MicroPDP-11 systems

There are two variants of the MicroPDP-11, one with a Unibus and one with a QBus, the 11/23 and the 11/24 respectively. The QBus systems appear to be about $5K or $6K cheaper than Unibus systems, for no very obvious reason. This does make the 11/23 the item of choice. The 11/23 looks like it has about twice the speed of the old LSI 11.

The operating system of choice would be RSX-11M-Plus, with which we are at least familiar. Although it tends to be billed as a competant real time system, it has a few disadvantages over many others in such use. For example, there is no very convenient way to halt a task and then resume it from where it stopped, which is something one often desires very much to do in a real-time context. Also, the monitor console routine will cheerfully tell you that your task is waiting for something, but neglects to inform you just what it is it is waiting for.

It is interesting to note that although C has been available for PDP-11s under UNIX for many years, DEC does not supply it for RSX systems.

In short, the PDP-11 is burdened by its 16 bit address space and an operating system a little more oriented toward multi-user multi-terminal applications than toward real-time unattended operation.

Decnet is a very nice system, and would provide almost all of the facilities that one might desire. The "almost" of the preceding sentence covers several features that one might like. One is the ability to install or run a new task with no local disk (one must copy the task image to local disk and then install it to get a task into a running system). Another is the ability (present in VMS but not in RSX) to operate directly on remote data with the system standard processors.

## 2.3. DEC Professional

The DEC Professional personal computer is essentially the 11/23 in a different box. Its operating system, POS, is apparently RSX sheared of its multiuser (but not multitasking) features. There is, though, one important difference. That is that, currently, the fast floating point processor is not available for the Professional. The floating processor that one can get for the system is a processor chip and microprogramming ROM which fit on the CPU board, and has a speed intermediate between hardware and software implementations, with a maximum floating add time of 70 microseconds.

The system does have a very nice general purpose I/O board that fits into the main box and supports two terminals, a parallel port (24 bit width) and a IEEE-488 bus.

In the end, despite its extreme age, and the development of far more attractive architectures in the DEC line, the most attractive DEC computer would appear to be the 11/23.

## 3. Intel systems

Intel makes several micros, essentially 16 bit machines, which are of interest as station computers. Intel sells them as board level devices which plug into Multibus crates to build your own, or as full fledged deveolpment systems with disk, etc. They have a wide variety of simple plugin devices, most of which are not currently of interest to us, but they provide an interesting path of growth if our future needs happen to follow certain lines.

The fact that the Intel processors are being used in the IBM PCs gives one a good feeling about the financial future of the company and the future availability of the processors.

It could also be noted that our bus interface board is based on an Intel processor, the 8055 series (which have insufficient support software and marginal speed to be station computers themselves).

Intel makes several processors of interest. The 8080 series appears overall less capable than the 8086 series, at no appreciable reduction in cost. The 416 seems to be a definite offshoot from the main line of development within Intel, and we would chance being on a branch that is eventually pinched off if we go in that direction. The 8086 family appears to be the one to go with.

The 8086 family members are the 8086, the 8088, the 80186, and the 80286. The differences are primarily in the way the devices handle memory, rather than in bus widths or raw speed. The devices, running simple, small, integer oriented Pascal programs seem to be about the same speed or a little faster than 68000 based systems. The member of the family of interest to us is probably only the 80286. Only this device supports a hierarchy of priviledged/unpriviledged programs. This separation has been found to be valuable in the VLA.

The 80286 is, like the PDP-11, a sixteen bit address machine. Whether this will lead to PDP-11-like address space crunches is not clear, however. Like the small PDP-11s, the 80286 has four base registers. The usage is quite different, however. Each always addresses 64K of memory. One is (always and only) used for instruction addressing, one is used as the address for the stack instructions, and one of two (selected by a bit in the instruction) is used for the data address for other instructions. One segment register is thus available for escaping from the local 16 bit environment (however, loading it may be expensive; it is an operating system call for unpriviledged programs in a protected system, and about 20 clocks in priviledged program or unprotected system). In addition, to make this dual data segment feature really useful, the high level language processors must know about it and use it properly. The Intel C does not. I currently do not havethe information to know whether Pascal and Fortran do; some remarks about the operation of the link editor seem to imply that they do.

The Intel real-time operating system is quite a nice thing, with two levels of object directories (directories of, say, programs or data which may be needed by an executing task). It has the usual task execution control features. The operator interface/programmers workbench comes at three levels, so that only what is needed can be configured into the system. Thus we could have a program development system running the same operating system as the actual station computers, with a fairly nice set of program development tools, which do not in turn burden the actual station computers.

## 4. Modcomp

The Modcomp world is a very nice real-time system, rich enough to provide many valuable services but still simple enough to customize for the particular application. Maxnet provides networking facilities that, for the current purposes, are probably superior to those of Decnet. The rub comes from the general programming support tools. Those that Modcomp provides are unquestionably inferior to those that the DEC VAX offers. We would therefore not like to inpose the Modcomp at a system-wide control computer.

A second rub comes when we consider interfacting the system to the operator displays, which, for compatibility with the correlator system, should probably run on a VAX. The only context in which we would conceive of using the Modcomp family would be Modcomp small computers (Classic 2/15) at the stations and a larger Classic at the array control center used, not as the array control computer, but simply as a communications concentrator. This would be attractive only because the Modcomp-to-VAX interface developed for this application would also find considerable use in the VLA.

## 5. Hewlett Packard

The HP A series of micro/mini computers would be suitable for use in this application. It has a well developed HP proprietary network, a nice real-time operating system, and a reasonable set of program development tools. However, it seems to offer the same disadvantages as Modcomp, with the additional one that there is no expertise in this system among those who would be doing the work.

## 6. Recommendation

As discussed above, the leading contenders are the DEC PDP-11 running RSX-11M, the Motorola 68020 running VersaDOS, and the Intel 80286 running iRMX. The relative strengths and weaknesses of these systems are summarized below:

| Computer | OS | Network | Strengths | Weaknesses |
|---|---|---|---|---|
| DEC 11/23 | RSX | DECNET | Network | Real-time OS, 16bit addresses |
| Intel 80286 | iRMX | OSI | Real-time OS<br>Software tools | Network, 16bit addresses |
| Motorola 68020 | VDOS | OSI | Real-time OS<br>24 bit addresses | Software tools,Network<br>Delivery |

In addition to the above considerations, we believe it is likely to be an important consideration to have the same computer family, as much as possible, throughout the VLBA system.

The Correlator Group, for fairly cogent reasons, has chosen to concentrate on the Motorola 68020 chip for the fringe processors built into the correlator. For this reason, we will tentatively elect to pursue the Motorola systems for the station computer for monitor and control. This system will require a greater initial software development effort, but it is believed that this greater initial effort will be paid back in software maintenance costs through the system life.

Because of the delivery problems, we cannot order a 68020 system and expect delivery on the right timescale for the monitor and control software development. We therefore propose to buy a 68010 based software development system. The 68020 systems are expected to be directly upwardly compatible with the 68010 systems currently available. The 68010 systems themselves, without hardware floating point (and there are problems, as noted above, with the ones with hardware floating point), are only marginally fast enough to serve as a station computer. 68020 based systems are expected to be two to four times faster than 68010 based systems in integer modes. For high level language routines employing a lot of floating point (which has very great advantages indeed for software maintenance), the 68010 is likely to be slower, relative to the 68020 with floating hardware, by another factor of five to ten. The system we purchase now, however, will be indefinitely useful as a software development system, and, possibly, as a laboratory based module control and diagnostic system.

The decision above remains somewhat tentative yet, because we have not studied the Motorola VersaDos networking software to sufficient depth. If, as is still possible, it proves impossible to adapt to our needs, we shall turn to the Intel systems, as it is fairly clear that adequate facilities are available from these.