CB-Note 305

Version 1996.09

LEAR Crystal Barrel Experiment, PS197
DAQ Operation Guide

Bruce M. Barnett
Institut für Strahlen- und Kernphysik,
Rheinische Friedrich-Wilhelms-Universität Bonn,
Nuβallee 14-16,
D-53115, Bonn

September 19, 1996

Contents

1 Introduction

This document has been written for two purposes. Firstly, it is meant to provide an outline of the operation of the DAQ as a whole and furnish some detail as to the structure of the software. The hope is that the detail will be general enough to be understood by the intended audience but specific enough to give some feeling for what is going on behind the console. Such understanding one hopes breeds efficiency in the physicists who are responsible for keeping the data rolling in and ensuring that its quality is up to the standard of the detector.

The second aim is to provide some hints (and recipes) for restarting the DAQ in the case that some failure should occur. Gradually as the software becomes more *benutzer-freundlich*, (that is to say that as I and others anticipate the spectrum of error conditions which can occur, and make allowance for them) this manual becomes of less value, especially as familiarity with it will decrease in proportion to the need to refer to it. Hopefully, the detail contained herein does not hinder the reader from finding information to help him resolve the problems.

2 Hardware Configuration

The Crystal Barrel DAQ currently runs on Force 68040 based machines running OS9/V2.4. These are connected by a private ethernet and an inter-crate bus known as the VIC-8251 (Vertical InterConnect) bus (Fig 1). The VIC bus provides data collection, movement and inter-machine communication functionality. In addition, each machine is connected via a serial terminal port to a LAT server and is accessible via a port of the same name as its node. Ethernet provides TELNET and FTP services, as well as access to network disks (NFS).

Each Force CPU is located in a VME crate. The modules are single width, each having on its front panel two switches, *Reset* and *Abort*, as well as two status lights: one indicates that the CPU is in a running condition (green) or halted (red), and the other flashes to indicate bus access. The rate at which the bus access light flashes is a direct indication of the data rate, or can indicate continuous polling (x1xtal) of some device on the bus. The latter may be normal or an indication of an error, depending upon the circumstance.

The machines j1xtal, j2xtal, v3xtal and x1xtal are known as local event builders (LEVBs) as they build events for particular sub-detectors. On the other hand, e1xtal is a hybrid, as it builds part of each event locally and coordinates the operation of the LEVBS as well. It is the latter function that justifies its classification as the Global Event Builder. These machines, as does the slow-control machine, s1xtal, reside in the electronics hut downstairs outside the experimental area. The exception is v3xtal, which sits in the vertex-readout vme-crate in the experimental area.

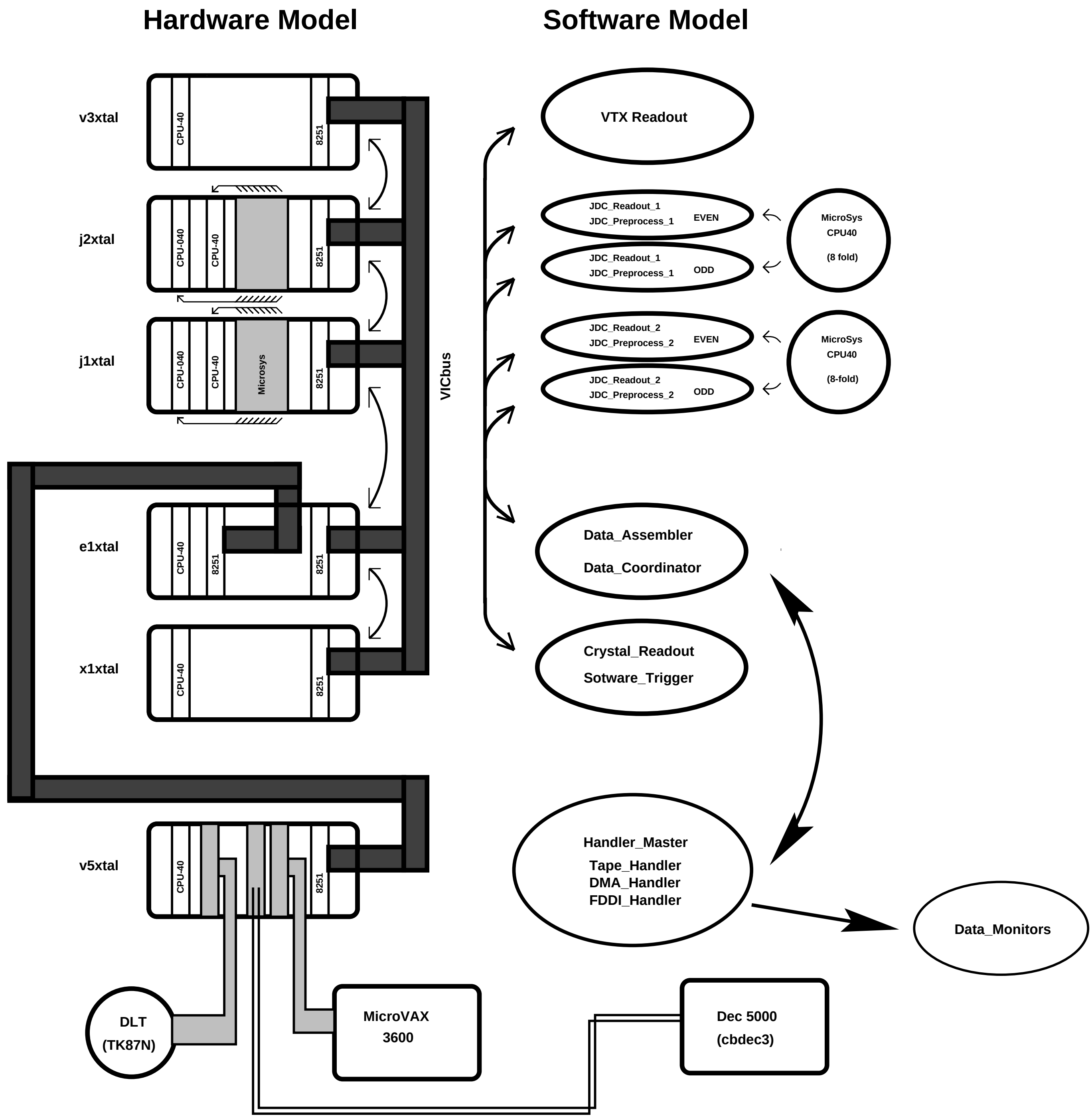

Fig. 1. Hardware-Software Model of Crystal Barrel Data Acquisition. Processes communicate via OS9 signals transmitted over the VME backplane and the VICbus. Note that the current configuration omits the double processor readout implementation in the jdc readout crate.

The last of the OS9 machines, v5xtal resides in the DMA crate (above the old cartridge tape-units). Acting as an event-server, it feeds, under operator control, events received from e1xtal, to the following devices:

- a DLT tape unit, the primary recording device.
- the VAX. Sampled events are sent to the VAX over a DR11-W (IKON) DMA interface, are subsequently buffered and delivered as requested to the monitor processes.
- cbdec3. Complete runs can be directed in parallel to the flow to tape (and not otherwise) through an FDDI interface to cbdec3. A TCP/IP daemon accepts connections, and creates a disk file containing all events from the selected run on a scratch disk (/os9_scratch/data) located on cbdec3.

In addition, of course, the Run control runs on the Macintosh in the control hut. It provides steering of the hardware and software setup, as well as basic slow control and event rate monitoring.

Details of the machines and their software components follow:

DAQ Machines

Node	Function	Running Software
x1xtal	Crystal Readout/Soft Trigger	*readout*, *trigger* or *laktrig*
e1xtal	Global Event Builder	*evb* *cebra_master*
v3xtal	VTX Local Event Builder	*vtx*
j1xtal	JDC Local Event Builder 1	*jet*
j2xtal	JDC Local Event Builder 2	*jet*
v5xtal	Tape Server / Boot Server	*tape_status* *master* *lt0_tapeio* *lt1_tapeio* *nt0_fddiio* *vaxdma*
s1xtal	Slow Control (68030)	*sc_mon* *sc_display* *hvc*
x2xtal	Development System	*development*
vsxtal	DMA DR11-W Ikon Interface (VAX)	*dma*
m1xtal	Run-Control Interface (Mac)	*CBRun*

Table 1. Process distribution over the various data acquisition machines.

3 Intercrate Synchronization

Interprocessor communication has been implemented via eight mailboxes provided by the VIC. The design specifies seven LEVBs in six crates, with one mailbox globally assigned for the private use of each crate. The mailbox communication is handled by an OS9 device driver which passes information to selected logical processes (Fig. 2) through the OS9 signal mechanism to OS9 processes with specific OS9-process-numbers determined when the receiving process opens the driver.

The implementation is consistent with an effective multitasking/multiprocessor environment and enhances flexibility over the previous shared-memory semaphores. No polling is required, and the use of signals provides an efficient mechanism for generating task switching upon demand. An additional advantage is that the interprocess and interCPU communication becomes essentially transparent once the logical process ID is defined (and of course the routing information is specified in some lower level structures and code).

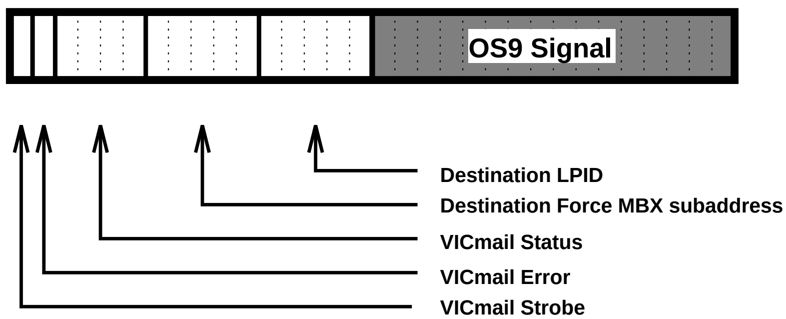

Fig. 2. Mailbox bit assignments for intercrate signal transmission. The upper word specifies the logical process ID (and slot ID, if pertinent) for the receiving process. The specific crate which receives the mailbox interrupt is determined by the memory mapping of the desired remote crate in the crate of the originator of the message.

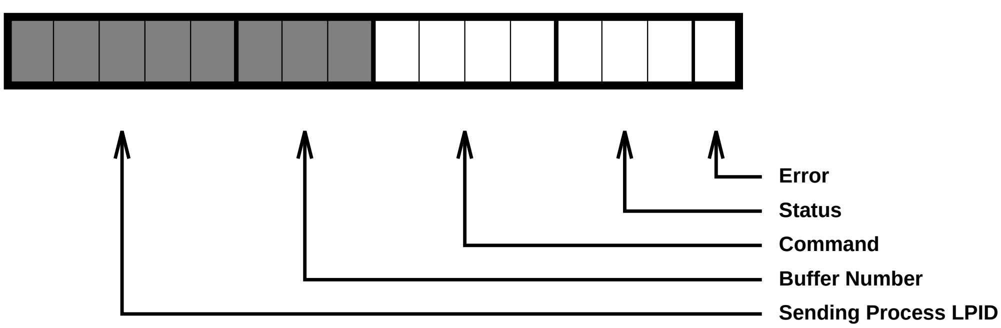

Fig. 3. Assignment of signal bits in the OS9 signal word. The upper eight bits can be masked by a “Wait” function to require a process to sleep until it receives a command pertaining to a specific buffer.

Buffer management proceeds via signalling where the OS9 signal (16 bits) is split into bit fields which specify command, status, sending process ID and buffer number (Fig. 3). The buffer and event synchronization is passed from process to process or machine to machine (Figure 4.) via the signal mechanism. In this way the buffers can be allocated, filled, processed, recorded and released in succession by relevant processes.

The signalling technique also provides for basic state management (start/stop/pause/continue, etc) between the machines.

4 Application Software

Application software is written in C, with time critical code and device drivers written in 68040 assembly language.

Each 4 MB VIC memory is mapped onto four, 4 MB ranges: one each for data, control, interrupt-service and broadcast functionality. The memory is mapped onto a different 16 MB range according to the crate in which it resides (what is local on one machine is remote to another, and so on), making the addressing appear uniform and symmetric over the multi-crate address-space.

Readout and preprocessing of one of the detector components (the JDC), which is already serviced by two independent LEVBs due to the large quantity of data generated, can benefit by the use of an additional SBC in each crate. Communication between the processors/processes in this case proceeds via mailboxes provided on the CPU-40 with the use of a driver similar to the one implemented for handling the VIC-mailboxes.

The multitasking environment allows, as well, for the introduction of optimized off-line JDC data-reduction code into the new multi-buffered environment. This decreases the volume of data generated by the JDC LEVB by a factor of five to ten and improves off-line reconstruction speed. (More recently, an upgrade of the microsys front end processors to 68040 CPU's has allowed this to be moved one level closer to the hardware in order to benefit from the resulting enhancement in that processing power.)

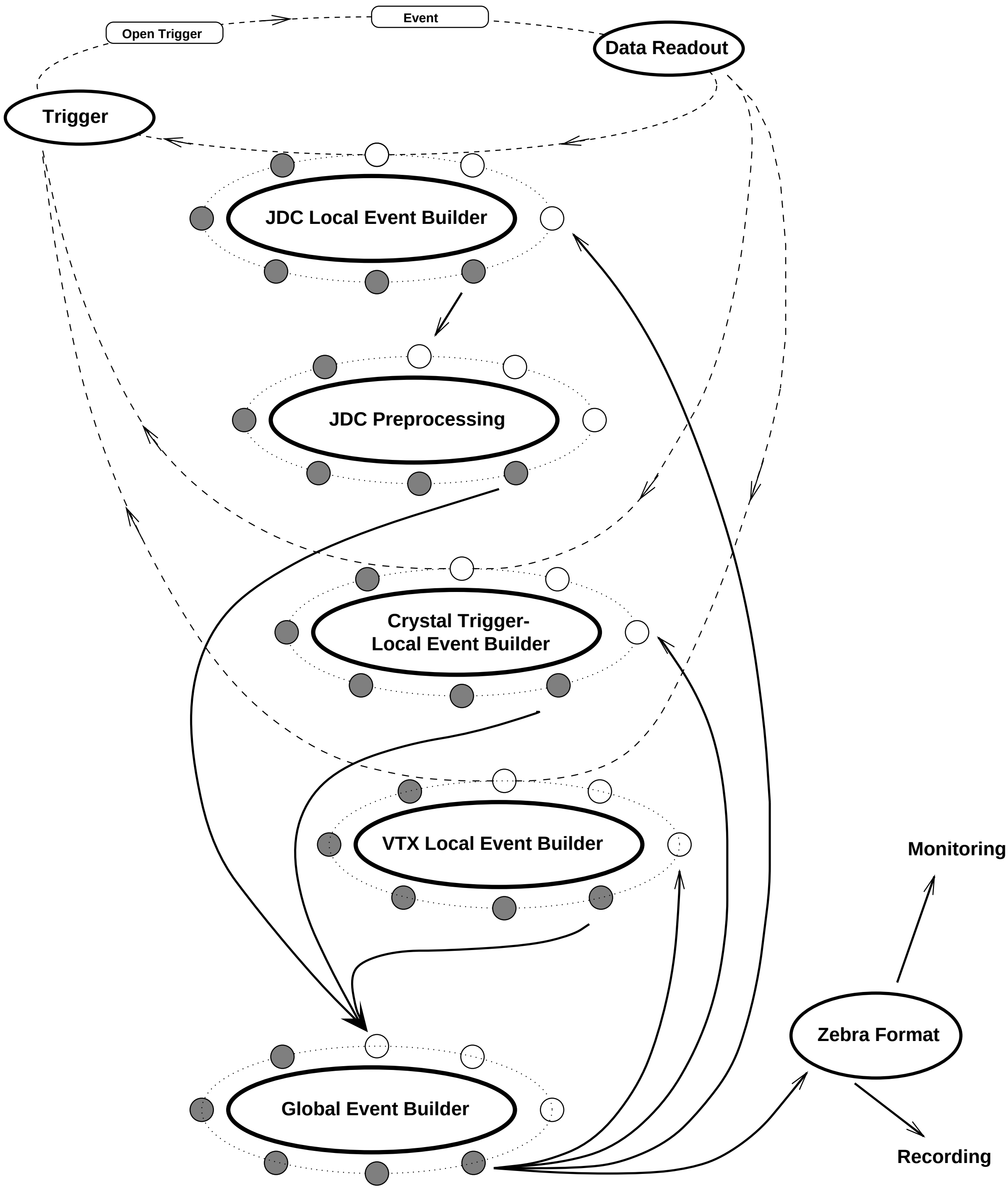

Fig. 4. Buffer Management of the Crystal Barrel Data Acquisition. Hardware synchronizes on an event-by-event basis (dashed lines), while buffer synchronization is made with signals acting as tokens for the buffers.

5 Booting

With the exception of x1xtal, which boots from a local hard disk (/h0), the OS9 machines boot via ethernet. The machines fetch their boot files (os9boot.node_name) via *tftp* from the boot server, after having determined the server and boot parameters with *bootp*. (Currently, cbdec3 acts as boot server. A change in configuration, however, allows the use of v5xtal or vsxtal as an alternate server.) The bootfiles reside on /os9_nfs/bootp/tftpboot (to which there is a symbolic link from /tftpboot). Each of the diskless machines mounts a system disk at boot time, and executes a startup file found on the device. With the exception of v5xtal and x2xtal, this is mounted as (/v0) from vsxtal, with the startup file being (/v0/BOOTP_STARTUPS/startup.node_name.) These other machines mount in addition to /v0 a system disk, /sd, from cbdec3:/os9_nfs/os9_h0, and execute, instead, the startup file (/sd/BOOTP_STARTUPS/startup.node_name.) Network parameters for the relevant machines are given below:

DAQ Network interfaces

Node	Interface	Ethernet Address	IP-Address
e1xtal	le0	00:80:42:00:33:91	192.65.186.2
j1xtal	le0	00:80:42:00:33:92	192.65.186.3
j2xtal	le0	00:80:42:00:33:93	192.65.186.4
v3xtal	le0	00:80:42:00:34:24	192.65.186.7
s1xtal	le0	00:80:42:00:33:90	192.65.186.8
x1xtal	le0	00:80:42:00:34:21	192.65.186.11
x2xtal	le0	00:80:42:00:34:20	192.65.186.13
v5xtal	le0	00:80:42:00:33:94	192.65.186.10
f-v5xtal	nt0	00:40:42:05:02:12	192.65.186.66
vsxtal	se0	AA:00:04:00:49:59	128.141.0.76
cb-vsxtal	se1	08:00:2b:09:bd:b0	192.65.186.1
cbdec3	ln0	08:00:2B:1B:D0:54	128.141.248.7
c-cbdec3	ln1	08:00:2B:1A:7E:34	192.65.186.16
f-cbdec3	fta0	08:00:2B:B0:90:11	192.65.186.65
m1xtal			192.65.186.12

Table 2. Network parameters of various data acquisition machines. Note that, in the case of the os9 machines, the ethernet number is used to define the ip-number.

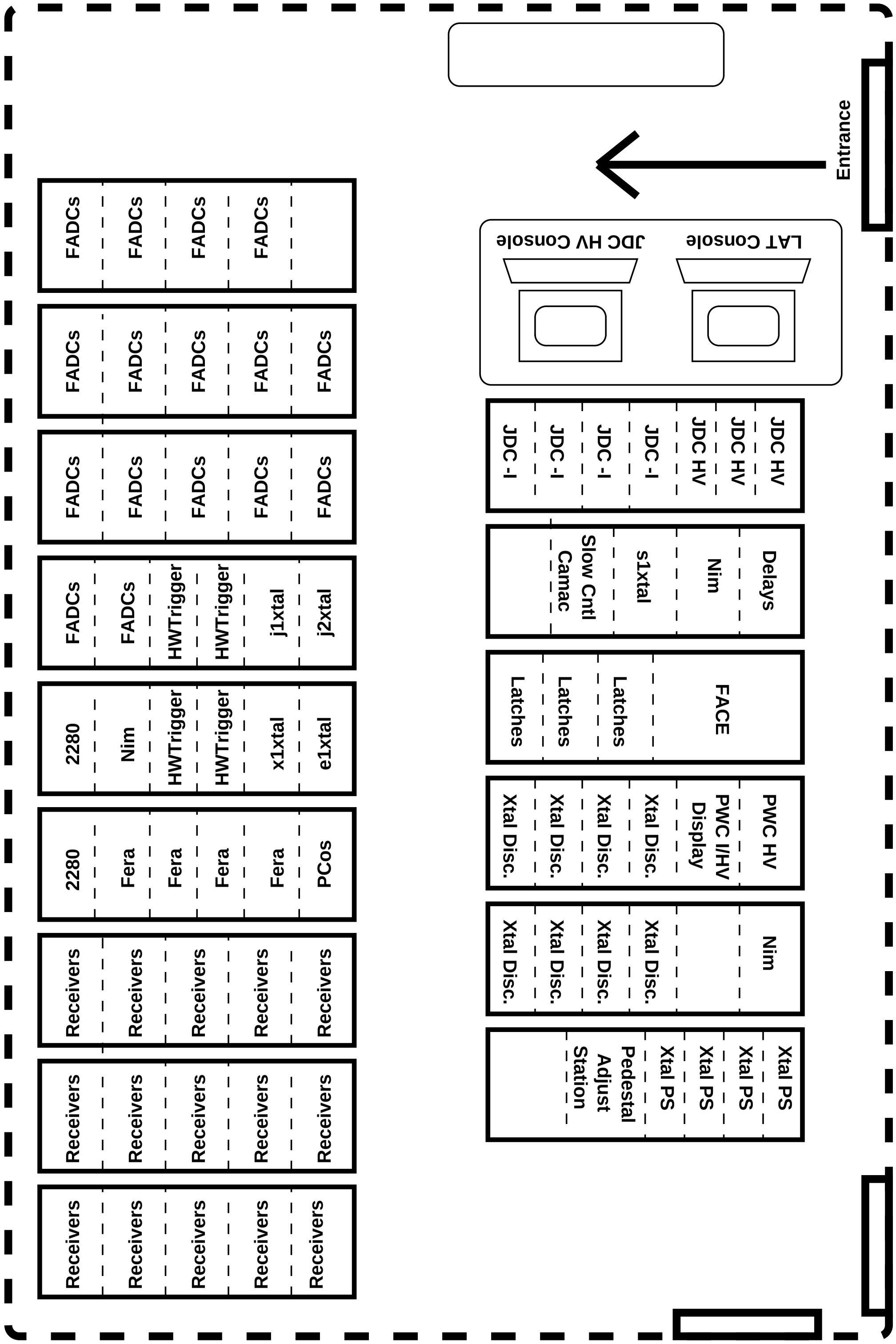

Fig. 5. Rack Configuration in the Electronics Hut.

The console ports for each event builder (x1xtal, e1xtal, j1xtal, j2xtal v3xtal, and v5xtal) are supposed to be connected via LAT to the 19" NDC Network terminal (Select Menu *Terminal* and sub-menu *New LAT*). Should a machine be reset (with the Mac Menu, the front panel switch labelled *Reset* or by typing *reboot* from the command line) it should boot automatically, displaying status messages on the console.

6 Failure to Boot

If a machine fails to do boot, a special reset, which downloads information into the onboard gate array controlling many of the machine's functions, may be helpful:

- Note the two switches on the front panel of the CPU with which you have problems. The top one is labelled *Reset*, and the other is labelled *Abort*.
  1. Hold the top one, *Reset*, up.
  2. Hold the bottom one, *Abort*, up.
  3. Release the top one, *Reset*.
  4. Release the bottom one, *Abort*.
- The gate array should by this have been freshly loaded.
- Lift the top switch *Reset* and then release it.
- The machine should boot.
- When the machine is up, verify that the time and date are correct. The time is normally adjusted at boot time via a request to an *ntp* time server at CERN. If the clock is out by more than some minutes, the update is refused. Should this be the case, use the OS9 command *setime*, to set the time approximately and the OS9 command *ntpdate time-1* to adjust it to synchronism with the other onboard clocks in the system.
- If the machine does not boot, but instead enters a configuration menu, call an expert.

Watch the display as the machine boots. If the output ceases before you are invited to log on, type RETURN and wait a little longer. If this fails, and the output has hung, or if the boot fails repeatedly (many error messages about exception vector 60), try once again. If a second failure occurs, it may be that one of the modules in the crate has locked the bus and the CPU cannot access it. In such cases, which occur infrequently, try the following sequence:

- control-c on the *evb* process if it is running.

- Power down the crate containing the offending CPU.
- Wait 10 Seconds.
- Power up the crate.
- The machine should automatically boot.

Important: Normally several machines may boot simultaneously. A more cautious approach is to wait for a given machine to complete the boot sequence before initiating the next. If the machine fails to boot, and displays a prompt *cbgo:* , the network disk server (vsxtal) may be refusing additional mounts. See the section entitled *Network Disks.*

If a crate has been turned off for some time, and the nicad battery which powers a battery backed-up ram on the force cpu-040, in which is stored certain vital parameters (such as the board's ethernet address), is old or failing, it may be that a cpu fails to boot properly. Then examination of the bootp broadcast, displayed on the system console upon boot, may prove helpful.

In the following case, for example, the host-specific parts of v5xtal's ethernet address have been lost:

```
inetboot: start 2 sec timer test
inetboot: timer test finished
inetboot: MTU=1514
bootp: 0:80:42:0:0:0 broadcasting for server
bootp: retry with timeout=0
bootp: 0:80:42:0:0:0 broadcasting for server
```

This message will repeat indefinitely, with various timeout values. In this case, it is best to call an expert. If that is not possible, execute a "special reset" as outlined earlier in this section. NOW, FIRST!!! Note both switch settings on the front panel of the problematic CPU. The direction of the arrow on the switch determines the setting. Next, using a jeweler's screw-driver of the proper size, set the top rotary-switch on the front panel of the CPU (labelled switch 2) to "A" and the bottom one to "0". Reset the CPU, with the front panel switch. Replace the switches to their original values. On the console should now be displayed:

BOOTING PROCEDURES AVAILABLE —– < *INPUT* >

Boot from QIC tape drive ——— < *vs* >
Boot from Teac SCSI tape drive — < *ts* >
Boot from floppy drive ———– < *fd* >
Boot from Teac SCSI floppy drive - < *fs* >
Boot from SCSI (SCCS) hard drive - < *hs* >
Boot from BOOTP am7990 LANCE —– < *le* >
Boot from BOOTP am7990 EAGLE2 —- < *la* >
Boot from ROM ——————– < *ro* >
Boot from RAMDISK —————- < *rd* >
Reconfigure the boot system —— < *rc* >
Restart the system ————— < *q* >

Select a boot method from the above menu:

Reply with *rc* to reconfigure the system.

FORCE COMPUTERS GmbH CPU-40 OS-9 V2.4 Booter V2.01

BOOT SYSTEM RECONFIGURATION

Do you want the debugger enabled? (y/n/q) →Type n
Do you want the boot menu enabled? (y/n/q) →Type n
Do you want to enter the Ethernet Address for this board? (y/n/q) →Type y
The current Ethernet Address is 008042000000
Enter the last six digits of your Ethernet Address

Enter the correct value here. The ethernet id's are listed in Table 2 of this manual.

Boot drivers available:

1 - Boot from QIC tape drive
2 - Boot from Teac SCSI tape drive
3 - Boot from floppy drive
4 - Boot from Teac SCSI floppy drive
5 - Boot from SCSI (SCCS) hard drive
6 - Boot from BOOTP am7990 LANCE
7 - Boot from BOOTP am7990 EAGLE2
8 - Boot from ROM
9 - Boot from RAMDISK

The priority of these boot drivers can be set below and will remain
set even when power to your system is turned off. This priority
determines the order that the boot drivers will be selected when your
system is configured to boot automatically. It also determines the
order that they appear in the "boot driver menu" as well.

The current system boot driver priority is: 123456789

Enter the numbers of the boot drivers in the order
that you want them prioritized.

All machines which network-boot require only 5. Enter 6 for x1xtal, as that machine boots from its own hard disk. Alternatively, one may enter combinations like 56, but that is most useful only when the *boot menu* has been enabled.

This is the order in which you have prioritized your boot drivers:

Boot from BOOTP am7990 LANCE
Restart the system

The debugger is disabled.

The boot driver menu is disabled.
(The system will autoboot according to the above prioritization.)
The Ethernet Address of this system is 008042003394

Is this the configuration you want? (y/n/q)
(y)es will reconfigure and restart the system.
(n)o will restart this reconfiguration program.
(q)uit will return you to the boot driver menu.

Type y for yes. Next the standard, correct, boot sequence should appear. What follows is typical of v5xtal:

```
OS-9/68K System Bootstrap
inetboot: start 2 sec timer test
inetboot: timer test finished
inetboot: MTU=1514
bootp: 0:80:42:0:33:94 broadcasting for server
GOT BOOTP RESPONSE: Use IP address 192.65.186.10!
My IP address will be: 192.65.186.10
My tftp bootfile is: /tftpboot/os9boot.v5xtal
My bootfile size is: 1395 (512-byte) blocks
My subnet mask is: 255.255.0.0
Network time offset is: 18000
tftp: fetch 1395 blocks
tftp: server response 192.65.186.16 port 3162
tftp: expected 1395: received 1394:160 blocks (697Kb)
A valid OS-9 bootfile was found.
Ethernet Address: 080:420:3394:
executing isp startup ...
+4
Trying to mount c-cbdec3:/os9_nfs/os9_h0 as /sd
-t
chd /r0
tmode -w=1 nopause
```

From here the execution of the host specific startup file should proceed.

7 Network Disks

All the OS9 machines use NFS disks (/v0, /cb1, and /usr) heavily. These are served from the VAX, which maintains the disk space on the VAX user disk, sys$user (currently DUB0:). In addition, v5xtal and x2xtal mount an NFS disk (/sd) from cbdec3:/os9_nfs/os9_h0.

The disks /v0 and /cb1 are owned by VAX user OS9 (sys$user:[os9.nfs] contains the directories which are mounted on /v0 and /cb1). The OS9 super user is mapped to unix user (os9_root,os9_sys) (256,256), which is in turn mapped onto the vax user [OS9]. This choice allows OS9 to have super user privileges over its own disks (it uses only a byte for the group and user numbers, and hence interprets (256,256) as (0,0)) while not requiring unix machines to allow it *root* privilege over those disks. The /usr disk is mounted from sys$user:[os9.usr]. Each directory entry in /usr/users is owned by a particular os9 user, who is mapped onto the corresponding vax user, from whose disk space the user directories are allocated. The directory /usr/os9sys is owned by [OS9] (ie super-user).

The disk /sd is owned directly by unix user os9_root.

In any case, if too many stale mounts from a given host exist, the server may refuse additional mounts. Symptoms are the failure of machines to boot properly (*cbgo:* prompt), or random disk access errors on the VME-machines. The vax server can be restarted as follows:

- log on to the VAX system account (if you do not know what the password is, maybe you should call an expert!)
- type: multinet configure /nfs
- type: restart
- type: quit
- logout of the system account. DO NOT FORGET THIS LAST STEP.

8 Logging in to a Force CPU

Inspite of some privileged hardware and address space accesses which make it awkward for many of the DAQ processes to run under any but the super user account, we are now able to run them under the CB account.

When you type return from the LAT console (or telnet in, for that matter), prompts are given. The response is shown in italics, and should be followed by a carriage return ($< cr >$).

Username?: *cb* $< cr >$
Password: ____________ $< cr >$

9 Cold Start

The instructions in this section assume that you are starting from scratch or wish to do so. Perhaps *nothing* is working. Here is what you can do.

- First, go down to the electronics hut and verify that all the crates are powered on.

- Check that all flash-adc (FADC) crates are on, and that the FADC status lights are green (jdc-readout: there are 3 per FADC module).
- Check that the trigger crates in the area are functional, or at least powered up.
- Next, shutdown the DAQ and reboot:

  1. Stop the run first, if the Mac is not indicating a run stopped status.
  2. Quit the Macintosh *CBRun* if it is running.
  3. Press control-y on the VAX DMA monitor.
  4. Using the front panel switch *Reset*, reboot v5xtal
  5. Wait until you are invited to login: *Hit return to login*
  6. For each of the remaining machines (x1xtal, j1xtal, j2xtal v3xtal, and e1xtal) in sequence, use the front panel switch to reboot. In each case, wait until the machine invites you to login.

- Finally Restart the DAQ processes:

  1. check that *sc_mon* is running on s1xtal.
  2. on x1xtal: type *readout*, or *trigger* a software trigger is in use.
  3. on j1xtal: type *jet*
  4. on j2xtal: type *jet*
  5. on v3xtal: type *vtx*
  6. on v5xtal: type *tape_status*
  7. on vsxtal: type *dma*
  8. on e1xtal: type *evb*
  9. on m1xtal: select *CBRun*

10 Warm Start

More often, some particular processes may have hung or crashed. Check, with a control-e, that all processes are responding. If not, some status information may be shown on the *V5XTAL Status Monitor* which can be of help in understanding the situation.

1. Stop the run first if the Mac is not indicating a run stopped status.
2. control-c on the *evb* process.

3. If a second control-c is required to stop the *evb*, then probably the *cebra_master* has died or hung. In this case, control-c on *tape_status*. (Never exit *tape_status* when *evb* is running). Check that none of the processes: *lt0_tapeio*, *lt1_tapeio*, *nt0_tapeio*, *master* or *vaxdma* is still running on v5xtal. This can be done with the *procs -e* command; the processes may be killed, if necessary, with the *killproc* command.

4. Exit all DAQ processes which appear abnormal. Abnormal more or less means that control-e fails to produce some response. Control-c should exit such a process. If that does not work, telnet into the machine and use *killproc* to kill the process. Then logout both the telnet session and the console connection, logging in again into the console session.

5. Note that if some process has exited in the initialisation phase at startup (*jet*, for example), there may be a hardware problem. Check that no Microsys CPU has come into a halt state (red status led on front panel).

6. Restart the DAQ processes which you have killed or have died. The order does not matter, with the exception mentioned above (avoid killing *tape_status* before killing *evb*.) (Remember to click on the left mouse button first, to activate the window before typing control-c.) Should a process not exit within 20 seconds, try again. If this does not work, telnet in from another terminal and use the command *killproc* to kill all DAQ processes (listed in the table in an above section) on the particular machine. If you are not able to log into the machine or get a response, reboot the machine.

7. Start up the Macintosh. Select the *CBRun* program from the CBRun folder.

If the DAQ has hung and you do not know why:

- Stop the run. WAIT 30 seconds, or so, until the v5xtal status monitor indicates "cebra-master found".
- If you are using a newly mounted tape, CHECK THAT THE TAPE NUMBER CORRESPONDS TO THAT ON THE TAPE LABEL.
- Start a new run on the Macintosh.
- If after some patience, the run still has not started:
  1. Stop the run on the mac.
  2. killproc *evb* (on e1xtal).
  3. control-c on *tape_status* (on v5xtal).
  4. restart *tape_status* and *evb*.
  5. try again.
- If after several tries you can still not bring up the DAQ, you can try the COLD start procedure. At this point, you might in any case consider calling an expert, as it is likely to be the faster solution.

11 Software Trigger

The operation of the software trigger is outlined in the manual by N.P. Hessey. Appropriate trigger files are selected from the Macintosh by entering the software trigger file name in the Macintosh trigger file (which specifies the hardware trigger and other setup information), Specification of an unknown file name will force *trigger* to exit its initialisation phase without the normal response to the *evb* which would otherwise signal completion. The run start sequence does not in this case continue: the solution is to stop the run and restart it with a correctly specified software trigger file.

If you have tried to start the *trigger* program, and the program exits with the message: "No Software Trigger Decision required, use readout", then understand that *readout* is much faster at startup than trigger, and should be used when no software decision is demanded. To restore the functional trigger, then from the x1xtal console, type:

- chd /h0/CMDS
- copy -r trigger trigger.sv
- copy -r trigger.last trigger

12 Vertex Readout

The VerteX-microstrip-Detector (VTX) readout provides readout as well as pedestal and threshold management for the CAEN CRAM (V550, viking readout) modules, the backplane amplitude (V556 or V456) adcs, and the V258 (backplane) discriminator unit. In addition to readout, software datacompression and hit finding is performed for the V550's before data assembly.

Options for the readout are specified in a file on /cb1/pro/CONFIG, vtx_config.dat. Here pedestal subtraction and zerosuppression may be turned on or off, backplane adc reset mode may be specified, and dead-strips may be flagged. This file is only read at program startup, so any changes are not reflected until that time. IE: to invoke alterations, *vtx* MUST be stopped and restarted!

13 Microsys CPU's

The JDC readout is based on two master local event-builder CPU's (j1xtal and j2xtal) each of which coordinates and fetches data from 8 CPUs known as Front End Processors (FEPS), the *Microsys*. These 68040 machines run OS9 as well, reading out the FADCs upon interrupt request and communicating their data-ready conditions to the LEVBs via mail-box interrupts.

The Microsys boot directly from battery backed up ram-disks, loaded with OS9, sitting in the LEVB crates. These ram-disks are normally write protected, but may become corrupted if this is not the case or if the crates are powered down for a longer time. The current running version of the software may be loaded with the command:

/cb1/new/SOURCES/JDC/MICROSYS/loadup_r1.

The current version of the operating system files are loaded with the command:

/v0/USER/MICROSYS_041/init.r1

(executed from j1xtal or j2xtal). This is, however, recommended to be done only with the advice of the person responsible.

The Microsys normally only make their presence felt to the operator during their pedestal measurement cycle, which occurs at the end of each run, or under some circumstances at the beginning. When a new run is started, the LEVB checks that the pedestal measurment has occurred within the last 10 minutes. If that is not the case, the pedestals are declared *stale* (like bread which has been sitting on a shelf too long!) and remeasured. Pedestals which are *not stale* are considered to be correct.

Note that during the pedestal measuring cycle, a report is given as to the number of pedestals that are *not set* or *not within sigma*. Consistent complaints of this nature should be reported. Also, if the JDC status display shows large numbers of *Long Rams*, these should be reported, as well. A further indication of improperly set pedestals or noisy modules is a large variation in the number of events per tape for a given hardware trigger. Such inconsistencies should also be reported in the error log.

The pedestal measurement cycle normally fails 1 or 2 times after calorimeter pedestal and/or light pulser runs, or if a run is terminated with a stalled trigger (no incoming beam). This is due to a hardware flaw in the triggering of the FADCs, and is to be expected. Otherwise, pedestal measurement should complete within 20 or 30 seconds. If pedestal measurement fails repeatedly then it is likely that an FADC fuse has blown. Check the offending crate (there is a mask displayed indicating the completion of the measurement cycle for each FEP, as the cycle proceeds) for a missing fuse status light on some FEP, and call an expert.

IMPORTANT. The FEP measurement normally proceeds at the END of a run, in order to reduce operator-dead-time. To ensure that pedestals are measured anew at the beginning of an antiproton spill, the time since the last pedestal measurement is monitored. Do not start a run in a new spill before the beam is present and at a significant fraction of the maximum rate achieved during the spill. Otherwise the pedestal values will be biased by the lack of beam in the JDC. Remeasurement of the pedestals can only be manually forced by the exit (control-c on *jet*, but not while *evb* is running!) and restart of *jet*.

If for some reason you think that the microsys need to be reset, think again. It is unlikely. This is because the CPU's are booted automatically when several pedestal-measurement cycles have failed, and when *jet* is started up. If the *evb* has exited with a failure status indicating problems with the microsys, however, or if for some other reason you think the microsys may not be running correctly, then you may try the following:

1. First, stop the run.
2. Then use control-c to exit from *jet*. Go Downstairs and check the microsys halt lights. Record which, if any, are halted. Try to reset the offending CPU's with the reset switch on the offending CPU(s). The Microsys should boot, its (their) light(s) being switched off in the process. Resetting the right-most Microsys will force the reboot of all of the 8.
3. If you are unable to bring the microsys back into a running state, a crate hard-reset may be necessary:
   - Switch off the crate using the green switch at the bottom of the crate.
   - Wait 10-15 seconds, so that the crate power down is clean.
   - Switch the crate back on.
   - All microsys should now boot.
4. If problems persist, then likely there is a problem with one of the modules in the flash-adc crates. Each Flash ADC has 3 green voltage status lights. Check that all such modules in all 16 crates are on. Then call the expert (Karsten) and report what you have found.
5. Otherwise, go back upstairs and restart *jet* on the appropriate machine.

14 Control of Recording Media

All tape control is implemented on the Macintosh. The idea is that the start of run menu displays the current recording status. If this shows a tape number, that tape will be used to record the data, providing the tape is mounted. *Disk* and *Null* options are also implemented. (See the section on *VAX DMA*.)

To mount a tape, select the TAPE button on the start of run menu and enable the streams required:

- VME must be enabled to record to tape or to "echo" the data through the FDDI link onto the bicycle-offline scratch disk.
- VAX must be enabled if the vax monitors are to recieve sampled events and/or light pulser runs are to be written to VAX disk for immediate analysis.

Most important, if you are writing to tape, enter the tape NUMBER. Notice that there is only one tape-unit supported now, but also that some of the software continues to regard the "DLT" as the "Top Left DLT".

Upon start of run, the software will attempt to mount the tape on the specified drive:

- If the tape is write-protected, it will be unloaded repeatedly, until a tape is placed in which is write-permitted. (Status message)
- If the tape is already mounted, it will continue with it.
- If it cannot find the volume, it will hang, with the green vme-access light (on the CPU) burning brightly. This likely means that you have forgotten to enter the correct tape number at the start of run. In this case, stop the run and start a new one, being more careful this time.
- If during this procedure an unlabelled tape is found, it should be labellel according to what you have specified. This will only work if the os9-tape-table indicates that the tape with the desired label is software-write enabled.

The tape will, when full, dismount automatically. If you have decided to rewrite a tape or to fill one incompletely, then at the end of run select the *Special* Menu, *VME Tape*. With this menu click on Dismount, specifying the volume number. Note that this menu is only enabled at the end of run, before the normal tape setup has been accessed, and can only dismount a tape that is logically mounted (and has been written to). Control-c in *tape_status* will force the exit of the tape processes, the *vaxdma* and the *master* process. This should only be done if the *evb* has first been exited. The tapes will be ejected after a logical end-of-file and end of tape is written after the last zebra structure on tape.

In normal *production* running, the end of tape is reached and the Macintosh is flagged to stop the run. The operator is warned with a *requestor* box, which he acknowledges by clicking on *ok*. The tape is dismounted automatically. Upon the start of the next run, the *Tape* select menu is automatically entered. You will notice that the expected tape number and unit are updated in the menu window to be the next expected tape and unit. If the selection is correct, then you need only click on *ok*.

When the DLT tape has been rewound, the green ready light will turn on. Lift up gently but firmly on the unload handle, and the tape should eject. Do not be in a big hurry about this. Some drives are slower to eject the tape then others and pulling on the tape before it is fully dismounted can jam the drive.

Be CAREFUL when you mount a new tape that there is nothing on the bottom of the cartridge (like a do-it-yourself label) that can jam the toothed spool.

15 DLT Tape Drive

Introduction of the DLT drives into the data acquisition in 1995 resulted in much smoother operation of the DAQ as a whole, due to the greater reliability of, and a reduced interaction on the part of of the shift crews with, the devices.

In order to ensure ease of data handling and greater security in the stored data, tape file-header management has been introduced into the system. Each tape is labelled but now standard ANSI labels are generated in ASCII, not EBCDIC as before. These labels comprise an 80 byte VOL1 header which is of the standard form, with the exception of the block size (23040 bytes) which exceeds the maximum allowed by the standard. Individual runs are sandwiched between file-headers, consisting of two 80 byte records (HDR1 and HDR2) at the beginning of a run and two matching records (EOF1 and EOF2) at the end of the run. A file name of the form Runxxxxx.rdt, where xxxxxx is the number of the run, is contained in HDR and EOF records. In addition, the entries in EOF correctly reflect the number of bytes and number of records written (clearly, HDR cannot). Each EOF1 is proceeded by a (logical) tape mark. Each EOF2 and HDR2 is followed by a (logical) tape mark.

A few notes on handling exceptional conditions with the DLT drives may prove helpful.

- If the *cleaning light* on the on the DLT drive front panel, simply feed in the cleaning tape, allow the cleaning cycle to complete and remove it when the green ready light reappears. For each additional use, indicate such by marking in the space provided on the cleaning tape.
- Do not, in general, use the manual unload button. If for some reason, you must manually unload a tape, try the Macintosh *Special* menu, first. Otherwise, proceed as follows:
  1. Stop the run, if running
  2. control-c on the *evb*
  3. control-c on *tape_status*.
  4. press the unload button. The same effect may be accomplished through software on v5xtal: *tape -o /lt0*.
  5. restart *tape_status*
  6. restart *evb*.
- If there has been a tape error, indicated by flashing lights on the front of the drive:
  1. Check the *V5XTAL Status Display* for abnormal conditions (hardware error, or write error).
  2. control-c on the *evb*: a second time if there has been no exit after 20 seconds.
  3. Reset the tape drive by turning it off, waiting 15 seconds and then on again.
  4. Boot v5xtal, by using the *Reset* switch on the front of that machine. Booting will take about 45 seconds.

5. login to v5xtal and type *tape_status*
6. restart the *evb* on e1xtal

16 V5XTAL Status Display

A status display runs on v5xtal to provide a sequential indication of the operation of various of the DAQ processes. The process runs in the background on v5xtal, and normally requires no attention. It receives tcp messages from various processes on various cpus, or via the *notify* command available from the OS9 shell. (The syntax for that command is: *notify code error_message*, where code is a numeric status code.) Informational and error messages are displayed here from time to time.

17 Tape Status

There is a VME Tape Status display which runs on v5xtal. This shows much information about the tape unit status, current label, and operation. *Blocks Left* tell you how many blocks until end of device: it also ensures that blocks are being written to tape, if that has been selected.

IMPORTANT: IF THE *BLOCKS LEFT* ENTRY IS NOT CHANGING THEN EITHER THE TERMINAL OUTPUT IS HUNG (Try clearing and resetting the Terminal from its command menu) OR THE DATA IS NOT BEING WRITTEN TO TAPE!!! Data flow to tape is normally signalled at the start of run on the *V5XTAL Status Display.*

The lower half of the display refers to data passed to the VAX for monitoring. *Modulo*, which can be altered from the Mac Tape Menu, tells you that 1 in Modulo events are selected to be sent to the VAX. The select counter tells how many events have been sent to the VAX and so on. The record counter can be less than the block counter if Zebra fast blocks are being generated. NB: The VAX transfer must be enabled from the Macintosh for this display to be updated.

Much information about the proper operation of the tape master is on this menu and it should be monitored from time to time and especially at the beginning of run.

Update: The run start and stop times as well as the number of data events sent to tape are recorded and remain until a new run is started. The active tape volume number and block-left entries are highlighted. A beep signals the start and end of each run.

18 Window Terminal

This should normally contain terminal sessions connected to the 19" NDC Network terminal on the following ports:

LAT ports:	j1xtal	j2xtal	v3xtal	e1xtal	x1xtal	v5xtal	vsxtal
User Name:	cb	cb	cb	cb	cb	cb	model
processes:	*jet*	*jet*	*vtx*	*evb*	*readout*	*tape_status*	*dma*

After you connect to the LAT terminal, the prompt as well as the X-terminal icon and window-titles are set automatically on login. This with the exception of the *dma* window. In this latter case, the command *set_title* produces the desired labelling.

If you need to reset this terminal, Use the *Quit* menu on the terminal manager. Alternatively, power the terminal down, wait 10 seconds, and power it back up again. (The switch is on the right side of the monitor at the front). The terminal will reboot after some minutes.

To create new LAT sessions, find the NCD console (click the left button on some unused space of the terminal backgound if you cannot find it, select *Show Console*) and select the menu *Terminal* and sub-menu *New LAT*. Set up consoles on the machines listed above. As LAT servers announce their presence to the world every 3 minutes or so, it may take some minutes before all LAT ports are recognised. This can be monitored by updating the *Services* Box.

If for some reason you have closed a LAT port but are refused a reconnection, even though the service is recognised, you will need to disconnect the port:

The LAT ports for all services except V5XTAL can be controlled from the left falco terminal in the electronics barrack. The LAT port for V5XTAL is accessed from any falco terminal in the computer hut (except the VAX console in the back room, which should not be used!).

1. Go to an appropriate terminal. Disconnect from the current session by hitting the *break* key. Enter the following commands:
   - set priv
   - password: ___________
   - sho ports all
   - logout port nn, where nn is the port number of the relevant service
   - set nopriv
2. the LAT port should now be available

19 VAX DMA Program

The VAX *dma* program still runs on the VAX from a LAT terminal window located on the NCD Console. To start it up:

1. Log onto user MODEL (Password: ____________)
2. Make sure *dma* is not already running: (sho system)
3. *dma*

The DMA is steered from the *evb* (under instruction from the Mac) to send data to *Disk* or *Null* device. This latter selection is the default. It is the normal way to ensure that data is routed to the VAX monitor processes but not saved locally to some mass storage device (tape or disk) on that machine.

Note that the amount of Disk space available on the VAX is limited. Data is stored, when requested, in the file sys$user:[MODEL]test.dat. From there it can be accessed (for example by the light pulser analysis software) for quick analysis. This file may become quite large if data other than that from a short LP run is written, so care should be taken not to select *Disk* output indiscriminately.

The fraction of events which are sampled and sent to the VAX may be selected with the help of the *Modulo* entry of the Tape Menu at the start of run. Modulo N means that 1 in N events are sent to the VAX. Note that at high data rates, the use of modulo 1 slows the whole DAQ significantly, as the VAX cannot keep up with the data flow due to the limited bandwidth of its Q-Bus Bandwidth a lack of processor power. Modulo 10 is usually an appropriate value, except in pedestal and LP runs, in which cases appropriate values of 1 and 3, respectively, are automatically set.

20 Normal Startup and Stop Sequences

The *V5XTAL Status Monitor* sequentially displays some information as to the DAQ startup and stop sequences. Assuming all components are being read out, here are sketches of what is actually happening:

What does a normal startup look like?

- The run start is flagged by the Macintosh. The user has set up the trigger and tape control appropriately and clicked the final *ok*.
- *evb* sets up its local buffer addresses.

- Hardware Trigger setup occurs. *evb* console lists hardware accessed.
- *evb* checks for presence of LEVB processes (*jet* and *trigger*). If the LEVB is not yet ready to receive a command (eg: pedestal calculation in progress) the *evb* waits indefinitely, until the calculation is complete or a run-stop request is flagged by the Macintosh.
- LEVB cpus are sent the initialisation command. This forces them to self-configure (this can be hardware trigger-specific) and initialise their buffer addresses. A reply to the *init* command with error flag set, will force *evb* to exit immediately.
- *evb* requests run start from the LEVBs. JDC LEVBS check that the pedestals are not stale. If they are, the pedestals are recalculated.
- *evb* requests *CEBRA Startup*. This entails the synchronization of *cebra_master* and *master*.
- If this is successful, a tape is mounted (as requested). The top half of the VME *tape_status* display updates and indicates the active tape in highlight.
- *cebra_master* sends start of run header. VME tape status gets start of run and updates run number and run start time.
- Data acquisition proceeds.

What does a normal stop sequence look like?

- End of tape, end of time-limit, or maximum run-size is reached. Macintosh is informed that it must issue an end of run request (operator responds *ok*).
- *evb* and *cebra_master* collect outstanding buffers.
- LEVBS receive *stop* command. *evb* awaits confirmation.
- JDC pedestal pre-calculation commences.
- *master* is sent end of run header. Stop run time is updated by the *tape_status* process. Run trailers are written.
- Tape is dismounted and unloaded if need be.
- *evb* searches for *cebra_master*.
- *evb* is ready when the message "*cebra_master* found" is displayed on the v5xtal status display.

21 Bicycle Offline Sampling

CBDEC3 is now available at the experiment. In addition to its functions of event-display and histogram display for the bicycle offline, it provides a large scratch disk for the direct dumping of run-data which can take place in parallel to the taping.

Run-dumping is accomplished, via Macintosh control, by a daemon which awaits tcpip connect-requests from v5xtal over a private FDDI network link. The daemon can be started from the bicycle control window (see the bicycle offline documentation) and needs to be running before any data transfer is attempted.

The high bandwidth of the FDDI fiber connection (50 mbit/sec) allows transfer of the data to disk (/os9_scratch/data/Runxxxxx.rdt) with a minimum reduction in the data rate to tape.

For more details on the operation of the bicycle offline, see the manual by Mark Lakata on the subject.

22 OS9 Commands

OS9 is a real-time operating system, similar in syntax to unix but adapted to real time applications through its pre-emptability and well defined real-time response.

Following are some useful commands. Note that pathnames are case sensitive, at least in the case of NFS disks.

procs -e	shows *all* process name table entries
kill NN	kill process number NN
dir	display directory entries
sysmon	shows cpu activity
chd	change working (data) directory
chx	change working (executable) directory
logout	quit interactive session
load -d NAME	load module (program) NAME into memory
tape -r /ctn	rewind cartridge n
tape -o /ctn	unload-offline cartridge n
e filename	edit filename with and edt style editor.
grep item f	find string item in file f
diff f1 f2	list differences in files f1 and f2
type filename	list the file named filename
unlink NAME	decrease the link count of a module by 1

set_title	set up icon and window titles
killproc NAME	kill (first process found) with name NAME
pulldev N1 N2	remove a device N1, driver N2 from the system
nolink NAME	decrease the link count of a module NAME to 0
iniz N1	initialise the OS9 device N1
deiniz N1	de-initialise the OS9 device N1
lpr NAME	print in two-page mode NAME on the VAX printer
reboot	pull the plug on this cpu
notify ic msg	notify the error server: message msg status ic

23 Macintosh Trigger Files

1. On the Macintosh:
   - Double click to select the disk "Crystal Barrel"
   - Then on the icon "CBrun" to select the directory of that name.
   - Then on the icon "Trigger".
   - Select the file of interest, again by a double click.
   - This should start up an editor. (TeachText is ok, if the "application which created the file cannot be found").
2. An example is the following, a zero prong file from the 1996 scan run:

```
! Trigger setup (Aug, 1996) bmb : Zeroprong setting, 1050 MeV
!==============================================================
!
! 0) Hardware trigger:
bsvp1e1_i.sts
```

This line specifies the name of the hardware status file. It documents what is the beam definition:

- **bsv** *indicates beam-silicon-veto,*
- **bsv_p** *indicates beam-silicon-veto with active pileup rejection.*

as well as which hardware trigger decisions are active:

- **p1** *a decision is made on the pwc (read vertex) multiplicity*
- **e1** *a hardware energy sum decision is made*
- **j2** *a jdc multiplicity decision is made*
- **f1** *a face multiplicity decision is made*
- **s1** *a software multiplicity decision is made.*

If a file of the desired form is not available on (/cb1/pro/TRIGGER), then an expert *(H. Kalinowsky, U. Wiedner, or P. Kammel) will need to create one.*

! 1) PWC multiplicity (min...max):
0 0 0 15

Formerly the pwc-multiplicity, this now refers to the vtx multiplicity. The special value 255 255 255 255 forces downloading of the predefined multiplicity file into the malus (multiplicity logic units). Normally, the first two entries indicate the (inclusive) range (min $\rightarrow$max) of the vtx backplane multiplicity The last two entries indicate the (inclusive) range (min rla max) of the vtx stripside multiplicity.

! 2) JDC multiplicity:
j2m00xxxx.par

This is the name of a file which defines the jdc multiplicity malu setup parameter file. It is normally of the form j2maabbcc.par, where aa indicates the range of accepted multiplicities in the first two trigger layers, bb in the second two trigger layers, and cc in the last two trigger layers. The value "xx" means "don't-care".

! 3) FACE multiplicity (min...max):
0 15

Programmed range of accepted face multiplicity (inclusive).

! 4) Options (hex number):
1100 pedrun=0 generate=0 bypass=0 histo=1 notrig=0 nopedsub=0 low=0 hi=1 00000000

> *These options are not currently used.*

! 5) Total Energy Box Control (min, max, offset, shift_min, shift_max)
1800 2800 0 0 0

> *Minimum and maximum range of hardware total energy to be accepted by Tony's box: the TET or total-energy-trigger box. Be aware that these values are programmed. The sum includes offsets corresponding to the sum of the differences between pedestal and threshold, plus the 4 channels which by design (in order to force readback from each fera-crate) are set up with low thresholds. The first two values are minimum and maximum. The other values should be set to zero and can be used to offset and multiply (shift) the programmed minimum and maximum values.*

! 6) Software Trigger Name (as in title)
default.trg

> *Name of the software trigger in use. This should normally be reflected in the title of the file, but that is no longer mandatory. "default.trg" should be entered if no software trigger decision is required.*

! 7) con_trg:
8001

> *In early versions this indicated the number of consecutive software trigger decisions to be made. It is now used as flag (hex number, should be ¿0x8000) which indicates the damod version number, currently 0x8001*

3. From the "File" menu, save the file and quit the editor.

24 DAQ Shift Duty

- Please familiarise yourself with the information contained herein. If you are unsure where some crates or modules are, have a look in the hut or ask someone. Do this before your shift so if something is unclear you can contact someone who knows. (Figure 5 shows the location of the key components as they are located in the electronics hut.)

- Report details of DAQ failures in the DAQ Error Log. This will help those responsible identify errors and correct them.

25 Who to Call

The following is a list of who is the *best* expert (or at least responsible) for each DAQ software component. Note that as of 18 October, 1996 at 23:00 hours, French phone numbers will change. The prefix 10, previously used to reach France, will be replaced by 1004.

Which Expert?

What	*Who*	*CERN*	*Home*
DAQ Software Overall Event Builder Tape Control JDC LEVB	Bruce Barnett	78566	10-5040-4780
JDC Microsys/FADC Slow Control JDC High Voltage	Karsten Wittmack	75913	10-5028-3758
Crystal Readout	Chris Pinder		
Software Trigger	Nigel Hessey	72309	10-5041-8759
VAX General	Kersten Braune	73240	10-5020-8045
COLA SLOW Monitor Presenter	Chris Pinder	72309	
JDC Monitor Scaler Monitor	Klaus Peters		
PWC Monitor VTX Monitor	Stephan Spanier		10-5028-2544
Event Display Bicycle Offline	Mark Lakata		

Table 3. Software Responsibles for Crystal Barrel.

If a piece of software is failing but there is no urgency, send an e-mail describing the symptoms (with a copy to me, Bruce.Barnett@cern.ch) to the relevant person.

26 Known Bugs

- *evb* will not find slow-control module dynamically. slow control must be running first for *evb* to recognise the slow-control data area as existing.
- *vtx* sometimes does not respond correctly to control-e command.
- *vtx* fails to exit from pedestal determination loop on control-c
- *vtx* pedestal determination will not complete, if backplane QDCs are out of nominal pedestal range (eg: are not working). This will hang the DAQ.
- *evb* can hang on startup if a face-latch module vme interface fails to respond.
- specification of nonexistent status or multiplicity files to *evb* results in mimimum-bias operation, with complaints, but run-termination is not forced.
- *jet* pedestal measurment fails on many processors for 1 or 2 tries immediately after barrel-pedestal or light-pulser runs.
- v5xtal bus arbitration may hang (CPU crashes on boot, with a solid red-status lamp), when v5xtal crate is turned off and then on some seconds later. Solution is to turn both e1xtal crate and v5xtal crate for some minutes, then back on. This clears the vic links between them.