

Controls Rookie Book

Introduction	3
Controls Intro.....	3
Console Applications.....	7
Introduction	7
Conception.....	7
Program-To-Page.....	7
Proofing	9
Upgrading	9
New User Into.....	10
Links.....	11
Introduction	11
CAMAC	11
CAMAC (Booster).....	13
CAMAC (Main Injector).....	14
CAMAC (Tevatron).....	16
CAMAC (Switchyard)	17
CAMAC (Pbar).....	18
CAMAC Summary including PIOX, PIOR and BTR	19
Non-CAMAC Links.....	22
MDAT Link.....	23
Abort Links.....	27
QXR Link	30
JAVA.....	32
Specific Java uses	33
Networking.....	36
Introduction	36
What's a Network?.....	36
OSI Model	37
Network Topologies.....	38
Making Connections	40
Access Methods (Intro)	42
Access Methods (ethernet)	42
Access Methods (ARCnet)	43
Access Methods (FDDI).....	44
Internetworking (Intro).....	44
Internetworking (Repeaters).....	44
Internetworking (Bridges)	45
Internetworking (Routers)	46
Internetworking (Switches)	46
Internetworking (Gateways)	47
Internetworking (Firewalls).....	47

Controls Rookie Book

Internetworking (OFS)	47
Internetwork Protocols	48
Conclusion	51
Clocks and Timing	53
Introduction to Accelerator Timing	53
TCLK Generation and Distribution	55
The Timeline Generator and Basic Machine Resets	57
Timing Delay Hardware	59
Software Generated TCLK Events	62
TCLK Hardware Sources	63
Introduction to Beam Sync Clocks	64
Beam Sync Applications	66
Controlsopedia	69
The Introduction	69
The Controlsopedia - Modules 0–9 -	71
The Controlsopedia - 0-9 -	75
The Controlsopedia - A -	76
The Controlsopedia - B -	81
The Controlsopedia - C -	84
The Controlsopedia - D -	93
The Controlsopedia - E -	97
The Controlsopedia - F -	99
The Controlsopedia - G -	104
The Controlsopedia - H -	104
The Controlsopedia - I -	106
The Controlsopedia - J -	108
The Controlsopedia - K -	108
The Controlsopedia - L -	109
The Controlsopedia - M -	110
The Controlsopedia - N -	117
The Controlsopedia - O -	118
The Controlsopedia - P -	120
The Controlsopedia - Q -	124
The Controlsopedia - R -	125
The Controlsopedia - S -	126
The Controlsopedia - T -	131
The Controlsopedia - U -	134
The Controlsopedia - V -	134
The Controlsopedia - W -	135
The Controlsopedia - X -	136

Controls Rookie Book

Introduction

I think it'd be fairly safe to say that without controls, we'd be hard pressed to accomplish anything substantial here at Fermilab. I'll refrain from any abstract correlation's between love and controls, but it's a fact that around here, "controls makes the beam go 'round." Controls is not only a diverse collection of hardware and software building blocks, controls is the glue that binds the building blocks together.

CONTROLS INTRO

We're not going to begin by dragging you through accelerator theory one more time, but would suggest a good first read would be the ever-popular *Concepts* book for a concise and comprehensive over-view of life as we know it here at Fermilab. It's not a requirement, however, so if you're already to the point of being unable to put this down, by all means continue on — it only gets better. We're also going to try very hard not to wander off into the accelerator-system-specific controls — in other words, all the neat controls stuff that has already been covered in your other rookie books.

In the following chapters you're going to read about those major building blocks I spoke of, as well as the glue (there are different types you know). We'll begin where you live — Console applications. Where do these fine Operator tools come from, how are they developed and what, if any, influential role do *you* play in how they work? In the process, we'll give you a non-programmer's-eye view of the programmer's world, and we'll sidetrack ourselves into a few other areas that warrant space and comment.

The building blocks, of course, are pretty useless if they can't communicate with each other, so we'll devote an entire chapter to the sealing wax and binder twine, more formally known as "links". There, you can sit

Controls Rookie Book

back and let yourself become absorbed into the world of CAMAC and other fun places.

I'm sure that for little other reason than to confuse the whole "link" issue, there's another binding medium within the controls system, known as the "network." An entire section within the Controls Department is dedicated to the task of keeping computer systems communicating with one another. You know, things like allowing you access to the electronic logbook, email, weather information, and more than one or two other things. These Controls types talk in a lingo all their own, but after reading our chapter on networking, you'll be able to party with the best of them.

Someone once said that timing was everything. They obviously were daydreaming about accelerator controls and how it could be used to synchronize all the various parts of the Fermi electro-mechanical world. It only seems proper, then, that we devote an entire chapter to this whole "timing" thing.

When laying out this book, we found that the world of controls is made up of so many things that one can describe in a simple paragraph or two. Not believing in the philosophy that one should write a chapter where a paragraph will do, we incorporated all these pearls in our "Controlsopedia" chapter. More than just a glossary, the Controlsopedia will give you a wealth of information on more things than you ever wanted to know.



☛ If, perchance, in your reading, you should happen across a paragraph denoted like this one, it's an indication that we're about to

Controls Rookie Book

wander into deeper waters than the neophyte swimmer may wish to paddle. It's fair game to skip these if you feel like you're already in over your head.

Probably more so than other rookie book subjects, the world of controls is always changing. Writing this book has been a lot like painting the Golden Gate Bridge — by the time one completes the final layer, it's time to start over (especially if you're writing it while major accelerators are coming on-line... or going off-line). One simply has to realize, however, that controls is an ever-changing world and go on from there. Certainly within Controls, there's ever a movement underfoot to look to the next level and where we want to be when your kids are running this place. Although Fermilab's breakthroughs and accomplishments in the physics world are very high profile, these milestones couldn't happen without the lab's support departments continuing to challenge the edge of technology. Yes, believe it or not, boys and girls, the future is us — scary stuff, huh?

Controls Rookie Book

Notes:

Controls Rookie Book

Console Applications

This chapter will help you take a look at the tool which Operators use most often... those often taken-for-granted console pages. Before you begin to cringe, you'll be happy to know that the intent here is NOT to spoon-feed you with programming techniques, but rather to give you a simple overview of how a typical console page comes to be.

Who comes up with this stuff? More often than not it's the associated equipment that dictates the need for control. *Simple* equipment, which requires basic control and monitoring, can be dealt with from generic pages like the *Parameter Page* or the *Digital Status Page* (S53), but more complex equipment or the need for detailed control may dictate that a dedicated page is required. So, in the grand scheme of things, physicists/engineers discover a need to control some aspect of the accelerator, such as "flying wires" or "synch light" for instance. They play and fiddle with their hardware behind the scenes until they come up with a gizmo that does basically what they hope for. They define the gizmo's inputs and outputs, and suddenly discover they're now up against the obstacle they've been trying to ignore... the daunting task of making a human interface to their hardware. Often to their chagrin, it's not the interface of their choosing but rather the interface we recognize in the MCR... the console page... the ACNET console PA or *Primary Application*.

Without getting into the nitty-gritty, suffice it to say that programmers write their program in the C language or Java (many old FORTRAN programs still exists however). The program then gets converted to machine-executable code and presented to the user on a chosen console index and page. For those actually interested in *creating* console applications, there are on-line tutorials and guides on the World Wide Web beginning at:

INTRODUCTION

CONCEPTION

PROGRAM-TO-PAGE

Controls Rookie Book

www-bd.fnal.gov/controls/consolesii.html

All of the software tools are defined on this and its related Web pages, making it an indispensable reference for programmers.

☛ Since approximately 1985 all program code has been “captured”. What that means is that there’s a sort of librarian which takes all of the programmer’s files related to a given PA, checks them for errors, compiles them into machine-executable code, makes sure it conforms to certain specifications and, if there were no problems, the librarian stows away a copy and makes the resulting program available to the users. In our environment, that librarian is called MECCA[†] — a program. Use of this technique insures that one can always find the most current *working* version (and a few of its predecessors) in one central location. A similar program, called FileMaster, keeps track of any *files* that may be associated with a given program.

Using the available software tools, the programmer generally begins by creating a test (beta) version. Its user interface generally falls significantly short of robust, but allows the gizmo’s designer to try out the remote readbacks and control to see if the hardware and the software algorithms work as planned. Early (and subsequent test) versions of the program will be placed on the Z-index page. This particular page provides the programmer debugging tools and allows a legal “back door” into the system, at an expense of course; shortly past midnight, all of the previous day’s programs are automatically removed from the Z page. This technique allows programmers to make quick modifications and tests to an unproven program while at the

[†] Management Environment for Controls Console Applications.

Controls Rookie Book

same time not allowing it to take root long enough for it to become a forgotten land mine. Too, just the fact that the program is *on* the Z-index is a flag to users that its unsolicited use is ill advised.

Once the major software and hardware kinks are worked out, the programmer will generally make the program user-friendly (at least to *some* degree); insuring that it conforms to recognized look-and-feel standards. That done, the program will likely be assigned to the W-index page (the “Working” page). This page is reserved for programs that are considered out of the debugging stage and ready for the real world. Through necessity, word-of-mouth, lunchtime chatter or sundry other obscure methods, Operators will catch wind of the new program and then the *real* testing begins.

In an ideal world, this point is where there would be a standardized review and trial period and, in a coercive kind of way there is. Although programmers are not *forced* to make their program Operator appealing, it’s generally in their best interest to do so. Simply put, if the Operator’s don’t like it, they won’t use it. Ultimately the Operations Department Head (usually through Operator feedback) reserves the right to reject programs that don’t meet the right standards.

PROOFING

Although Operators generally don’t have enough backup time to create new programs from scratch, they’re often able to make quality changes to existing applications. Observing a program’s interface from the driver’s seat over and over again, is an ideal way of identifying assets and shortfalls. It shouldn’t be too surprising that Operators are masters at this and that often, without a major time investment, an Operator can turn an

UPGRADING

Controls Rookie Book

application page into one which is significantly more user-meaningful and user-efficient.

Although anyone has the capability to jump in and make changes to any program, normally only “orphaned” programs are considered fair game — those programs which have been abandoned because their programmers have found gainful employment elsewhere. Modifying the code belonging to an *existing* (some would say “ungainfully employed”) programmer without an “OK” is never a good idea, besides which, MECCA being the clever software librarian that it is, has the capability to notify a program’s author of what’s going on. Too, it’s worth pointing out that if the code malfunctions, all fingers point to the person who last modified it, no matter how innocuous the change(s) may have been.

NEW USER INTO

Those interested in more details concerning programming, can find some very useful information on the following URL:

adcon.fnal.gov/controls/clib/applications_style_guide.html

There, one can find such things as programming style guidelines which documents some of the conventions that Operators, over the years, have come to know and love. For example:

- what colors are and are *not* used and where?
- when and when not to assert a “caution”.
- rules about using other screens.
- acknowledging/processing user interrupts.

Other information at this URL will describe dealing with errors, program debugging, file IO, graphics, etc. Enjoy.

Controls Rookie Book

Links

All of the consoles, front ends, and applications software are completely useless unless there is a way to talk to the hardware out in the field — the hardware that makes the accelerators work. "Links" are what provide that communication. There are several different kinds in use, each with a different protocol, hardware setup, or transmission medium.

The most pervasive controls hardware found in the service buildings is of the CAMAC type. Booster, Main Injector, Tevatron, Pbar, and Switchyard all depend heavily (but not exclusively) on CAMAC, so it makes sense to describe it first

CAMAC is an acronym for "Computer Automated Monitor and Control". It conforms to an international protocol for standardizing cards and related hardware. The ubiquitous black CAMAC crates can be found in every service building, where they are constantly gathering information from nearby equipment or disseminating orders that may originate miles away. Specially designed CAMAC cards can converse with local microprocessors (such as BPMs) set the flattop value of a dipole, or ensure that an abort kicker fires at precisely the right time.

Data processing within the CAMAC crate is carried out using an eight-bit parallel format. The crate has 25 slots to hold the various cards, which can be either one or two slots wide.

Backing most CAMAC links is a VME front end. At least some of the VME front ends are located along the north end of the west wall of the computer room. Each of these front ends controls its own empire of CAMAC crates. Each CAMAC link controlled by a front end is separate and distinct from the others.

INTRODUCTION

CAMAC

Controls Rookie Book

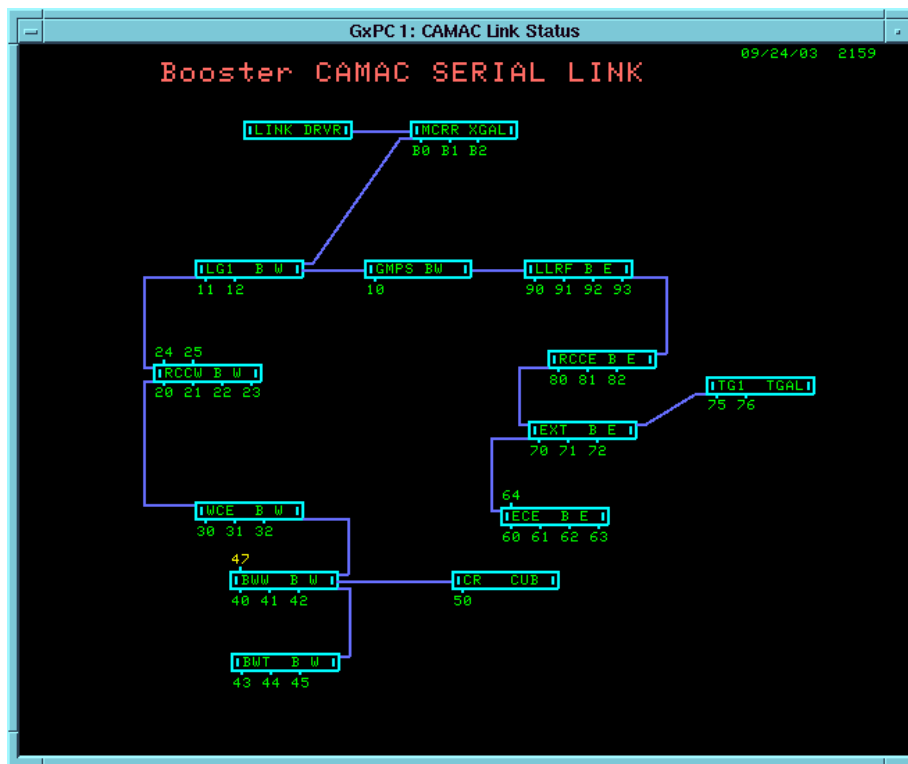
There are actually three data links associated with each front end. The first is known as PIOX, or the transmission link. Requests originate at the front end and are sent via repeaters throughout the link, where the appropriate CAMAC crate decodes them. The second is PIOR, or the receiving link; here replies to requests are shipped back to the front end. The third link is BTR, or Block Transfer. The BTR link allows humongous gobs of data to be sent back to the front end uninterrupted.

Each of the CAMAC links works on the same principles, using PIOX, PIOR, and BTR, but each has a unique configuration and some special features worth knowing. All are launched from their respective VME crate via the SLD (Serial Link Driver), which converts the parallel data inside the front end into a serial stream to be put onto the link.

The first stop is usually Rack #30 in the back racks of the MCR, informally known as "Repeater Central". The repeaters in Rack #30 then send the data on to each of the links.

An overview of the general layout of each of the individual links is given below, followed by a more detailed description of the CAMAC hardware common to all of them.

Controls Rookie Book



After feeding three Booster crates in the MCR, the link goes to a location in the 400 MeV area of Booster, where it divides into east and west branches. (Unfortunately, a power outage in the Booster West Gallery will take down both branches of the link.) One of the terminal crates of the west branch is at CUB, which means that LCW readbacks depend on the integrity of this branch. The other terminal crates of the west branch service the upstream portion of the MI-8 line. The east branch takes care of the GMPS area before passing through the Booster LLRF Room and continuing on to the south end of the Booster East gallery.

Controls Rookie Book

crates are part of the Main Injector link because they contain the cards that control the P2 and P3 extraction lines from the Main Injector; the extraction lines are predominantly made up of old Main Ring components. The F Sector branch of the link terminates in the F0 Service Building, where the cards interface to devices in the P1 line.

The remaining two front ends feed the MI ring itself. MI2 front end connect to the MIS (S for South) repeater. This repeater handles communications from MI60S including the MI-60 control room and continues clockwise until MI-40.

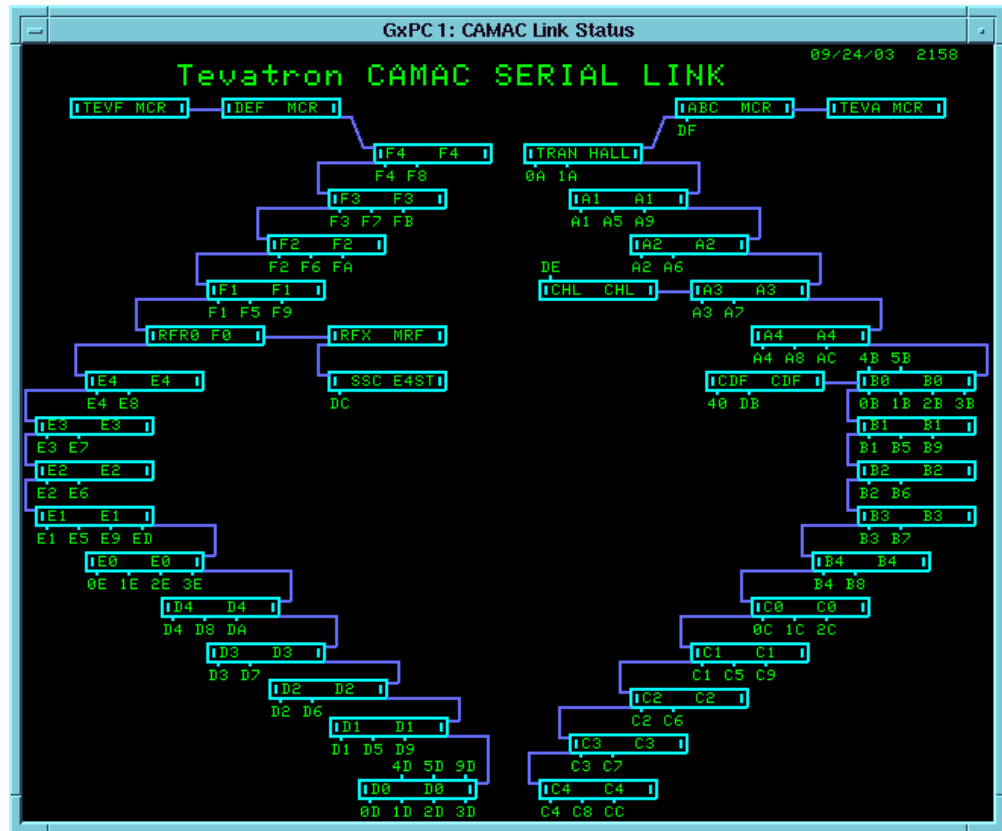
MI3 front end connects to the MIN (North) repeater. This repeater controls communications from the northern half of MI-60 counter-clockwise around the ring until MI-30. There is also a branch extending from MI-10 to MI-12 for MiniBooNE.

The old Main Ring link, of course, went around what is now the Tevatron ring. It was also a fiber-optic system, and many of the repeaters were left in place so that they could be used by other links. These links will be described later.

It is important to recognize that in the Main Injector ring, many subsystems do not make use of CAMAC at all. Among these are the LCW systems, digital control of pulsed power supplies, and vacuum.

Controls Rookie Book

CAMAC (TEVATRON)



The Tevatron link begins with two front ends located in the back racks of the MCR, near Repeater Central. TEVA connects Transfer Hall through C4 with short branches to CHL and CDF, while TEVF connects F4 through D0. Each FE has its own repeater group, TEV ABC for TEVA, and TEV DEF for TEVF.

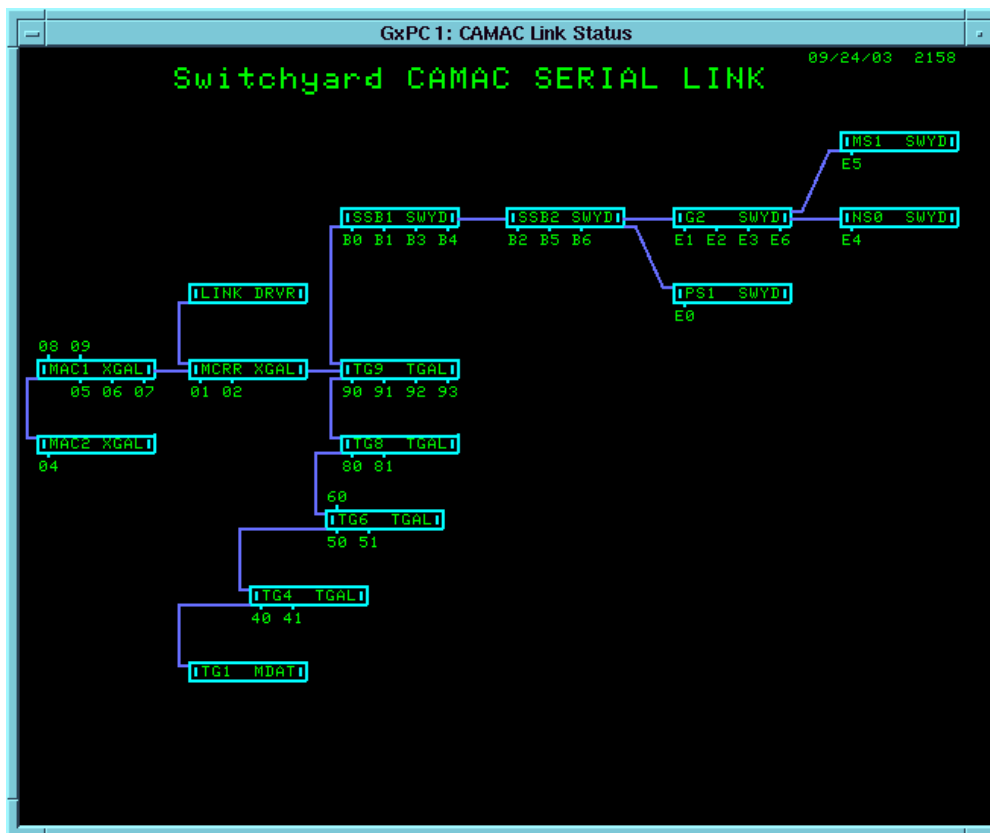
The Tevatron link uses a 19-conductor Heliax cable to propagate the signals. Why a 19-conductor cable? It was once used to return data — quickly — from underground nuclear test explosions in Nevada. We got the part that wasn't vaporized. Cheap. The cable can be seen fanning out from a large junction box attached to the ceiling at each service building. In

Controls Rookie Book

between buildings it is buried below ground, where it is a favorite target of back-hoe operators.

The “19-conductor cable” also carries several other links, to be discussed later.

CAMAC (SWITCHYARD)

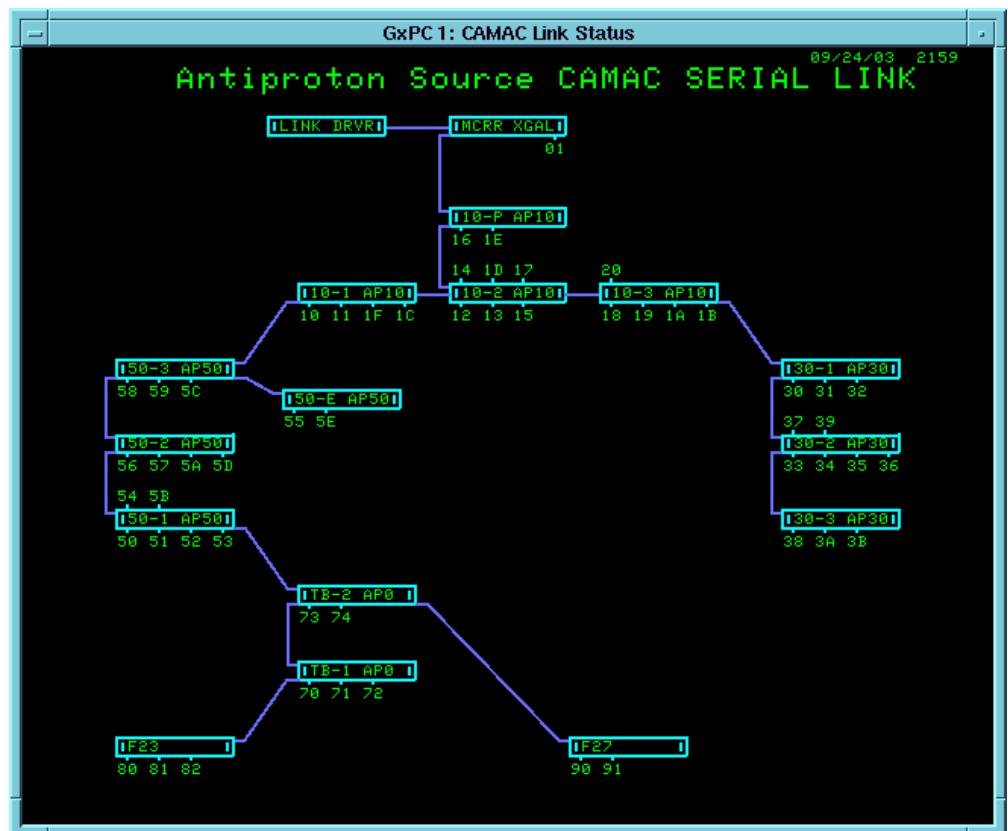


From "Repeater Central," the link is initially split into two branches. In the MAC Room, it supports four CAMAC crates whose primary purpose is to generate TCLK, the accelerators' main timing system.

Controls Rookie Book

The other branch goes to a set of repeaters at TG-9, where it again splits into two branches. The first works its way back to TG-1, where historically it provided control of most of the devices in the old 8-GeV line. The second goes to the SSB, the G2 Service Building, and to remote outposts in the former Demilitarized Zone between the Accelerator and Research Divisions. A final link repeater does not service any CAMAC crates, but provides clock events to the former Research Division controls system.

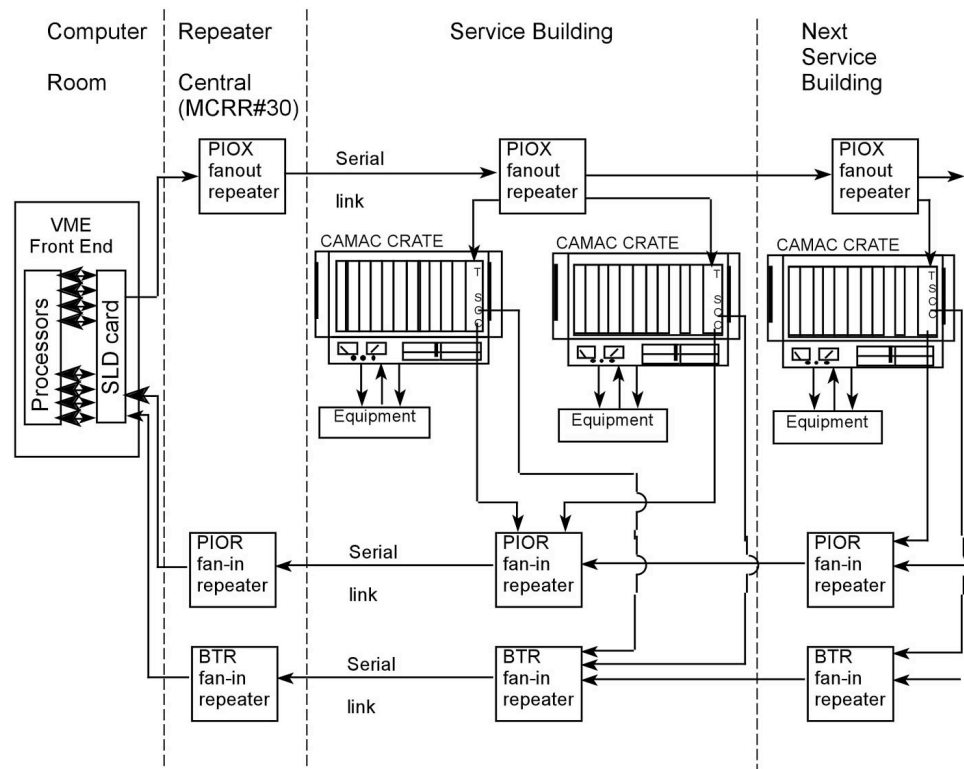
CAMAC (PBAR)



This one is fairly straightforward. From Repeater Central, it goes to AP-10. One branch from AP-10 goes to AP-30, with another branch going to

Controls Rookie Book

AP-50 and from there to AP-0. Branches from AP-0 connect to the F23 and F27 Service Buildings.



CAMAC SUMMARY
INCLUDING **PIOX, PIOR**
AND **BTR**

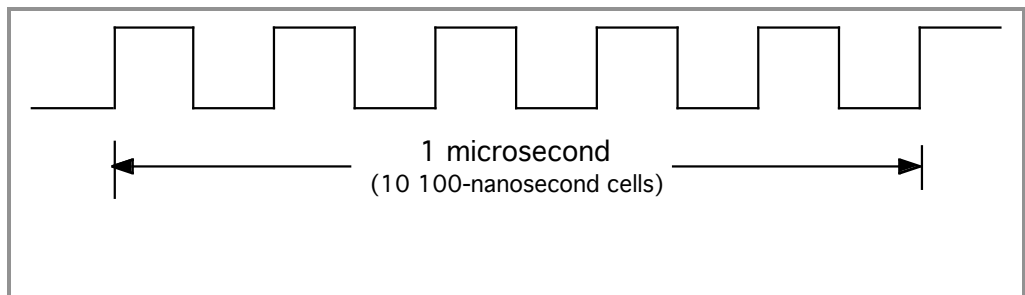
Fig 6 Generalized CAMAC Links

So much for geography. Now for dissection of a CAMAC link, bit by bit (well, almost). Remember that the PIOX link originates at the VME front end (Fig. 6). The INTEL 386 processor on board the front end, like most processors, uses a parallel bus. To avoid the nightmare of having to run miles of ribbon cable, the data to be sent to the CAMAC crates are converted to a serial format by the integrated Serial Link Driver. The PIOX transmission is organized into a handful of frames that are either 16 or 24 bits

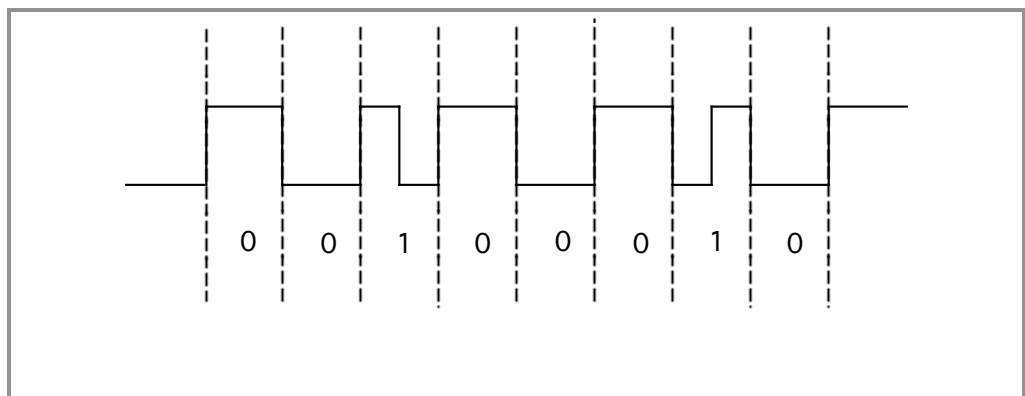
Controls Rookie Book

long. Many of the bits are dedicated to communications protocol, such as for instructing the crate how to organize the data it receives from the link.

The transmission is clocked out at a 10 MHz rate. It is coded as a square wave alternating between 0 and 2 volts (or so). The square wave is interpreted by the hardware as consisting of "cells" 100 nanoseconds long:



If the voltage remains constant for the full lifetime of the cell, the bit is interpreted as a zero. It does not matter if the voltage is high or low. If the voltage makes a transition from one state to another in the middle of the cell, the bit is interpreted as a "one."



Here, the vertical dotted lines delineate the cell boundaries. This particular string of bits could be interpreted as "00100010" (that's 22 in

Controls Rookie Book

hexadecimal, for the digitally impaired). The encryption scheme described is known as "modified Manchester" coding.

This electrical signal will have to face large distances — in the case of the Tevatron, about 260 meters between buildings — and will attenuate so as to be unreadable if not boosted by repeaters at regular intervals. The first repeaters seen by most of the PIOX links are the ones at "Repeater Central" (MCR back Rack #30) mentioned earlier. The repeater cards are housed in a half-high (5.25") NIM crate. NIM (for Nuclear Instrument Module) is another international hardware standard, not quite as common as CAMAC, used at Fermilab. Repeaters for other major links — for example, MDAT and TCLK — are also found in these crates. These links will be described later.

Whether electrical or optical, the signals on the link will be sent throughout the regions where they are needed. In the Tevatron, Pbar, and Switchyard there are repeaters at every service building. In Booster, they are distributed throughout the East and West Galleries. It is important to know that if the NIM power supplies (found on the right-hand side of the NIM crate) fail, not only will the CAMAC crates it serves disappear, but everything downstream as well.

If the front end did its job right, sooner or later the transmission will be decoded at a specific CAMAC address, say, at Tevatron Crate \$D3. (The \$ sign is a reminder that the number is hexadecimal). The CAMAC card that actually recognizes the address is the TSCC (Tevatron Serial Crate Controller) card, which inhabits the rightmost 2 slots of the crate. ("Tevatron" is part of the name regardless of the machine, left over because that was the machine where the CAMAC links were first implemented.) Since CAMAC

Controls Rookie Book

uses an 8-bit parallel bus, another task of the TSCC is to convert the serial stream of data back into a parallel format. Finally, the TSCC routes the data to the card for which it was intended.

Frequently, a request for local data is a part of the PIOX transmission. Requests are constantly being generated by applications programs running in the MCR, fast time plots, alarms screens, etc. Microprocessors, MADCs and the like which are overseeing particular subsystems have cards representing them in the CAMAC crate. These cards are alerted and gather the requested information, which is shipped back to the TSCC. The TSCC serializes the transmission and sends it out along the PIOR, or return link. (To understand how the PIOR link works, just read the last few paragraphs backwards.) Eventually it makes its way back to the front end, which in conjunction with the database and the applications program organizes the information for display on an MCR console.

The Block Transfer link (BTR) works much like PIOR, except that longer strings of data are sent back in a single transmission.

NON-CAMAC LINKS

PIOX, PIOR, and BTR are not the only links; there are several others. Usually these others originate somewhere in a CAMAC card, but they are not directly dependant on the VME front ends. Except for the front ends, they follow paths similar to PIOX, PIOR, and BTR, since they often use the same half-high NIM repeater crates, or fiber-optic repeaters, and usually provide services to the same CAMAC crates. However, those same services can be used by non-CAMAC equipment as well. We'll now discuss some of these links.

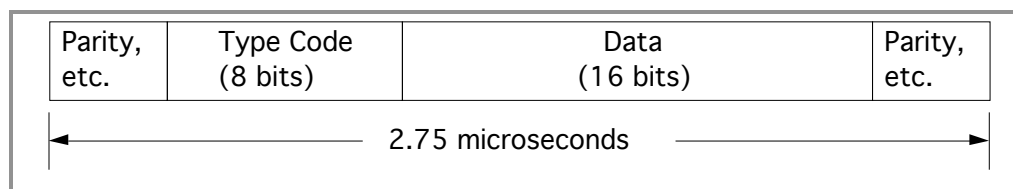
Controls Rookie Book

MDAT (pronounced em-dat) stands for "machine data." More often than not, the data referred to involves bus current in the Main Injector or Tevatron. The bus current values are transmitted in several different forms:

MDAT LINK

- there is the "measured current," which comes from a transductor
- the "programmed current," which is what MECAR or TECAR is requesting from the magnets
- there is a "measured Idot," or measured rate of change of current in the magnets
- and a "programmed Idot" — sometimes values for beam "momentum" are transmitted as well.

These are supplemented by a potpourri of measured values such as beam intensity and "time @150 GeV". Rather than being targeted at a specific crate, these data are made widely available throughout the accelerators because there are a multitude of devices that scale directly to the current in the main buses. The most blatant example is that of the dipoles and other correction elements: the Main Injector and Tevatron each have thousands of magnets which play "energy ramps", that is, currents which are in some way proportional to the current in the main bus.

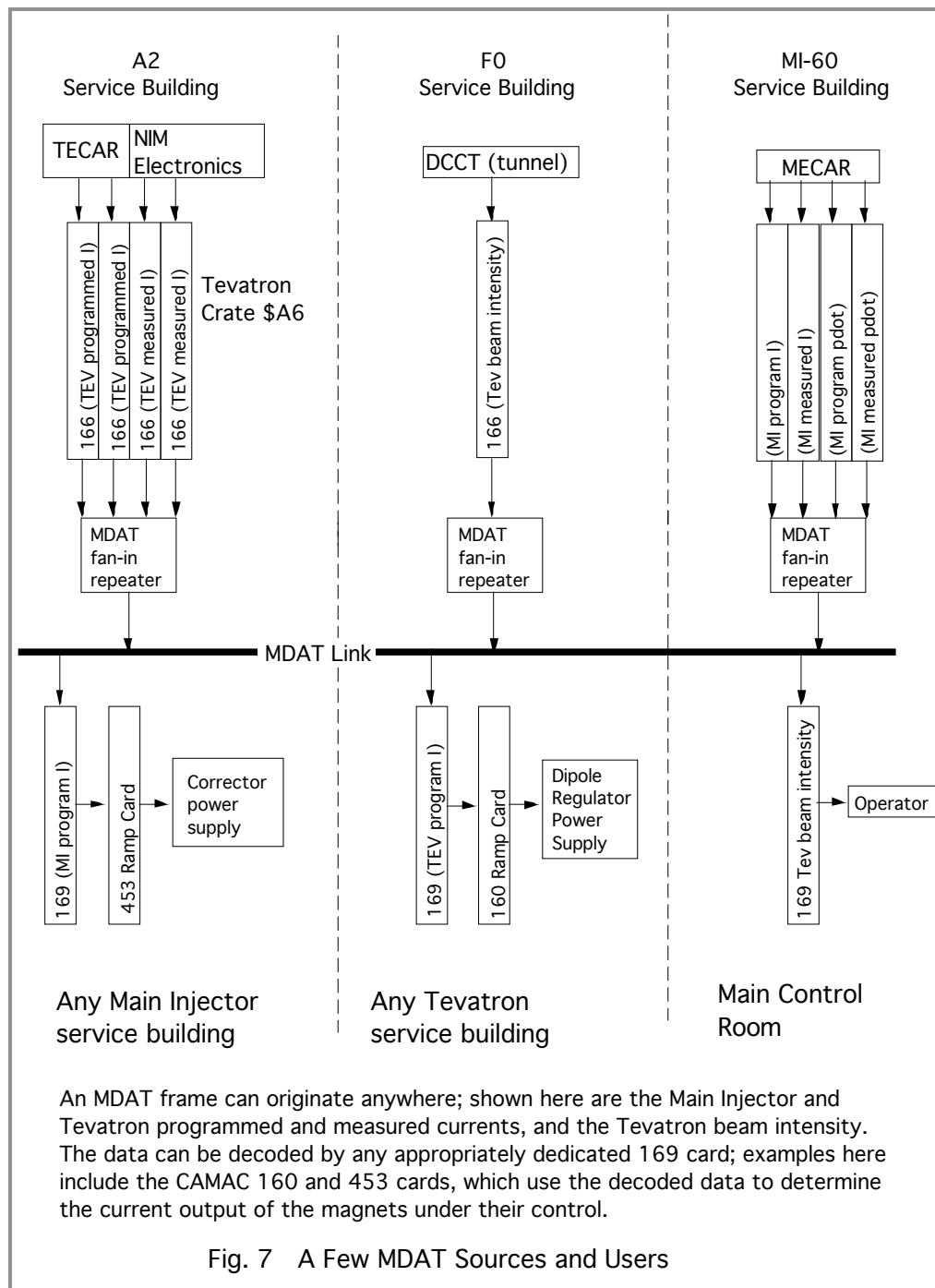


Like PIOX, PIOR, and BTR, MDAT bits are broadcast at a 10 MHz rate. The data portion of the 28-bit MDAT frame is 16 bits long, allowing for considerable resolution. The data are preceded by an 8-bit "type code" which

Controls Rookie Book

identifies the kind of data in the frame — for example, a type code of \$10 indicates that the data in the frame represent T:MDAT10, or Tevatron programmed current. There are a couple of bits on either side for parity and administrative overhead. Altogether, the package weighs 2.75 microseconds.

Controls Rookie Book



MDAT frames are transmitted at a 720 Hz rate, the same rate that the Main Injector and Tevatron power supply links are updated. (Coincidence? I

Controls Rookie Book

think not!) At the time of this writing there are 16 types of frames circulating on the link. They all share the same cable and the same repeaters. At the beginning of each 720 Hz interval they are launched sequentially, one every 10 microseconds or so, in numerical order. In that way the interval between like types will remain constant. There are 1,389 microseconds in every 1/720 of a second, so there is plenty of room for newcomers.

As shown in the accompanying figure, many of the MDAT signals are launched from dedicated 166 cards; the cards are usually located close to the source of the signal. For example, there are four 166 cards at the A2 Service Building, near TECAR, that encode the Tevatron bus current data on to MDAT. The 166 cards are analogous to the SLD of VME fame, in that they convert parallel data into a serial format to place on a link. MECAR, however, is an exception to the rule, since the MDAT data is encoded directly from the hardware on board the VME crate via a VMIC 2510.

Once on the link, MDAT is available to any device that wishes to use it. It is decoded by yet another dedicated CAMAC card, the 169. Each 169 can be configured, using a PAL and dip switches, to decode a given type of MDAT frame. For example, if the Main Ring correctors need to track the Main Ring programmed current (which they do), a 169 module that decodes M:MDAT30 can be placed at each service building. Likewise, a 169 module dedicated to T:MDAT10, the Tevatron programmed current, can perform a similar function for Tevatron correctors.

In the service buildings around the ring, MDAT utilizes one of the 19-conductor cables. Other accelerators that need MDAT have similar copper-based links, except for the Main Injector with its fiber-optic repeaters.

Controls Rookie Book

By now it should be obvious to the partially alert reader that the 169 card will convert serial data on the link to an 8-bit parallel format for local consumption.

It is worth noting that even though CAMAC cards are often used to generate and decode MDAT frames, MDAT is not considered to be a CAMAC link because it uses a different set of link repeaters. CAMAC, by rule, consists only of PIOX, PIOR, and BTR links.

The Main Injector and the Tevatron each have kickers which remove beam from their respective machines. Sometimes beam is aborted at a scheduled time near the end of a machine cycle, but sometimes it is also necessary to remove the beam instantly should certain devices unexpectedly fail. Each machine therefore has an "abort loop" which watches over those devices. If all is well, the abort loop is continuous and beam is permitted. If a failure occurs, the loop is broken and beam is kicked out of the machine. (Linac, Booster, and Pbar each have an abort loop, but as there is no effective way to get beam out of the machine before completing a regular cycle, the loops are more accurately thought of as beam inhibits. An abort in Switchyard is implemented by aborting beam in the Tevatron.) The hardware and applications are similar in all of the machines mentioned.

ABORT LINKS

The Tevatron has an abort dump at A0. During Collider operation, beam is aborted "internally" in Transfer Hall. The option is exercised when antiprotons are present. An internal, removable dump is feasible during collider mode because of the relatively low intensities encountered.

Controls Rookie Book

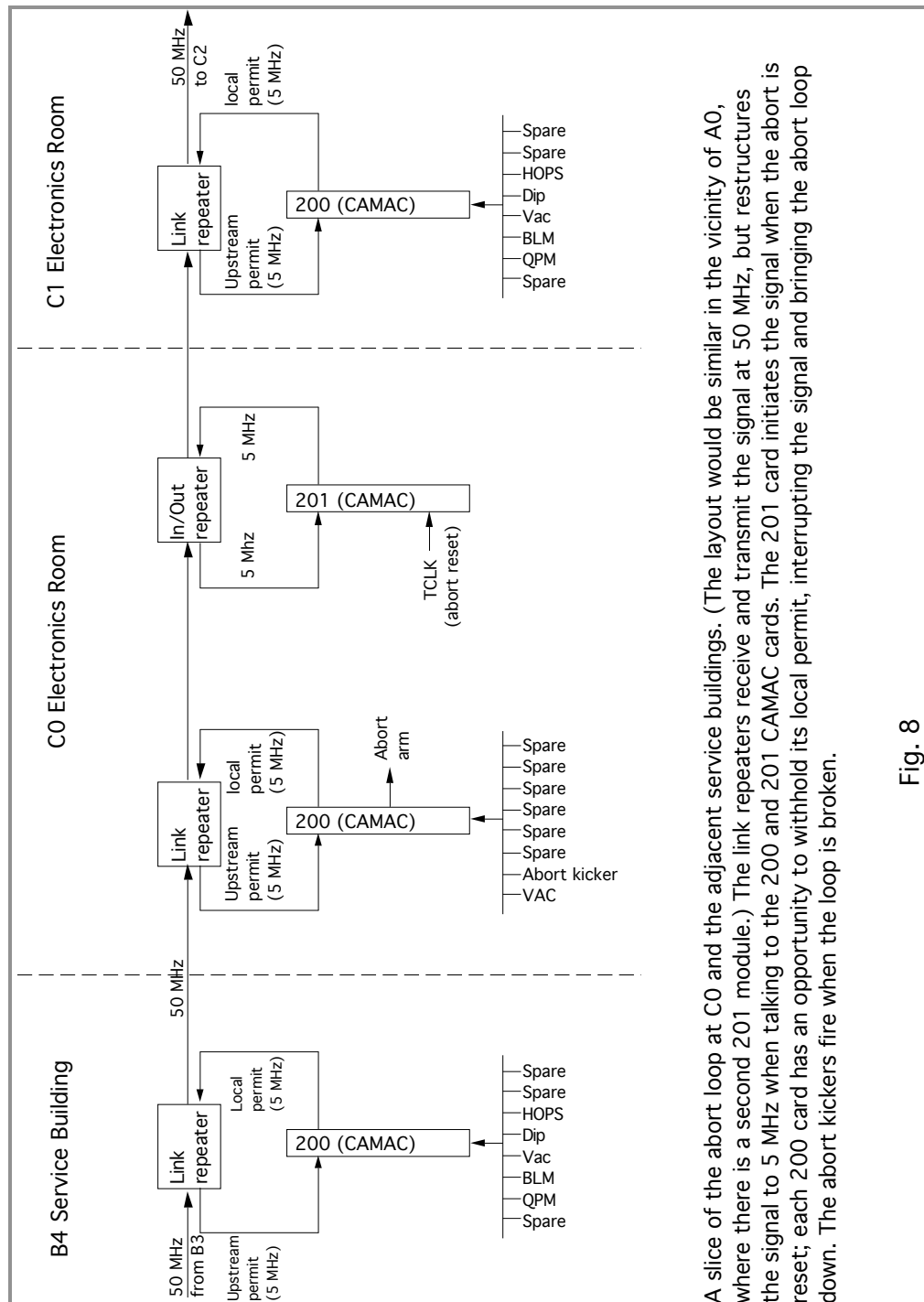


Fig. 8

In the Tevatron, the abort loop utilizes — what else? — one of the 19-conductor cables. The link is simple: a 50 MHz signal must be present on

Controls Rookie Book

the loop. If it isn't, the loop is down. The 50 MHz is generated by a CAMAC 201 module housed in a crate at the A0 Service Building (Fig. 8). The loop circles the ring, and at every service building there is a CAMAC 200 module waiting to bring it down. The 200 module is known as the "Abort Concentrator," it accepts up to 8 inputs (a number which should seem familiar by now — there is one input for each line of the 8-bit CAMAC bus). Behind the CAMAC crate is a patch panel with the abort input cables running into it. Below the line of spigots for the abort inputs is another line of spigots labeled "current sources." The 200 module interprets the presence of current as "good;" in the absence of current the module will interrupt the 50 MHz signal and take down the loop. The inputs can be jumpered using the current sources. Obviously this is not done trivially.

When an abort reset is sent from the "Abort Link Status" page (T67), the CAMAC 201 module is told to initiate the 50 MHz signal. It takes about 34 microseconds for the signal to come full circle if the loop is healthy. If it is not, the 201 card will cease after about 100 microseconds and wait for the next reset to be sent.

Most abort inputs in the Tevatron will immediately remove beam if asserted. Included in the group are most of the magnets: dipoles, correction elements, low beta quads, and the Tevatron ramp itself via the TECAR input at A2. The QPMs will issue an abort in conjunction with a quench or ramp dump. The Beam Loss Monitors generate an abort if losses are too high; likewise the vacuum crates will cause the beam to be kicked out if a beam valve closes. The 200 module in the MCR accepts inputs from the manual abort buttons and the safety system.

Controls Rookie Book

Abort inputs that cause beam to be removed immediately from the Tevatron are called "Type 0" aborts. "Type 1" aborts, which are used for less urgent situations, allow the present cycle to conclude normally but issue an abort at the beginning of the next supercycle. Injection devices belong to this group.

Main Injector beam is aborted at MI-40. The abort loop uses the same fiber-optic repeaters as the other links. MECAR has an input to the system that aborts beam in the event of poor power supply regulation. The loop will also drop if an RF anode supply or the safety system trips, or if a vacuum valve opens.

QXR LINK

The VME for the Main Injector QXR is at MI-60. The QXR magnets are in the MI-30 straight section, so the one CAMAC crate in the QXR link is at MI-30. The individual cards are similar to those in the Tevatron system.

Controls Rookie Book

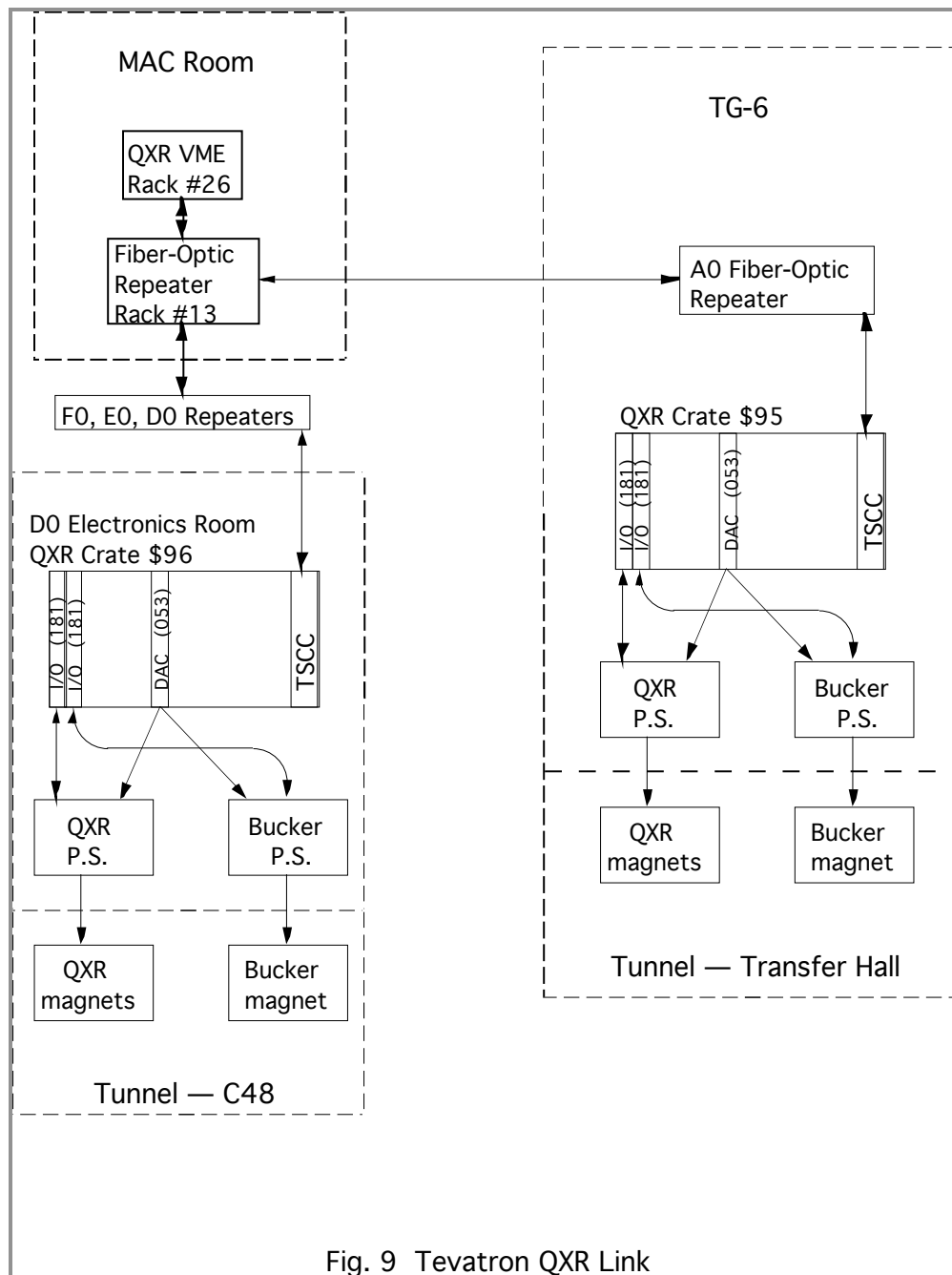


Fig. 9 Tevatron QXR Link

Controls Rookie Book

JAVA

Fermilab is constantly evaluating and upgrading its control system. The most recent trend is the use of Java applications to update, and ultimately replace the VMS rooted controls system. While not technically a link per se, Java applications are being used as a method to make our links faster and more efficient.

Java burst onto the scene in 1996 as the latest and greatest programming language. It is a versatile, easy to learn, cross-platform language. Java programs can be written for use on the Internet, programs called Applets, and stand-alone programs called applications.

Java applications are run from typical batch command files, are faster than Java applets, and have fewer restrictions when run on site. Applets are only allowed to talk to a specific web server, thus are subject to greater restrictions.

The development of these applications is unrestricted, and the integrated development environment improves the productivity and reliability of these applications. These applications make logging parameters such as connections, jobs, bandwidth, and device information an easier task. Another important consideration for using Java applications is to provide control logic execution on the server, instead of the client. The VAX/VMS control system servers only acquired data, and the user application contained the instructions for the data handling. For instance a datalogger's logic for logging and retrieving data is easier to reuse when developed as server logic, instead of single program logic. This way each client has more available resources for other tasks.

Controls Rookie Book

ACNET is a connectionless, peer to peer networking protocol. DAQ (Data Acquisition), plot, and alarm message protocols that we use everyday in our job, are built atop ACNET messages. ACNET is the protocol used by our front ends.

Java's implementation of ACNET supports multiple connections, is asynchronous, supports timeouts on single and multiple replies, supports large message sizes, supports message buffers and handling of that buffer. Java clients do not support ACNET by definition.

To match these faster, more robust Java clients with the ACNET control system we have DAEs (Data Acquisition Engines). The DAEs are the portal for client access to the ACNET control system. The Java applications connect to the DAEs via RMI (Remote Method Invocation). RMI is a portable networking protocol between Java applications.

The DAEs perform several functions. The first of these is front-end consolidation. Say you wish to plot T:IBEAM. Fine, now someone else would like to plot T:IBEAM. OK, say 50 people want to plot T:IBEAM. Can you see a problem with bandwidth developing? The front end is now trying to handle 50 different requests for T:IBEAM, on top of everything else it is supposed to do. This takes a great deal of unnecessary bandwidth and slows the network down considerably. We use the DAEs to consolidate these requests into one request before the front end sees it. The front end can now breathe easier and communicate more effectively. You may ask why it is okay to just move the bandwidth problem from the front end to the DAE. Well, the DAE is centrally located near high bandwidth links, and are much easier to upgrade than the front ends. They are not responsible for as many things as the front ends, and can therefore handle the load more effectively.

Controls Rookie Book

This consolidation does increase the complexity of our controls system by raising the number of layers of message handling, but it is necessary if we want to be able to upgrade ACNET to a more processor intensive, portable protocol sometime in the future.

DAEs operate as part of a cluster. When a DAE starts, it configures itself to be a member of the operational cluster, the development cluster, or as a stand-alone engine.

The operational cluster is a group of machines cooperatively supporting each other, front ends, and client nodes in providing access to the control system. Some of the machines in the cluster are assigned front-end consolidation duties. These engines ping each other and their assigned front ends and consolidate data acquisition across the cluster.

The development cluster is a small group of machines similarly setup like the operational cluster, but utilize independent state transitions and message consolidation.

Stand-alone DAEs are not part of a cluster. They do not share responsibility with other engines and do not participate in front end consolidation.

There are other types of DAEs. One type is the DUE (Data Utility Engine). This is a DAE with no front-end consolidation duties and primarily supports open access front-end clients, dataloggers, and servers.

A DSE (Data Server Engine) is a DAE that primarily supports application clients. A DAE with a client demanding large amounts of its processor time would be ill suited for front-end consolidation. Thus the DSE is used as a bridge between the client application and the DAE. This

Controls Rookie Book

way a DSE can be rebooted, and only the node it is located on will be affected.

Another type is the DPE (Data Programmer Engine). This is a DAE assigned to a programmer for application development purposes.

To find out more information on DAEs and their various uses and permutations go to

<http://www-ad.fnal.gov/cgi-bin/osinfo.pl>

Here you can find the status of each machine at a glance. If you click on the link for any machine it will bring you to a page full of more information than you'll probably want, and with all likelihood will never want to know. The importance of this page is the "Connect via VNC" link at the top. This link will bring up a virtual desktop connection with the machine. It allows you to use its desktop to reboot the DAE running on the machine, or the OS if necessary.

Controls Rookie Book

INTRODUCTION

Networking

In a previous chapter we talked about some of the dedicated links which allow specialized systems to perform specific tasks. Within our Controls Department, there's a group of dedicated souls who talk fluent CAMAC, PIOX, PIOR, BTR, QXR, etc. — all of which we've discussed in the Links chapter. These links are based on tried-and-true technology that has held the Lab together from day one (whether we like 'em or not). This chapter, however, is dedicated to another group within Controls that deals more with the up-and-coming world of LANs, hubs, protocols, computer-to-computer communication and, yes, the Internet; I'm speaking, of course, of the Networking group. Within this chapter we'll discuss those buzzwords, delve into Ethernet and all kinds of other fun things you can share with your closest friends.

WHAT'S A NETWORK?

Perhaps the best point to start is by defining the term... a network is nothing more than a link over which two or more semi-intelligent devices (usually computers) can communicate with each other. More specifically, in a simplistic sense, you could say a network is made of connections, communications, and services.

Connections are all the paraphernalia that's required to physically connect a device to a common network. It's the *medium* that would be used to pass messages between devices, as well as the *network interface* that physically connects the device and the medium.

Communications is the agreed-on rules by which the devices will exchange messages. Much as we, as individuals, agree to communicate in

Controls Rookie Book

English, it works equally well if some other dialect is used so long as those who are expected to respond intelligently can understand it.

Services are what generate the need to communicate. It's what enables a device to make reasonable requests or generate proper responses.

In the past each node was connected back to a central computer. If you looked at the back of our old refrigerator-sized VAXen, for instance, cables ran off hither, thither, fro and yon. Nowadays *networks* have taken over the task of getting messages efficiently from one point to another. Where, in the past, a print job, for example, would be routed from a requester, through the central computer and on to the printer, in the land of networking, the requesting node can route the job directly to the printer over the same or an adjoining network.

Networks come in two popular flavors: LANs and WANs. The LAN is the Local Area Network that encompasses individual networks — those that serve a dedicated purpose, such as the Linac VME network. Then there's the WAN, which stands for Wide Area Network. This usually refers to the site-wide threads that connect everything together so they can communicate like one big happy family. But then there's this thing called the Internet (the WAN of WANs), which makes anything else look pretty localized. So, I suppose, you could say the difference between a LAN and a WAN is defined in the perspective of the context.

If you wanted to skip a topic anywhere (or take a stiff drink) this is the perfect place; for the rest of you, go get a cup of coffee and bear with me. I'll be kind and try to make this as short as possible but this is all necessary because it covers the networking code of conduct.

OSI MODEL

Controls Rookie Book

Believe it or not, the networking world is not only governed by many sets of standards, but by a set of standards for standards. These over-all standards were developed by the International Organization for Standardization and dubbed “The OSI Model.” Why not the IOS model? Beats me — dyslexic, me-thinks.

As you can see here, the OSI model consists of seven layers, each having its own unique processing function. Too, each layer provides functions for the layer

OSI layer	Function Provided
Application	Network applications (i.e. file transfer and terminal emulation)
Presentation	Data formatting and encryption
Session	Session establishment and maintenance
Transport	End-to-end delivery reliability
Network	Delivery and routing of information
Data Link	Transfer of units of information, framing and error checking
Physical	The medium and electrical characteristics of the transmission signals

directly adjacent to it. The whole idea is that a module that operates on one layer can only be replaced with another module that operates on that same layer. For example, a mechanism which uses binary transmission would be defined on the *physical* level, however putting definition to the bit patterns would be the job of the *data link* level. Enough of that!

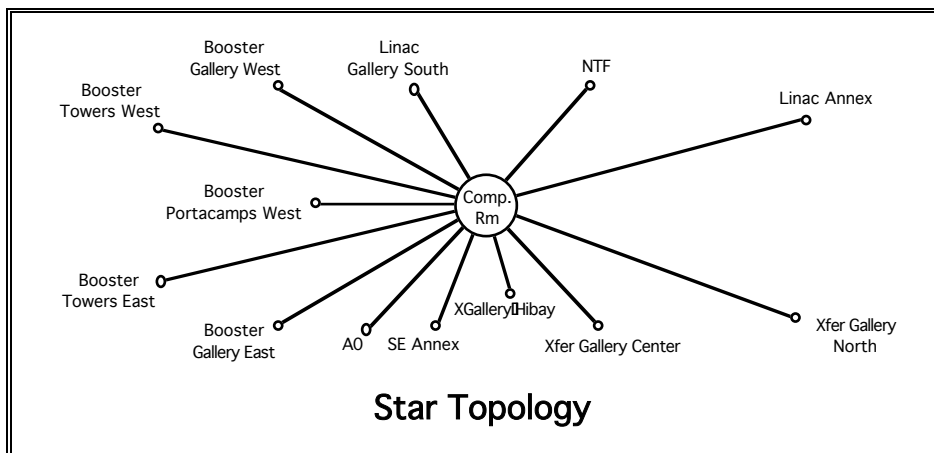
NETWORK TOPOLOGIES

Topology is just a fancy word which loosely translates to “a description or drawing of how one’s network is laid out.” In the world of networking there are three popular topology types: star, ring, and bus.

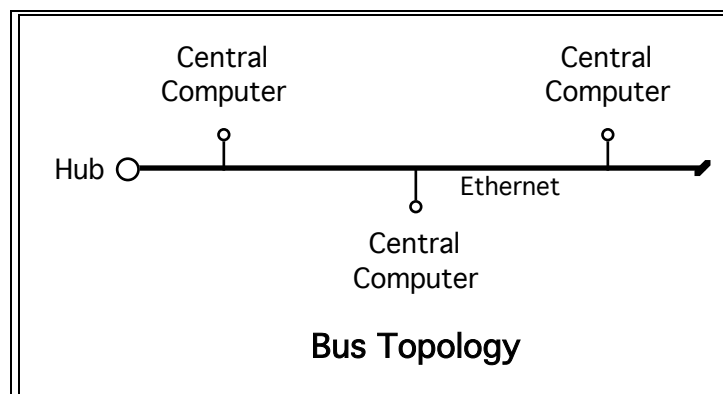
Certainly one example of a *star* network (on a large scale) is the fiber system that is the backbone for passing communications around the entire division. This system is set up such that the computer room acts as the star’s center, or hub, and it communicates with smaller nodes that conveniently reside within the desired finite range; in case anyone asks — and they won’t — that’s 2 kilometers for multimode operation. Besides the fact that all communications is efficiently transferred through one rather convenient spot,

Controls Rookie Book

another major advantage is that problems can be isolated to a specific branch of the star.



Secondly, we come to *bus* topology — the technique that simply allows the various nodes to attach directly to a common medium.



Now the bad news: in practice an overall scheme might involve one or a combination of these topologies — mixing and matching is perfectly legal. There are, of course, other topologies like ring, tree, and mesh (which are

Controls Rookie Book

considerably more convoluted) but the two described above are used most extensively in our networks.

MAKING CONNECTIONS

Getting our accelerator world connected in a coherent fashion that allows for reasonable communication involves the use of any one or more of the following cable types:

Twisted Pair (shielded and unshielded)

Coaxial

Fiber Optic

Although there are also wireless networks (like satellite and microwave), such as the one the laptop on the crew chief's desk uses, however we will not spend much time on these types of networks.

If one could take all of the cabling that comprises our accelerator networks and place it all end-to-end, do you have any idea how far it would reach? Neither do I, but it's an awesome thought. Even more awesome is how much it all costs. Obviously, one wants to use the lowest cost medium that's reasonable to work with and satisfies the traffic requirements subjected to it.

Although one just can't get much cheaper than twisted pair cabling, of the three types it's the most susceptible to electrical interference — both from the outside and also from cross-talk within its own bundle. This little flaw really hampered its use in high-speed (100 megabit per sec.) applications. Fortunately, great strides have been made in the design of twisted pair technology and this is no longer a limitation.

Controls Rookie Book

In the unshielded twisted pair (UTP) world, lunchtime chatter centers on *categories* of improved specifications. Since I really don't want this to degrade into a discussion on wire technology, suffice it to say that category 5, or "cat 5," cable meets the above speed requirements and opens doors previously reserved for the more expensive stuff. Incidentally, the cat 5 cable we use around here is pencil size and is easily identifiable by its light blue color.

For every new technology coming in (like that of UTP), there's one on its way out; such is the case of coaxial cable (or "coax"). Of course, because it outdates the improvements in twisted pair, it already exists in most every location known to man. The high-speed Ethernet traffic was once destined to travel over the thick RG-11 garden-hose-sized brightly colored cable, while the lower speed traffic used the thinner pencil-sized RG-58.

Fiber optics now comes into play when one wants to run network traffic over relatively long distances. It is typically used to pass traffic from local networks back to our computer room (as in the star network above). A fiber link has a very large *bandwidth*, which means it can carry thousands of times more traffic than, say, that of the RG-11 hose. It also has this nice feature of being immune to electrical noise. While the fiber cable is competitively priced with its coax predecessor; its use is actually governed by the significant expense of its rather challenging terminations. One only has to look at the current directions taken by telephone and cable TV industries to see that fiber optics is taking over as the medium-of-choice over coax.

A note of safety here: in your travels, should you ever come across a tiny strand of fiber optic cable, handle it like the piece of glass that it is. Fate says that snapping it is a sure way to eye surgery.

Controls Rookie Book

ACCESS METHODS (INTRO)

Techniques for passing communications over a medium-of-choice — how it moves around and how it behaves — is what's referred to as access method. In an attempt to keep your eyes from going dry, I'm going to dedicate most of this (brief) discussion to these techniques:

Ethernet

ARCnet

FDDI

There are however other access methods such as Token Ring, and ATM (Asynchronous Transfer Mode) which in the past were used in various locations around the lab. Both replaced by Ethernet, I will not devote time to explaining them in detail.

ACCESS METHODS (ETHERNET)

In the networking world, Ethernet has become the defacto standard of choice. It uses the CSMA/CD protocol, which stands for Carrier Sense Multiple Access with Collision Detect. What the heck does that mean, and how can I use it to make my life more meaningful? Simply put, a packet of data is tagged with a destination address and, at the first sign of an opening; it gets tossed onto the Ethernet freeway. On occasion two or more stations vie for the same opening or arrive at the destination at the same time and a collision occurs. In such cases the packets are simply randomly retransmitted and life goes on.

If you think this technique sounds a bit chaotic, you're absolutely correct. However, it obviously works quite well. It begins to break down when traffic gets too heavy (exceeding the media's bandwidth), then — like rush hour — things begin to bog down considerably. Typical Ethernet traffic travels at a

Controls Rookie Book

rate of 10 million bits per second (10Mbps) and, in spite of the fact that bandwidth limitations cut that rate to around 6Mbps, that's pretty darn whippy. Later we'll talk about how the addition of bridges and routers can improve the chaos considerably.

Should clever techniques not suffice however, one can move up into the realm of *Fast Ethernet* (100base-T) and jump to a 100Mbps. This is often referred to as 100base-T, whereas Ethernet implemented on **10Mbps Twisted pair** media, is referred to as **10base-T**, get it? Incidentally, the aforementioned cat 5 UTP cable, you'll recall, handles 100Mbps just fine. Here we use 100base-T for most of the connections, but the backbone is comprised of gigabit Ethernet. That would be 1000base-T if you were keeping track. This is used for all the connections between networks here at the lab, and to the Feynman Computing Center that connects us to the outside world.

ARCnet, is another access method that has its roots in the distant past but, due to its high reliability, continues to hang on even though it lumbers along at a mere 2.5Mps and is limited to distances of 300–2000 feet. It seems to be one of those methods that implement itself in a combination of topologies and methods. It acts like a bus, but everything is broadcast to all nodes at the same time (like a star) and yet uses a token-passing technique to control member communication. I mention it here only for completeness. It is not used by our Networking group, but nonetheless is used in a few places here at the Lab. For example the TeV Frig Ethernet network terminates at all zero buildings; ARCnet is used thereafter to branch to the associated numbered buildings. ARCnet is also used in some fixed-target areas.

**ACCESS METHODS
(ARCNET)**

Controls Rookie Book

ACCESS METHODS (FDDI)

Yet another technique is called Fiber Data Distributed Interface, or FDDI. This is configured like a double token-passing ring system, with messages traveling in opposite directions on each. This technique is the backbone of choice for the Computing Division's Data Communications people, mainly because it allows for nice long runs — to the village, beamlines, D0, and us — at a nice brisk 100Mbps. The high quality fiber and associated adapters get fairly expensive.

INTERNETWORKING (INTRO)

Not to be outdone by any other technology, networking is filled with a world of devices which are designed to enhance its capabilities (and, of course, make someone lots of money). As briefly as I can, I'll describe what some of these devices are and what role they play — sort of the inner workings of a network. Among the devices we'll talk about are things like repeaters, bridges, routers, firewalls and switches. If you try any of this networking magic at home, it's very unlikely that you'll need any of these toys, but for larger networks most of them are a must.

INTERNETWORKING (REPEATERS)

A repeater comes into play when one needs to extend a network's capabilities beyond the prescribed cable length of the medium being used. For the sake of confusion: CAMAC link repeaters do essentially the same thing, but they're not being included in this discussion. For the most part network repeaters are used most heavily in thick (RG-11) and thin (RG-58) Ethernet coax runs, which have optimal length limits of 500 and 185 meters, respectively. Basically, they accept information that's passed into them from one segment, amplify it, and pass it on to the next segment. A segment, by the way, is an entire cable run from terminator to terminator (i.e.

Controls Rookie Book

three computers connected by 50 feet of thin coax would be a 100 foot segment).

A repeater by any other name is still, in effect, a repeater. Thus in the 10base-T world a *multiport hub* acts as the repeater; in the ARCnet world the *active hub* is a repeater; in the Token Ring world each node acts as a repeater.

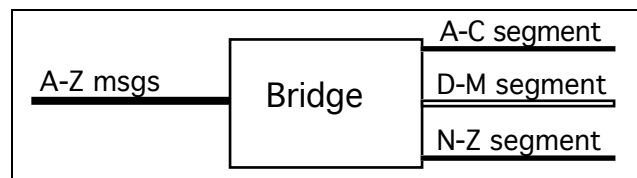
Another interesting tidbit is that it's an unwritten rule that one doesn't put more than three repeaters or hubs between two Ethernet nodes. Also, although there are gizmos that claim to extend the optimal length limits, the role of a true repeater is to simply tie two segments together.

Bridges are a lot like repeaters except they're smarter. Many have the ability to bridge the gap between two different networks like, say, Ethernet and ATM. More often, however, they're used to break a large network into two or more smaller ones.

INTERNETWORKING (BRIDGES)

Remember, in an earlier discussion, that Ethernet messages simply get broadcast to the network at large. There's a potential for a lot of traffic

looking for its assigned destination. When messages encounter a bridge (from either side),



they're quickly examined and only those messages that have a potential destination are allowed to pass. In the example drawing show here, the traffic on each of the three segments on the right side of the bridge should be significantly less than that on the left. So, for a slight knock in speed one can get a big gain in efficiency.

Controls Rookie Book

INTERNETWORKING (ROUTERS)

Whereas as a bridge can route groups of traffic based on message addressing, a router is intelligent enough to recognize traffic congestion and alternate paths of reaching the same goal. A router looks at the network as regions that have been given unique network numbers, sort of like phone number area codes. It then intelligently directs messages to these regions, invariably using other routers and bridges in the process. Much of a router's intelligence is built into its routing table, which acts as sort of a detailed map to its region. By sharing router table information with other routers, message passing can be accomplished with a fair amount of intelligence.

As you might expect, manufacturers tend to design smarter and smarter bridges that begin to rival routers and differences can start to blur, especially on smaller networks, someone nick-named these gizmos *brouters* (they need a break). Smarter routers are in the offing too which will share router table information across protocols, such as TCP/IP vs. AppleTalk — more on these two shortly.

INTERNETWORKING (SWITCHES)

Switches are designed to link several separate LANs which may be utilizing different access methods. For example, it has multiple ports, each of which can support a single node or an entire Ethernet or token ring LAN. Packets of data get switched between different LANs as needed. The switch acts like a very fast multiport bridge routing messages to their proper destination. A network designer can break up a large network into a smaller, more manageable, group of small virtual networks, each of which may have different bandwidths.

Controls Rookie Book

Going one step further in the direction of intelligent message passing, we encounter gateways. A gateway is designed to enable two radically different computer system networks to communicate with each other, acting basically as a translator. They're used also to provide a solution for bridging proprietary design differences, such as between two different email systems.

**INTERNETWORKING
(GATEWAYS)**

Taking routers to yet one more level of intelligence, one can, for a little extra change, incorporate security into this whole networking business. A firewall router is smart enough to block access to a specified network by banning traffic from unknown or undesired nodes. The most common use for a firewall is between the Internet and the local intranet, like in the case of our restricted-access Web pages.

**INTERNETWORKING
(FIREWALLS)**

There are other goodies which are part of internetworking that have nothing to do with the movement of data from one place to another and I've appropriately labeled this as OFS — Other Fun Stuff.

**INTERNETWORKING
(OFS)**

As you might expect, sniffing out problems on a high-speed network, both actual and potential, has got to be an interesting challenge. One such device is appropriately named a "sniffer." These little guys, in a more precise sense, monitor traffic patterns, watching for everything from general congestion to major pile-ups. In the hands of a hacker, they're disastrous.

Important to any network that serves a population such as ours, is on-line storage. Such equipment falls neatly into the OFS realm and is called a RAID system. RAID stands for "Redundant Array of Inexpensive Disks." In effect it's like taking a whole bunch of disk drives and treating them like one

Controls Rookie Book

humongous drive, with one important twist: the data on any one drive can be reconstructed by the others. What this means is that if a drive dies, it can be replaced without anyone being the wiser. Though the odds are slim, more than one drive could die and, still, network access could continue.



Those with PC's and Macs use a RAID system as a shared area (operations.bd). You can find it physically located in the computer room, if you take a left when you enter through the north door and walk down about 10 feet, you'll see it sitting there on the floor looking back at you.

INTERNETWORK PROTOCOLS

In order for two nodes (or people) to communicate intelligently, they need to both be conversing in a language having the same rules. In the networking world, these rules are called protocols. In the box to the left, are just a few of the more common protocols, hopefully some of them will ring a bell. I'm not going to burden you with an in depth analysis of these, but I will touch on them briefly. What will become readily apparent is that vendors, like people, tend to think their solution is the only one that makes sense.

IP
IPX
AppleTalk
DECnet
NetBIOS

“IP” stands for *Internet Protocol* and operates much like a series of letters posted through the mail. Each letter (*packet* of information) has a unique sending and receiving address, and they simply progress from station to station until the destination is reached. As such, IP is called a connectionless protocol because there's no pre-established connection between the sending node and receiving node. Incidentally, here's something to write home about: packets sent on a connectionless protocol are sometimes call *datagrams*. Our networking guru's like this technique for its ease in troubleshooting.

Controls Rookie Book

You're already familiar with IP addressing, the set of four numbers (each less than 255) which uniquely identifies every computer, printer, console, or what-have-you, to the network. They take the form 131.225.123.219, for example. An IP address roughly works like your zip code together with your box number and mail station (60510, PO Box 500, MS 306) — each of the set of numbers is used in different parts of the system to optimize delivery to a specific address. You're probably not used to referring to an IP address in its numeric form but rather in its text form, such as `mcrkeys.fnal.gov`. A *Domain Name Server* (DNS) is the network's way of translating between the two.

If you see references to *IPng*, it refers, of course, to “IP: the next generation” (who says networking types don't have a sense of humor). It's really referring to the new and improved IPv6 that incorporates a number of bells and whistles, including extensions for more numbers (they're worried about running out) and message origin authentication — those dealing with security are bouncing up and down over this one.

Going hand-in-hand with IP you will often see TCP or UDP (represented as TCP/IP and UDP/IP, respectively). Both of these methods incorporate port numbers within messages to expedite server requests. TCP takes care of the assignment and reassembly of the packets that IP is delivering, since there's no guarantee that they will arrive in the same order they were transmitted. When dealing with short messages or if this process is already being handled in other software, this overhead is not needed, which is where UDP comes in. UDP also will include a checksum to ensure packet integrity. Incidentally, TCP stands for *Transmission Control Protocol* and UDP stands for *User Datagram Protocol*.

Controls Rookie Book

“IPX” — *Internetwork Packet Exchange* — is another connectionless packet-passing protocol. It’s used in networks that incorporate Novell Netware clients and servers.

“AppleTalk” is an Apple computer protocol. It’s another one of those datagram-variety types. It associates nodes with zones — there may be multiple zones per network. Anyone who’s used the Chooser on a Mac knows about AppleTalk and zones.

“DECnet” is Digital Equipment Corporation’s answer to communications protocol and, though it’s available on our local VAXen, we no longer use it for long-distant point-to-point communication. (Compaq now owns DEC, if anyone cares.)

“NetBIOS” stands for *Network Basic Input Output System*. It was developed by Big Blue for its early PC network and then was ~~stolen~~ adopted by Microsoft. NetBIOS is used in Ethernet, token ring and Windows NT networks. Like the others, it operates in datagram mode; however, it can also be operated in *session mode*, wherein an established connection is identified beforehand. This allows for longer messages to be passed and other stuff like error detection and correction.

Before leaving the subject of protocols, I should warn you not to be confused between these and what some refer to as application protocols. Some of the more popular ones you’ve probably heard of are FTP (File Transfer Protocol), SMTP (Simple Mail Management Protocol), or HTTP (Hypertext Transfer Protocol). These, and others, describe how the contents of messages being passed over TCP/IP are to be handled. For instance, HTTP is the set of

Controls Rookie Book

rules for exchanging text, graphic images, sound, video, and other multimedia files on the World Wide Web.

The world of networking is a fast growing field. Network managers have their hands full trying to stay on top of the ever-changing technology. The never-ending struggle is to optimize each part of the system such that traffic flows through it quickly and efficiently while, at the same time, trying to blend in a reasonable balance of security, integrity, and maintainability.

Interestingly enough, here in the Beams Division, we've seen a swing from Ethernet, to token ring, back to Ethernet again and on into ATM, now back to Gigabit Ethernet, as technology continues to outdo itself. The expense of maintaining a state-of-the-art system, both in terms of equipment and training, is enough to keep network managers in tranquilizers, and yet is a necessary process.

From an Operations point of view, the network has always been a sort of "black box." This chapter was an attempt to help you understand the basics — perhaps making the box a few shades lighter. The good news is that the network performs so well that we tend to take it for granted (a testimonial to our Networking people), the bad news (from a training point of view) is that the network performs so well that there's little need for learning complex troubleshooting techniques. I suppose we can live with that.

CONCLUSION

Controls Rookie Book

Notes:

Controls Rookie Book

Clocks and Timing

Unfortunately, accelerator timing is one of those weighty subjects that is not easily described without many references to other devices and accelerators. Those who are wading into this for the first time shouldn't worry too much about detail but rather should just try to grasp the general concepts of what's going on. Later, when you've gotten a better global picture of things and your accelerator vocabulary has expanded, return and see if some of this makes a bit more sense – not a lot, but a bit.

Throughout most of the accelerators at Fermilab, beam travels near the speed of light; that is, about a thousand feet every millionth of a second. Initial attempts to coordinate these activities using a wristwatch and a joystick were dismal failures, so timing hardware was installed to trigger events when needed.

Virtually all of the timing depends in one way or another on the Tevatron Clock (TCLK — pronounce “T-Clock” in Operator jargon), named for the machine where it was first implemented. TCLK can generate up to 256 fundamental events that are broadcast to all of the accelerators, and additional timing delays for triggering equipment can be derived locally through the use of certain CAMAC, VXI, and VME cards.

TCLK can be used in a myriad of ways. Each of the accelerator regions — Linac, Booster, Main Injector, Tevatron, the Debuncher, Accumulator, and Switchyard — receive a reset event which triggers a sequence of secondary events appropriate for a particular kind of cycle. (The type of cycle can vary depending on the operating mode, so generally each machine has several different resets available to it.) For example, power supplies in the Tevatron need a 720 Hz clock to update their ramp programs. MADC

INTRODUCTION TO ACCELERATOR TIMING

Controls Rookie Book

controllers need to be able to correlate their time stamps with the hardware they monitor, and of course dipole function generators, separators, and stochastic cooling all need to know when to begin and end their activities.

In turn, TCLK uses many different sources in order to generate clock events. The “skeleton” around which most of the other events are coordinated is the Time-Line Generator (TLG), which provides TCLK with all of the basic machine resets. There are oscillators and frequency synthesizers for producing the “carrier wave” that TCLK uses as a time base, and some of these frequencies are transmitted as TCLK events. There are CAMAC cards which, when referenced to TCLK events, generate timing delays, and the resultant times can in turn be used to produce new TCLK events. There is software, particularly the sequencer and ramp generator programs, which can place events directly onto TCLK. Even our friends in the experimental areas have the capability to send us clock events generated from *their* control system.

There are some processes that TCLK, even though it can update events at almost once a microsecond, is not precise enough to handle. Beam transfers, abort timing, and certain diagnostics are some of the things which are more appropriately handled through the Beam Sync Clocks, which are derived from RF frequencies and are not calibrated in absolute time. Although they are slightly slower than TCLK, the fact that they can generate events with respect to a particular RF bucket makes them valuable for timing kickers and certain other devices. Beam Sync events are transmitted on separate links from TCLK, but most of those events depend on TCLK for their initial triggers.

Controls Rookie Book

Before diving headlong into the exciting details of the world of timing, if you haven't already done so, we highly suggest a slow stroll through the "Links" chapter — not only because timing can be thought of as a specialized link, but also much of what is discussed there is relevant background material for our subsequent discussions.

TCLK is produced by a cluster of CAMAC crates in the MAC room. These crates are all on the Switchyard link and report to the Switchyard front end.

TCLK GENERATION AND DISTRIBUTION

The processed TCLK signal is broadcast from a single-wide CAMAC 176 transmitter card in SY Crate \$06 (a \$ preceding a number represents a hexadecimal value). This card is effectively the first link driver in the distribution of TCLK through the accelerator complex. Like all of the Tevatron-style links, it is a 10 MHz serial signal transmitted using a modified Manchester code. The card requires two inputs: a 10 MHz "carrier wave" to clock the signal, and the encoded events themselves. The 10 MHz input is derived from a precision frequency DDS (Direct Digital Synthesizer) in an adjacent rack.

The desired clock events are encoded onto TCLK by a set of CAMAC 175 cards (Hardware Clock Request Modules, or HCRMs) in the same crate. There are 16 of these modules, and each one is capable of stamping 16 different events onto TCLK, for a total of 256 potential events. The HCRMs are connected in the front by an 16-line ribbon cable that terminates at the 176 transmitter card. The HCRMs put the 8-bit event, in parallel format, onto the ribbon cable, and the 176 card converts it to a serial format and sends it out on the TCLK link.

Controls Rookie Book



The modified Manchester code utilizes “cells” transmitted at a 10 MHz rate; that is, one cell every 100 nanoseconds. This compact scheme uses a bipolar signal and is defined so that if the polarity, positive or negative, does not change over the 100 nanosecond period, the bit is “0”. If there is a transition at mid-cell, the bit is “1”.

If there are no events going out on the clock, as is the case much of the time, a solid string of “ones” will be broadcast. The bit before an event will always be “0” to distinguish it from background. At that point the clock decoders, like operators on an owl shift, must wake up and listen to the incoming message. The event itself will be 800 nanoseconds (8 bits) long, to be followed by a parity bit and at least 2 “ones” to reestablish background. This means that events can be sent no faster than 1.2 microseconds apart.

The HCRMs do not have the authority to decide when a clock event should be encoded onto TCLK; they must wait for external hardware or software to trigger the event. They are, however, equipped to establish priority should two events be requested simultaneously. The cards are arranged in order of priority, determined by lemo cables in the front. Within each module, events are also prioritized.

As soon as the events are encoded, TCLK is transmitted throughout the accelerator complex. Linac (which uses TCLK to synchronize its own timing, through LCLK), and NTF, receives TCLK from the NIM fan-out repeaters in rack #13 in the MAC room. A fiber-optic repeater chassis at the bottom of this rack launches TCLK toward the Main Injector, while the copper-based

Controls Rookie Book

links of the Booster, Tevatron, Switchyard, and Pbar derive TCLK from the NIM repeaters in MCRR #30, located in the MCR's back racks.

When TCLK arrives at a service building or other point of local distribution, a CAMAC 178 module is used, via a patch panel, to fan the signal out to cards that use CAMAC. The 178 card is also capable of decoding up to 8 events, and specially modified 178 cards can generate new TCLK events from the ones that it decodes. VME and VXI crates use the same TCLK as CAMAC, but have different (and considerably more sophisticated) decoder modules.

As mentioned earlier, the HCRMs require event requests from external sources in order to encode them on the TCLK link. The most important single source of clock events is the Timeline Generator (TLG), because it creates a framework of fundamental machine resets upon which most other clock events depend.

TCLK events are each defined in terms of two hexadecimal digits (8 bits in binary). The fundamental machine resets are grouped by their first digit: "1" designates Booster, "2" the Main Injector, "3" the Switchyard, "4" the Tevatron, "8" the Debuncher, and "9" the Accumulator.

For example Event \$29 is the Main Injector reset for antiproton production at 120 GeV. When the TLG issues a \$29, many devices in the Main Injector will begin to play their tables: power supplies, RF ramps, etc. Many of these curves are reset-specific, that is, they will play different tables for a \$29 than they would for, say, a \$2B. Others are more generic. Altogether, thousands of devices will be referenced directly or indirectly to that one fundamental event. When troubleshooting, it is important to know

THE TIMELINE GENERATOR AND BASIC MACHINE RESETS

Controls Rookie Book

when, with respect to the machine cycle, a problem is occurring, and it is usually understood to be the time elapsed since the fundamental reset.

The hardware supporting the TLG consists of a VME crate in the Mac Room's Rack #15 (Actually, there is *TLGA* in rack #17 and *TLGB* in Rack #18 — normally, *TLGA* operates the timeline, but *TLGB* can be used as a backup or for training.)

The TLG requires inputs of two different frequencies for its time base: 1 MHz and 15 Hz. The DDS frequency synthesizer through a divide-by-10 circuit provides the 1 MHz. The 15 Hz used by the TLG is derived from a circuit which is directly connected to the 60 Hz at the Master Substation. A phase-lock loop divides the 60 Hz by four. The same 15 Hz feeds directly into TCLK to produce the \$0F event. (A $\times 12$ multiplier uses the same source to generate 720 Hz for use by the power supplies, but that frequency is encoded directly onto TCLK without ever seeing the TLG.)

The Tevatron machine reset events are issued almost simultaneously with the \$0F and are essentially line-locked. Booster, however, develops resets in a much more circuitous manner. The 15 Hz developed from the phase-lock loop is sent to the CAMAC BMIN module in the MAC room, which develops a 15 Hz sync. The sync signal — a square wave — is used to drive GMPS (Booster's Gradient Magnet Power Supply) without ever having seen TCLK or the TLG. When GMPS is running, a sense winding (B-dot coil) in the Booster reference magnet, and NIM electronics in the GMPS racks produce a B-dot zero crossing pulse that is delivered to the CAMAC BMIN module in the MAC Room. That pulse is passed on to the TLG that adds an ACNET controllable delay called B:RSTDLY (Booster Reset Delay). After the delay a Booster reset event request is sent to the HCRMs. When GMPS

Controls Rookie Book

is *not* running, Booster reset event requests are derived directly from the 15 Hz sync; in this case the sync pulse becomes known as the “Backup 15 Hz.” The Backup 15 Hz is also used when GMPS is running in DC mode.

In other words, GMPS is driven from a line-locked signal, without knowledge of TCLK, and Booster’s TCLK reset event is driven by GMPS. Due to variations in power line frequency and the Booster magnet circuit resonant frequency, the Booster resets may drift in time relative to the line-lock 15 Hz. Variations greater than 300 microseconds have been observed.

All of this fancy hardware is to ensure that beam arrives from Linac when the Booster magnet current is actually at a minimum (IMIN). RSTDLY is set so that Booster resets occur about 2 milliseconds before IMIN. Linac receives its reset at the same time. That gives the ion source, power supplies, and kickers time to prepare for beam.

After a machine reset successfully arrives at its local destination, it may be used by many different devices. Each of these devices may be called upon to act at different times in the cycle. The cards that most commonly generate timing delays from a TCLK event are the CAMAC 177 and 377 cards. These are “octal timers”, i.e. they are capable of supplying up to 8 independent delays. The 377 cards are similar to the 177’s, the primary difference being that the delay values (and their TCLK reference events) are stored in nonvolatile memory, making these values resistant to power outage corruption.

The 177/377 cards accept TCLK from the CAMAC 178 cards’ fanout panel. Each of the eight 177/377 channels can be programmed to decode a specific TCLK event, at which time it starts a countdown timer. The timer is

**TIMING DELAY
HARDWARE**

Controls Rookie Book

also programmable through ACNET, and a secondary event is generated at the end of the countdown. This secondary pulse can be used to trigger local hardware, or if used in the MAC room, to create new TCLK events. The 177 timers can operate at 1 MHz, 100 KHz, 10 KHz, or 1 KHz for a total of 64K counts. For the mathematically challenged: at the slowest rate, the cards can delay an output for about 65 seconds. The 377 cards, on the other hand, can delay an event for up to 1.2 hours.

Ramp generator cards — for example, the 165 or 465 — have built in timetables that can be used to play out their waveforms. These timetables are usually triggered by a TCLK event. Even if, as is often the case, the waveform tracks beam energy via MDAT, the energy waveform is initiated by a TCLK event.

Believe it or not, the following is a *simple* example describing the way in which TCLK events and delay timers can work together to synchronize the various machines.

S:V100 powers a magnet string in the Switchyard, which effectively acts as a switch to the Neutrino and Muon (i.e. KTeV) experiments. When the string is powered, the beam passing through is directed to the experiments; when it is off, beam continues straight ahead until it encounters the Switchyard dump. As with most Switchyard magnets, to keep it at full current at all times would be prohibitive in terms of power consumption and cooling requirements, so it is only powered during those intervals when beam is expected. The full Tevatron fixed-target ramp requires about 60 seconds, but — since beam will only be extracted during the 23 seconds of flattop — the Switchyard magnets, only needing to be powered at flattop, spend less

Controls Rookie Book

than half of that time working. The ramp tables for S:V100 are stored in a CAMAC 165 card, which is located in Switchyard Crate \$B6 at SSB.

There are several dozen magnets in Switchyard, as well as SWICs, BPMs, and loss monitors, which are active over approximately the same interval. The most efficient way to coordinate their activities is to issue a single clock event a few seconds before the start of *Tevatron Flattop*. Tevatron flattop (using numbers from previous runs) occurs at 19.779 seconds. Event \$31, the Switchyard reset, appears on TCLK at 13.779 seconds, which gives the various devices six seconds to prepare for beam. Like all TCLK events, it is initially transmitted from the CAMAC 176 card (in the MAC Room). The event is broadcast via the link repeaters until it finds its way to Crate \$B6 (among many others), where it is decoded by an alert CAMAC 178 card. From the 178 card it is distributed to interested customers in the crate, including S:V100. Upon receipt of Event \$31, the card begins to play out its waveform to the power supply. Six seconds is an eternity in the world of accelerator physics, so the card waits, in this case, 4.8 seconds before instructing the power supply to produce current. That leaves 1.2 seconds to bring the current to full value and to let it stabilize. If more time is needed, the waveform on board the card can be adjusted without impacting any of the other supplies. The total length of the power supply's waveform value is set up to correspond roughly to the interval of the Tevatron's flattop.

Simple trigger, simple response. But where does Event \$31 actually come from? It is stored in one of the CAMAC 377 cards in the MAC Room; the 377 timer is referenced, or triggered from, Event \$41 (Tevatron reset), which occurs more or less simultaneously with the beginning of the supercycle. The 13.779 seconds represents the delay from Event \$41.

Controls Rookie Book

How does the 377 card know that Event \$31 should occur at 13.779 seconds? Remember that clock events can come from a variety of sources. In this case, it is calculated by software — specifically, by console page C49, which is the Tevatron ramp generator program. This program makes a hideous number of calculations, and some of them determine clock events associated with the Tevatron ramp. C49 not only calculates Event \$31, but also Event \$41 (*Start of Tevatron Ramp*), Event \$45 (*Start of Flattop*), and Event \$46 (*End of Beam Operations*), among others. The calculated values are downloaded into the 377 cards, in the Mac Room, when the ramp is sent to hardware. Event \$31 is calculated along with the other ramp times so that if the ramp timing is changed, the Switchyard reset is automatically adjusted accordingly.

Other TCLK events may come from other sources. What they all have in common is that they are broadcast throughout the accelerator complex, and that they can be used by any number of devices.

SOFTWARE GENERATED TCLK EVENTS

Other TCLK events are generated directly by software, without going through a predetermined hardware delay. The Main Ring and Tevatron abort programs, I67 and T67, issue a clock event when an abort is reset in one of their respective machines. TECAR issues an event when a quench is detected. STATLOG issues an event that prevents Switchyard devices from ramping if there is no beam in the Tevatron.

By far the most prolific producer of direct software-generated TCLK events is the Tevatron “sequencer”. It creates about a dozen events, including the opening and closing of the helix, and those that initiate the

Controls Rookie Book

“Goto Low Beta” and “Un-Squeeze” aggregates. These events can be completely asynchronous to the TLG.

The following are a number of miscellaneous hardware sources for TCLK. The faint-of-heart can skip over this and return when life doesn't appear so threatening.

TCLK HARDWARE SOURCES

- LBOE: This unpronounceable acronym stands for “Little Bit Of Everything”. It is a CAMAC card in the MAC room. It generates about a dozen TCLK events that are not necessarily related to each other; it also provides time base services for the 070 profile timer cards. Some of the TCLK events produced include gating for 8 GeV extraction logic, BPM “Prepare For Beam” requests, and embarrassing proclamations of quenches in the Tevatron or low beta strings.

- BSSB: This stands for “Beam Switch Sum Box”. This is the chassis in the Main Control Room that coordinates the various beam permits with the clock events that request beam. If the beam permits, beam switches, and TCLK events are all in agreement, the BSSB sends a request to the pulse shifter to accelerate beam. Clock events are issued when HEP beam is either permitted (Event \$52) or inhibited (Event \$53). Events are also issued by the BSSB whenever aborts are detected in the Main Ring or Tevatron.

- There is a NIM crate in Pbar which issues a clock event to trigger the circular buffers of crucial power supplies in the Accumulator in the event of sudden beam loss.

Controls Rookie Book

- Beam Sync Clock: When certain Beam Sync Clock events are detected, a TCLK “reflection” event is issued. There are applications programs — the Tevatron sequencer is an example — that use the TCLK reflections as evidence that a particular beam transfer is supposed to have taken place.

INTRODUCTION TO BEAM SYNC CLOCKS

The beam sync clocks are different from TCLK in that they count RF cycles rather than units of absolute time. This is because certain accelerator functions, such as beam transfer from one machine to another, require that they be executed with respect to a certain “place” in the string of RF buckets. For example, kickers need time to get to full current, and this is best done when there are gaps in the beam. Injection, extraction, and abort kickers can use the beam sync clocks to effectively count the number of bunches so that their rise-time occurs during the gap, allowing them to arrive at *full* current when it *is* desired to kick the beam, and so that there is *no* current when it is *not* desired to kick the beam. Altogether, transferring beam through several different accelerators requires considerable synchronization.

There are several different beam sync “clocks,” each with its own link repeaters: Main Injector (MIBS), Recycler (RRBS), Tevatron (TVBS), and Antiproton/Tevatron (APTVBS). However, in practice almost everything will use MIBS or RRBS. MIBS is the one that is beginning to function at the time of this writing, and will be the one described in this section.

The bulk of the MIBS hardware resides in the Main Injector RF Control Room at MI-60 — specifically, the CAMAC cards in Crate \$94 that work with the LLRF VXI. The VXI, which of course knows what is happening with the LLRF, generates a beam sync clock “tick” which corresponds to every 7th RF bucket produced. There are 588 RF buckets in a single

Controls Rookie Book

revolution around Main Injector, so there are 84 ticks per revolution. Since the MI LLRF frequency is roughly 53 MHz, the frequency of the beam sync clock ticks is roughly 7.5 MHz. (This is in the ballpark of the TCLK base frequency, 10 MHz, so much of the hardware is built from modified TCLK hardware.) Since the velocity of the beam in Main Injector changes slightly over the course of an acceleration cycle, the frequency of the beam sync ticks, measured in absolute time, changes as well.

The VXI hardware also produces something called the “AA” marker once every revolution — that is, once every 84 clock ticks. Each of the ticks can then be referenced with respect to its position relative to the marker. Another way of looking at this is to think of every 7th bucket having a “label.” If action needs to be taken against some specific section of the beam, that section can be located by waiting for the “AA” marker and then counting the appropriate number of clock ticks. It would be next to impossible to do this by trying to measure absolute time at widely separated parts of the accelerator complex.

Popular uses of beam sync clocks include beam transfer from one machine to another, abort timing, and triggering of beam diagnostics.

Main Injector Crate \$94 houses several CAMAC cards used to generate MIBS; Crate \$95 holds the equivalent cards for RRBS. Continuing the story of MIBS, all of the events are broadcast from a CAMAC 276 card in Crate \$94, analogous to the 176 transmitter card used by TCLK. A single CAMAC 175 card encodes the “AA” marker and all of the specific MIBS events onto the 276 card, and link repeaters carry the encoded signal to the remote locations where it will be used.

Controls Rookie Book

Each beam sync event is initiated by an appropriate TCLK event. There are three CAMAC 377 cards in Crate \$94 which store the delays from the TCLK triggers to produce the specific beam sync events. CAMAC 279, 379 or 479 cards, can decode these events. These cards add a delay measured in “Main Injector Revolutions” (MREV with 84 clock ticks making up one revolution. The 479 cards also have the capability of adding a delay measured in absolute time (i.e. nanoseconds) to the delay measured in Main Injector revolutions.

BEAM SYNC APPLICATIONS



• It is time for an example — the following discussion will show how the TLG, TCLK and MIBS can all work together. One of the several beam sync events is MIBS \$79, which coordinates Main Injector-to-Debuncher beam transfer during antiproton production. There are widely dispersed sets of kickers and pulsed septa (at MI-52, AP0, and AP50) that must all fire at the precise moment when beam is present. The 1.6 microsecond slug of beam only passes by once, and the kickers can be neither early nor late.

The groundwork is laid by the TLG, which produces two fundamental TCLK events at the beginning of each stacking module: \$29 and \$80. Both events, generated in TLGA, are virtually simultaneous, occurring within a few microseconds of each other. An instant after the events have been issued, they are being decoded by the local hardware.

The \$29 is the Main Injector reset, and initiates the sequence of events that governs everything that happens to the Main Injector during the acceleration cycle. The cycle culminates 1.4 seconds later, with circulating

Controls Rookie Book

120 GeV beam. The \$80 is the Debuncher reset, and prepares it for the beam that is expected to arrive at 1.4 seconds.

Event \$80 is fanned out to the appropriate CAMAC crates by the 178 cards at MI-52, AP0, and AP50, and from there it is passed on to the kicker hardware. Upon learning that beam may be present in 1.4 seconds, the kickers begin charging their capacitors for the anticipated pulse. Event \$80, a TCLK event, is sufficient because the timing constraints for this part of the task are not stringent.

Timing constraints for the kicker firing trigger *are* stringent, so Event \$80 is also intercepted at MI Crate \$94, and it is here that the beam sync event will be generated. MIBS \$79 (not to be confused with any similar TCLK event) comes from the 377 timer known as I:MIPBTX. When the timer detects Event \$80, it begins a 1.4 second countdown. The resulting pulse, which occurs during Main Injector flat-top, is encoded onto the 175 module as MIBS \$79, and broadcast to the world from the 276 transmitter card.

Meanwhile, Crate \$94 has also been producing the “AA” marker and beam sync clock ticks. This it does continuously, day and night, whether or not anything special is happening. There are 479 cards at MI-52, and 379 cards at AP0 and AP50, which quietly listen to the MIBS signal. But these particular cards are hardwired to respond to MIBS Event \$79, and when it appears on the link at 1.4 seconds they “arm” themselves to fire their respective kickers. Once armed, the card waits for the next “AA” marker. When the marker is detected, it is simply a matter of counting the appropriate number of MIBS clock ticks before pulling the trigger.

Controls Rookie Book

The kicker pulse at all three locations should ideally be accurate to within a single bucket. The reason that it can be, in spite of the 7-bucket resolution, is that the position of each bucket with respect to the “AA” marker was defined by the LLRF systems and MIBS as soon as beam was transferred from Booster into the Main Injector.

Some of the beam diagnostics, such as loss monitors, can get away with using TCLK events to trigger their activities. Others, such as flash frames and multiwires, must rely on beam sync events in order to be active during the instant when beam is present.

MIBS events govern transfers involving the Main Injector and the Booster, Debuncher, Accumulator, 120 GeV extraction, or the Tevatron. Operations involving the Recycler will depend primarily on RRBS.

Controls Rookie Book



Controlsopedia

Ok, why the strange name for this chapter? Initially it was just going to be called a (gasp) *glossary*, ugh! As you scan through this collection, you'll see that that conveys the wrong image. A glossary is something one refers to when they run across a term or phrase in their normal reading that they need clarified. This chapter *is* a glossary, true, but it is more than that. Since, you'll actually find no reference to many of these terms anywhere else in this book, I suppose, technically, one could call it a *Reference Manual*, but that was too, um, techno-stuffy. Perhaps *Controlsopedia* isn't much better; oh well, the idea however, is that one would just not consider actually *reading* a chapter that resembled a dictionary, and yet that (for the most part) is exactly what you should do here.

To clarify: I'm sure you've discovered, *controls* is a myriad of odds-n-ends that glue our accelerator together and make it work. Operators play a unique role. They're often the ones to make the initial diagnosis of an accelerator ailment and then are able to make the repair or comprehensively describe the symptoms with an appropriate technician or engineer. That requires at least a minimal amount of knowledge on a vast amount of things. Therefore, although one could write individual chapters on all these topics, these are the ones voted to "keep short and to the point."

THE INTRODUCTION

Controls Rookie Book

The topics have been listed alphabetically, which allows for quick look-up of a term someone's recently sprung on you, however, unlike a normal chapter, it won't be a progressive read; that is, you may encounter terms within a term to which you've not yet been introduced — the relevant ones of which will appear in bold typeface. We think you can handle it. The big test comes right away in that you'll notice all the popular crate modules appear at the beginning. **Please feel free to jump past these on your initial read and refer back to them as the need arises.**

For convenience, a checkbox has been placed in front of each topic so you can feel free to jump around as you please, simply check off where you've been.

Finally, be forewarned that this chapter (and this book) will *always* be in a state of flux; terms and concepts will disappear tomorrow only to return another day in some other obscure form, sort of like a flu virus. Having said that, please be encouraged to email suggestions to your Controls Specialist on new terms to be added, or old ones to be modified.

Controls Rookie Book

CAMAC Module Comments: The following are but a small subset of the CAMAC modules used throughout the accelerator. In order to make a new Operator's life a tad easier, only those modules that our panel of experts deemed "noteworthy" have been listed. In addition, for the true rookies among us who don't want to embarrass themselves at large social gatherings, any module beginning with zero is pronounced as a long "O". So a 080 module, for example, is pronounced "Oh Eighty".

THE CONTROLSOPEDIA
- MODULES 0-9 -

- ❑ **007 Module:** A beamlines analog-to-digital (24-bit input) interface module. (*See also these other important analog readback modules: 190 and 1010.*)
- ❑ **053 Module:** 4-channel DAC- power supply controller.
- ❑ **057 Module:** A stepping-motor controller module. (*See these other motion control modules: 014, 044, 154, and 1045.*)

See also: www-bd.fnal.gov/controls/camac_modules/c057.txt

- ❑ **080 Module:** A **parallel I/O interface module.** The interface module between a microprocessor and its associated CAMAC crate (such as in our BPM system). Some microprocessors use one parallel bus (Multibus) whereas the CAMAC Dataway is a different parallel bus. (*See also 488 module.*)
- ❑ **145 Module:** A Vacuum readback interface module. (*See also 170 Module*)
- ❑ **151 Module:** This is a beamlines power supply controller module sporting an internal/external analog reference. (*See these other important power supply communication cards: 160, 165, 365, 453, 465-468, and 1151*)

Controls Rookie Book

- ❑ **165 Module:** A power supply controller for many devices. The ramps are loaded from the MCR. The ramp is of the form:

$$V_{out} = \text{Scale factor-table value} * \text{ring energy}$$

The scale factor is set by the D/A value entered by an Operator on a parameter page. The table value is a time-dependent multiplier set from a 165 control page. The beam energy is represented by MDAT and defaults to full scale if not specified. *(See these other important power supply communication cards: 151, 160, 365, 453, 465–468, and 1151)*

See also: www-bd.fnal.gov/controls/camac_modules/c165.txt

- ❑ **166 Module:** MDAT transmitter card.
- ❑ **169 Module:** MDAT receiver card.
- ❑ **170 Module:** A CIA crate vacuum controller. *(See also 145 module)*
- ❑ **175 Module:** Encoder for TCLK. This module generates an eight bit parallel signal, known as a “clock event.” This is then shipped to the TCLK transmitter, (a 176 module) via a front panel ribbon connector, where it is converted to a serial transmission. Each 175 module contains 16 prioritized channels. The event is generated whenever a channel receives an external trigger.
- ❑ **176 Module:** TCLK transmitter module. This module takes the clock events generated by the 175 module and encodes them for the serial link.
- ❑ **177 Module:** A time delay module for many devices around the accelerator. Each module has 8 channels that may be triggered independently, and each may be referenced to as many as 15 TCLK events. Each channel has a programmable delay ranging for 1 microsecond to 65.535 seconds. Upon receipt of a trigger each channel

Controls Rookie Book

that is enabled outputs a TTL pulse that may be used to trigger any other device. (*See also these related timer modules: 178, 277, 377, 379*)

See also: www-bd.fnal.gov/controls/camac_modules/c177.txt

❑ **178 Module:** A TCLK repeater/decoder module. This module decodes the TCLK events and fans them out to the rest of the crate. (*See also these related timer modules: 177, 277, 377, 379*)

❑ **190 Module (290):** A module that interfaces the MADCs around the accelerator to the controls system. It can support up to 128 channels, and is capable of supporting up to 6 plots at a 2.1 kHz rate or a single channel at 70 kHz. It is able to determine which of the devices under its care are in an alarm state, and can decode events on the accelerator clock system. (*See also these other important analog readback modules: 007 and 1010.*)

❑ **200 Module:** *Abort Concentrator Module.* This card accepts up to 8 inputs from devices in a given service building. If the permit signal originating from a device disappears, an abort is generated. Most frequently used for the TeV abort link.

See also: www-bd.fnal.gov/controls/camac_modules/c200.html

❑ **201 Module:** *Abort Generator Module.* This module generates the 5Mhz signal broadcast as the abort link.

❑ **204 Module:** A programmable beam permit module that allows beam permits to be issued under different sets of circumstances.

❑ **269 Module:** A Main Injector power supply link receiver module.

See also: www-bd.fnal.gov/controls/camac_modules/c269.html

❑ **278 Module:** A Beam Sync Clock decoder/fanout module.

Controls Rookie Book

- ❑ **279 Module:** Provides a programmable delay from a TCLK or beam sync event. The passage of time is measured directly in terms of the number of beam revolutions. A clock “tick” represents 7 RF cycles so, in the case of the Tevatron, for example, there are 159 ticks per revolution. The clock operates at approximately 7.5-MHz and is capable of delaying a signal up to 412 turns.

See also: www-bd.fnal.gov/controls/camac_modules/c279.html

- ❑ **290 Module:** An updated 190 module that allows for more plot channels and lists.

- ❑ **377 Module:** A TCLK octal timer module (an improved 177). (*See also these related timer modules: 177, 178, 277, 379*)

See also: www-bd.fnal.gov/controls/camac_modules/c377.txt

- ❑ **379 Module:** A Beam Sync Clock octal timer module used heavily by MDAT (an improved 277). (*See also these related timer modules: 177, 178, 277, 377*)

- ❑ **453 Module:** A waveform generator / power supply controller, primarily used for Main Injector correction elements. (*See these other important power supply communication cards: 151, 160, 165, 365, 465–468, and 1151*)

See also: www-bd.fnal.gov/controls/camac_modules/c453.txt

- ❑ **460 Module:** The card was designed to be used for the Tevatron dipole correction elements. In this usage the output tracks the Tevatron magnet current as represented by the MDAT signal (as well as the low beta sequence step) and may contain a “time-bump” which changes the dipole current with respect to the time in the accelerator cycle. These cards are also used for

Controls Rookie Book

Tevatron RF curves, flying wires, abort kicker waveforms, and some injection magnet waveforms.

(See these other important power supply communication cards: 151, 165, 365, 453, 465–468, and 1151)

- ❑ **465–468 Modules:** Waveform generators / power supply controllers. *(See these other important power supply communication cards: 151, 160, 165, 365, 453, and 1151)*

See also: `www-bd.fnal.gov/controls/camac_modules/c465.txt`

- ❑ **488 Module:** A common **GPIB** interface module. *(See also 080 module.)*
- ❑ **1010 Module:** This beamlines module controls and reads data from an MADC. It has a TCLK decoder and onboard data pooling is supported. It is ARCnet link capable. *(See also these other important analog readback modules: 007 and 190.)*
- ❑ **1045 Module:** A beamlines motion-control chassis interface module. *(See these other motion control modules: 014, 044, 057, and 154. See also the 080 and 488 interface modules.)*
- ❑ **1151 Module:** A precision Beamlines power supply readback and control module. *(See these other important power supply communication cards: 151, 160, 165, 365, 453, and 465–468)*

- ❑ **19-Conductor Cable:** Those signals deemed worthy enough, are allowed to use this high-speed Helix cable that extends around the Tevatron. This cable was once used to return data from an underground nuclear test site in Nevada and has since become a can't-miss target for all backhoe operators everywhere.

THE CONTROLSOPEDIA

- 0-9 -

Controls Rookie Book

THE CONTROLSOPEDIA

- A -

- ❑ **Abort Concentrator Module:** A CAMAC **200 module** in the Tevatron abort system capable of accepting up to 8 inputs from devices in a given service building. If the permit signal originating from a device disappears, an abort is generated.
- ❑ **Abort Link:** Also known as the *abort loop*, this is the **hardwire** system dedicated to removing beam from the appropriate accelerator(s) in its never-ending quest to protect personnel and equipment.
- ❑ **Abort Link Generator Module:** A C201 card which generates the 5-MHz abort permit signal.
- ❑ **Abort Reset Command:** A command sent from the MCR that clears the latched abort status and restores a beam permit.
- ❑ (1) **ACNET [System]:** The acronym for Accelerator Control NETwork. This term, which is, more often than not, used without its *system* qualifier, occasionally causes some confusion, which will be described below. The term, in this context, refers to our entire *system* of computers, links and the software required for the monitoring and control of all the accelerators. When one says a given crate is part of ACNET, they're really saying it's an integral part of the accelerator control system.
- (2) **ACNET:** A VMS-integrated software communications package (and its subsequent protocol) written here at Fermilab by Frank Nagy. It is the means by which the **central computers, front ends, consoles** and various other subsystems communicate. If ACNET were a more visible thing, it'd look similar to the US postal system where various packages, all of unique size and shape, are distributed over a vast network but all having to adhere to the same addressing, packaging and size protocol.

Controls Rookie Book

Likewise, if you abide by the rules, ACNET will deliver the message, if not... well, try to drop a package adhering to the FedEx protocol into the US Postal System and see how far it gets.

So where can you find this ACNET software entity? Look on each **central computer** and you'll find a process called **NETPROCESS**. This, my friend, is ACNET. Shut down this process and your main communications system ceases to be; along with it go all of its associated processes that were depending on it to communicate. Just as you would do if your phone line went dead in the middle of a conversation, you'd hang up — so do they.

The confusion, that I spoke of earlier, comes when someone talks about “restarting ACNET,” they're *speaking* in the terms of the latter context, but they often *think* in the terms of the previous. That is, many mistakenly visualize that bringing down (and restarting) ACNET brings down a physical network, including all the central computers, whereas ACNET (NETPROCESS) is nothing more than a process running *on* a given computer.

Historically ACNET (the software package) was designed because there was no commercially available network communications product(s) that was lean and mean enough to meet the stringent requirements of this rather unique application — running our accelerator. Although commercial packages have improved, ACNET has also evolved and continues to do a faithful job of serving our needs. Not that it won't be abandoned one day, but it won't be trivial as its roots run deep (*very* deep) into the **VMS** operating system.

□ ACNET Console User's Guide:

See www-bd.fnal.gov/controls/consollesii.html

Controls Rookie Book

- ❑ **Acknowledgeable Alarm:** This refers to an **alarm screen** message that requires a manual “recognition” interrupt.
- ❑ **ACLKWATCHER:** A process on the VAX that decodes TCLK events and generates timing information for internal consumption (for such things as the frequency of data acquisition).
- ❑ **A/D, ADC:** (1) In hardware this is an Analog-to-Digital Converter which converts a voltage presented at its input to a binary representation of that voltage on its output for use by the control system. Most A/D’s in the control system have a measurement resolution of the less than 5 mV and accept input voltages in the range -10.23 to +10.24 volts. In some applications (Linac PS regulation) special units are used which have a resolution of less than 1.25 mV.

(2) In software, an A/D is the Analog readback of a given device.
- ❑ **AEOLUS:** A predecessor to our current **ALARMS** system.
- ❑ **Aggregate Control:** A command used to control the digital status of multiple devices. For example, one might reset, turn on or turn off all the power supplies in a given accelerator, with one interrupt.
- ❑ **Alarm:** A controversial term referring to a message, usually generated by the **ALARMS** software, indicating that the digital or analog status of a device is not within the tolerances set for it.

What makes this term controversial is the term “alarm” as used in the context of an *emergency* situation requiring an immediate manual response? In the accelerator world, conditions that would harm personnel or equipment happen far too fast for one to manually react. Such is the job of our **hardwired** abort and safety systems — a software system is inherently unreliable for such notification and therefore will never be incorporated as

Controls Rookie Book

the primary reactive solution. Incoming status on our “Alarm System” is not real time nor, in fact, is it guaranteed. Important facts to remember. (*See also: Abort Link.*)

- ❑ **ALARM Annunciation:** Alarms are now annunciated from console 7 in the MCR through a connection with the PC to the right of it (MCRTV) that is connected to the speakers in the MCR.
- ❑ **ALARMS:** A set of processes on the VAX which collect device status information from the front ends, combine that information with appropriate parameters in the database, and ultimately send the package to the console CPUs for display (or anyone who’s particularly interested). If you’re interested in a more detailed description of the **ALARMS** processes, go to the “Heartbeat Users” page (D130) and under abstracts, select (you might have guessed) **AEOLUS**.
- ❑ **Alarm Screen:** Sometimes more accurately referred to as the **Status Display Screen**, this color display is actually an ACNET window, the strategic location of which is (normally) the upper right-hand area of each MCR ACNET **console**. The location is historical — that’s where the display was when it was a dedicated monitor — but now it exists there simply for console-to-console consistency reasons. The purpose of the display is to interactively list devices that have exceeded a user-defined preset limit. See the discussion on **Alarm**, above.
- ❑ **Alarm System:** The integrated system through which out-of-tolerance conditions reach the **Alarm Screen** from devices in the field. The front ends each have their own way of generating “alarm” codes that are then forwarded to the **ALARMS** processes in the VAX. These processes, in turn, obtain information from the **database** before sending the combined package to the MCR console display.

Controls Rookie Book

- ❑ **Application [Program]:** A computer program designed to accomplish a specific task. All of our console pages are considered “application programs.”
- ❑ **Archive:** Permanent storage of information regarding a given accelerator system. Magnet tape is the primary medium. An archive should be distinguished from a “Save,” where information is written onto a disk and is likely to be written over at some future date.
- ❑ **ARCnet (Attached Resource Computer Network):** A slow (2.5Mbps), but simple computer networking access method still in use in various places around the accelerator (such as **SRM** communication in the Linac). It was developed back in the horse-n-buggy days (1968). See the chapter on *Networking* for more information on ARCnet.
- ❑ **Assembly Language:** A programming language that is once removed from a computer’s **machine language**. Assembly language consists entirely of numbers and mnemonics that have the same structure as machine language and difficult (ok, insane) to read and write, which is why programmers resort to the more comprehensible high-level languages like **C** or **FORTRAN**.
- ❑ **ATM (Asynchronous Transfer Mode):** A network technology based on transferring data in cells or packets of a fixed size. The cell used with ATM is relatively small compared to units used with older technologies. The small, constant cell size allows ATM equipment to transmit video, audio, and computer data over the same network, and assure that no single type of data hogs the line.

Controls Rookie Book

Current implementations of ATM support data transfer rates of from 25- to 622-Mbps (megabits per second). This compares to a maximum of 100-Mbps for **Ethernet**, the current technology used for most LANs.

See the chapter on *Networking* for more information on ATM.

- ❑ **Backplane:** A circuit board or frame containing sockets into which other circuit boards can be plugged in. The back of a **crate** is probably the ultimate example. You could also use the term *motherboard* without raising too many eyebrows, although there are subtle differences.
- ❑ **Back Racks:** Electronics racks behind the primary working region in the Main Control Room; consists of patch panels, link modules, hipotters, and much more.
- ❑ **Backup:** See *Archive*.
- ❑ **Basic Control:** The ability to change the digital status of a device by interrupting on a parameter page. This function is supported by the database. (*See also Basic Status, below.*)
- ❑ **Basic Status:** Digital status of a device as displayed on a **parameter page**. There are typically three characters displayed in a standard format (. . .) at the end of each device line. Normally, the first indicates ON/OFF status; the second indicates the trip/reset status; and the third indicates if the supply is in local or if the breaker has tripped. If all of the digital status for a device is good, three green dots will be present. Some devices have a fourth bit of +/- usually (but not always) to show polarity.
- ❑ **Batch [processing]:** The execution of a series of non-interactive jobs all at one time. This term originated in the days when users would give a batch

THE CONTROLSOPEDIA

- B -

Controls Rookie Book

of programmed cards to a system operator, who would then feed them into the computer.

❑ **Baud:** A unit of signaling speed; defined as the number of code elements (i.e. bits) per second.

❑ **BBM (Beam Budget Monitor):** Before beam can be accelerated in any given accelerator, (shielding assessment) studies and mathematical models are used to determine the safe intensity range or *operating envelope*, given the current conditions. As one of the MCR **comfort displays**, the BBM gives Operators a graphical view of current beam intensities vs. a given accelerator's operating envelope. It's of particular interest following a **timeline** adjustment.

For example, entering a large number of beam events for a particular experiment may well benefit that experiment but, on the BBM, the dramatic climb of their respective integrated intensity will begin to show that continued operation under those conditions will eventually exceed the permitted limit. A subsequent adjustment to the timeline is usually all that's needed to bring things back in line. A definite, but admittedly more drastic approach is simply turning off the beam switch.

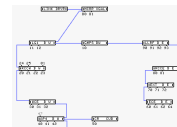
Exceeding the operating envelope is serious business — don't do it! It's not safe and accelerator laboratories are shut down for such things.

❑ **Beam Switch:** A device used to manually permit/inhibit beam from entering a particular accelerator. This switch only disables the Linac pulse shifter from shifting to allow beam from the Proton Source. For instance, Pbar transfers of antiprotons to the TeV during shot setup can be done with the switch taken. Because of the variety of modes in which we

Controls Rookie Book

run, this seemingly simple action actually requires a high degree of logic — such is the purpose of the **Beam Switch Sum Box**.

- ❑ **Beam Switch Sum Box:** The hardware logic circuit that determines the response of any **beam switch** request, based on machine modes and current operating conditions.
- ❑ **Beam Sync Clock:** The clocks MIBS, RRBS, and TVBS, for which the basic frequency is directly related to the revolution frequency of the beam. They are derived from their respective low-level RF systems. They operate at frequencies of about 7.5-MHz and produce a clock “tick” every 7 bunches. Beam sync clocks are used for all critical timing of beam transfers between accelerators. For more information, see the chapter on *Timing*.
- ❑ **Bit:** This stands for a Binary digIT. It may have a value of 0 or 1 and is the smallest unit of binary measure. (*See also byte and word.*)
- ❑ **Block Transfer (BTR):** An option for data transfer in some accelerator links. Although data is transmitted serially, it arrives at the front end in a continuous stream rather than in discrete units. Typically, a block of memory is established beginning at a specified location and data is then streamed in to fill it. The BTR link is on a separate cable from the CAMAC serial link. It is able to transfer information at a 10-Mbit/sec rate in blocks 196-Kbytes in length.
- ❑ **Booster Serial Link:** This is the CAMAC link to which all Booster CAMAC crates are connected and through which control cards communicate with the Booster front end. For the most current map, see CAMAC Link Status page, D20.

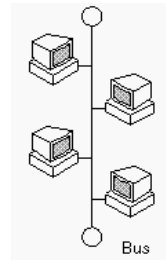


Controls Rookie Book

❑ **Boot:** Restarting a computer by means of resetting its internal registers to a known condition then loading a fresh copy of the managing programs from nonvolatile memory. The system may be thought of as loading itself by its bootstraps. A *reboot* is an abbreviation for “remote boot.”

❑ **Bridge:** A networking device that is used to pass information efficiently between two networks or to break up a large network into two or more smaller ones. See the chapter on *Networking* for more details.

❑ **Bus:** An electrical medium between two or more components or systems. Not to be confused with “Buss,” NASA’s Biomedical Urine Sampling System, nor “Buss Bar,” the connection strips within an electrical service panel. However, it is fair game for being confused with “busses” — the plural for bus.



❑ **Bus [Networking] Topology:** See **Topology**.

❑ **Byte:** A byte consists of 8 bits. It may have a signed value of -128 to +127 or an unsigned value between 0 and 255. (*See also bit.*)

❑ **C [Language]:** Developed by Dennis Ritchie and Brian Kernighan at Bell Labs in the mid 1970s, the programming language “C” has become extremely popular because it enforces structure, takes relatively little memory and is closer to **assembly language** than most other high-level languages. Because of its “bare-bones” structure, it does require external support, in terms of libraries, to perform the involved mechanics of such things as

Controls Rookie Book

manipulating strings, and I/O (input/output). Although a large number of our older programs are still coded in **FORTRAN**, all new programs are now written in C. Of course, not wanting to get mired in complacency, we're moving on to the next language-of-choice — see **Java**.

- ❑ **C++:** A high-level programming language developed by Bjarne Stroustrup at Bell Labs. C++ adds **object-oriented** features to its predecessor, **C**.
- ❑ **Cache:** Literally, a place of storage. Within computer systems, information is cached to allow for speed differences between components. On our console system, whole programs are cached so that those most often used are the ones most readily available.
- ❑ **CAMAC** (Computer Automated Monitor And Control): There is an internationally accepted definition of hardware and interface protocols for this system and hardware is commercially available. At this time Booster, Tevatron, Switchyard, Debuncher/Accumulator, External Beamlines, and Console systems utilize the true standard CAMAC hardware. For a more in-depth look into our practical use of CAMAC, see the chapter on *Links*.
- ❑ **CAMAC Crate:** A commercially available 25-slot crate used to house CAMAC cards and to interface to the accelerator computer system. The crate has a back plane with connections to all cards in the crate. It has ± 6 -volt and ± 24 -volt DC crate voltages for use by the cards. At Fermilab, the crate controller uses the last four slots. All cards are electrically connected to the crate controller by two sets of buses: each card has its own address line, but they all share the same 24 parallel data lines. See **Crate** for illustrations of typical CAMAC crates.

Controls Rookie Book


- ❑ **Central Computers:** A term which is used to refer to the hardware and software which provide centralized services such as status (alarm) reporting, the sharing of files and shared data acquisition. These central computers are physically located in the computer room and are really the heart of the controls system.
- ❑ **Central Console:** The computer room console that is used to access the central machines and their associated software (once referred to as the “brain terminal” — don’t ask).
- ❑ **Central Program Loader (CPLD):** A **Central Service** program that **stages** (downloads) applications programs onto the console disk from the central system. When a program is activated by a console computer, CPLD works in conjunction with the Program Loader (**PLD**) to assure that the console has the newest version of the program. It also maintains a **cache** of the most commonly used programs at each console. (*See also cache.*)
- ❑ **Central Processing Unit (CPU):** 1. Literally, the “brain” or main processor chip of a computer system. It processes the sequences of internally recognized binary instructions (code) into logical commands.
2. The box containing the processor. (*See also Microprocessor.*)
- ❑ **Central Service:** Typically this refers to any one of a number of processes which interface to, or are themselves, a powerful software tool residing on one or more of the **central computers**. These services are centrally located because they can play host to all of the distributed systems. For a list and description (abstract) of central service processes, see Heartbeat Users page (D130). Our Controls Department has a *Central Services* group whose job it is to maintain and improve these processes.

Controls Rookie Book

- ❑ **Channel 13:** A lab-wide closed circuit TV channel used for displaying accelerator status information. The data displayed may consist of machine intensities, experimental area intensity requests, measured experimental area intensities, Main Injector and Tevatron ramp waveforms, and for colliding beams the batch intensities, store number, etc. The human interface to this information is the Channel 13 web page,

<http://www-bd.fnal.gov/notify/messages/>

where messages and the format of the channel known as *Notify Channel* can be updated. (See also *HERMES*.)

- ❑ **Chianti:** See **Wine Cluster**.
- ❑ **Circular Buffer:** A temporary area of memory of a finite length. When data reaches the end of the buffer, it wraps around and begins filling again from the beginning — a technique used for **snapshots**.
- ❑ **Circular File System:** A fixed set of **save files** that are used to save current conditions in a circular nature. That is, rather than asking  the user, the software will automatically use the next available file in sequential order, starting over from the top (overwriting the first file) when the last in the fixed set has been used.
- ❑ **CLIB (Console Library):** A collection of commonly used or convenient software modules that have been made available to all applications programmers.
- ❑ **CLIENT:** See *Client/Server Architecture*.
- ❑ **Client/Server Architecture:** A network architecture in which each computer or process on the network is either a client or a server. *Servers* are powerful computers or processes dedicated to managing disk drives

Controls Rookie Book

(file servers), printers (print servers), or network traffic (network servers). *Clients* are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

- ❑ **Clock:** A high frequency (MHz) signal that is used throughout the accelerator to synchronize all of its individual systems. (*See also: Beam Sync Clock, TCLK.*)
- ❑ **Clock (Beam Sync):** See *Beam Sync Clock*.
- ❑ **Clock Decoder:** Any card, such as a **177** or **279**, which recognizes a specific accelerator-clock bit pattern (clock event) and acts on it. Note: most of the decoder modules in the accelerator are based on a custom-made Fermilab chip built specifically for decoding TCLK. The 177 modules decode TCLK only; the 279 modules can decode the beam sync clock as well. (*See also Fermilab Clock Decoder.*)
- ❑ **Clock Encoder:** A card, such as a **175 module**, which stamps a bit pattern (clock event) onto a clock signal for recognition elsewhere.
- ❑ **Clock Event:** An 8-bit pattern superimposed on the clock frequency at a certain time in the accelerator cycle that is broadcast throughout the accelerator for decoding and action by specific systems and/or devices. The 8 bits limits the number of possible unique events to 256 — probably a good thing. Picture a given clock event as a marker on the accelerator timeline. (*See also Timeline.*)
- ❑ **Clockscope:** A console applications program that interactively displays (in a human-visible way) the clock events actually being sent out on the link. It currently resides on the Clockscope page, D33.

Controls Rookie Book

- ❑ **Comfort Display:** In the terms of *there's comfort in knowledge*, this refers to any one of a number of *dedicated displays* in the MCR which gives the operating crew an overall perspective of the monitored-system's status. These displays are usually located in the prime real estate at the very top of the MCR console racks.
- ❑ **Common Request List:** When an applications program displays data from accelerator devices it has to generate a list of what information is wanted. This list is assembled on the console computer and consists of DI/PI (device index/property index) pairs. The list is put together in the console computer by the console **DPM** task from information in the **database**. It is then organized and sent to the front end for further processing before requests for data are made to devices in the field.
- ❑ **Configuration Control:** An *unofficial* explanation is that this is a term used to describe the procedure of putting an area of the accelerator into a known pre-access configuration. Although it involves turning off power supplies and locking off breakers, it is NOT a replacement for **LOTO** — and this is as far as my conscience will allow me to involve this topic in the Controls Rookie Book.
- ❑ **Console:** Our human interface to accelerator control. This term can refer to a single-screen system as they appear in many offices or, as in the MCR, it can refer to one entire three-rack multi-screen setup and its associated scopes, analyzers and various other regalia.

Why are the consoles laid out as they are? As everything, the answer is buried in history. Physical monitors once replaced a subset of what you now see as independent windows. The unit closest to the keyboard displayed your current application page, the monitor above and to the one to the right were graphic-capable monitors, and the remaining

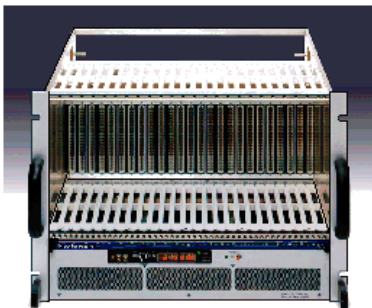
Controls Rookie Book

monitor was reserved for alarms display. The remaining rack space (mostly to the left) was intended to be reserved for scopes, counters and other monitoring equipment. It's interesting that the uniqueness of this latter equipment now defines, say, a "Linac" console from a "Booster" console; in an earlier life it was defined by the accelerator-independent computer that drove the console.

- ❑ **[Frig] Consolidator:** Communications with our ring-wide refrigeration systems is a sensitive operation. The nature of the beast is that when something goes awry (a quench for example) data requests start piling up. Too, the MCR has become accustomed to a rather (read that *very*) high-density refrigerator status display, which is updating constantly. The job of the consolidator is to gather all this data and present it to those in need as a single request. Once a separate front-end, it is now a Java application FRCONS running on DUE11.
- ❑ **Console Hardware:** See *Console*.
- ❑ **Controls Hardware Release (HDWREL):** A type of document once heavily published by the Beams Division Controls Group and distributed within the Division (including the Operations Department) which detailed the operation of new control cards and other hardware built by the Controls Department. They seem to have become an endangered species.
- ❑ **Controls Software Release (SFTREL):** A document once published by the Controls Department that included detailed information on new applications programs. Although they still exist, these have since been replaced by the more popular on-line help system.
- ❑ **Controls Software Update (SFTUPD):** A document that describes a change to a **Controls Software Release**.

Controls Rookie Book

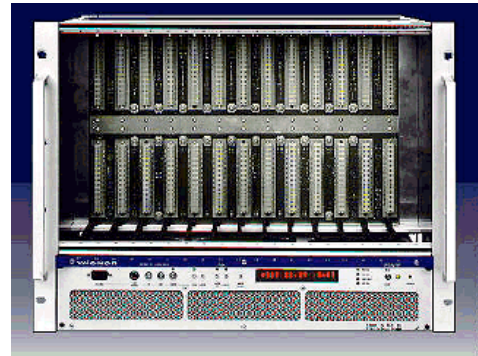
- ❑ **CoPaCa:** An acronym referring to the *Controls Parts Cabinet* located in the MCR. The purpose of this cabinet is to provide Operators with a wide variety of ready spare CAMAC cards, and also a common return point for basically *any* Controls Dept. hardware — broken or not. A form (located in the door) is to be filled out which describes the disposition of a returned part, and then, on a regular basis (usually daily) this cabinet is checked and dealt with by Controls hardware personnel.
- ❑ **COW (Console On Wheels):** A popular terminology referring to a temporary console. These 3-rack eye-sores (literally, on wheels) have been used in the past when the need has arisen to temporarily expand Main Control Room capabilities.
- ❑ **Crash Button:** In the event of an emergency, these devices are used in a variety of places around the accelerator complex to provide a means of readily removing a potential threat to the safety of personnel or equipment. **LOTO** must still be performed before working in the affected area.
- ❑ **Crate:** A physical frame with a **backplane** of expansion slots which allows the modular integration of dedicated-purpose circuit boards (cards or modules) to be connected to a common **bus**. Each type of crate has a different protocol for accessing and distributing information that must be recognized by the respective front end.



CAMAC Crate

Controls Rookie Book

There are various types of crates throughout the accelerator complex, and of course none of their cards are interchangeable. A few examples are depicted here.



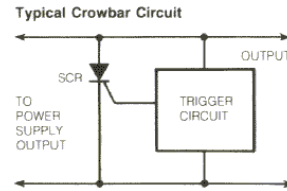
VXI Crate

Ignore the power supplies in the accompanying illustrations as they will vary depending on the manufacturer or, as you may find in a few of the installations around the site, are not integrated into the crate at all.

- ❑ **Crate Controller:** The card(s) in a given crate which decodes address information arriving on the link and accepts/distributes messages to/from the appropriate cards in that crate. These cards also arbitrate the data and messages to and from each card in the crate, as they all share the same data lines.
- ❑ **Crates and Slots Page:** A diagnostics applications program that allows the user to examine the CAMAC links and the types of cards residing in a given crate as well as its status. The applications program is available for each of the serial front ends (from any index select Search Title under the ♦Cmnds♦ menu and enter `Crates` as the search parameter; you'll find a slew of them.)

Controls Rookie Book

- ❑ **Crowbar Circuit:** An overvoltage protection circuit that rapidly places a low resistance shunt across the power supply output terminals if a predetermined voltage is exceeded.



- ❑ **Cursor:** A visible marker on a screen that indicates where on the screen a user may be interacting (or about to interact) with the program.
- ❑ **Cycle Time:** The respective timing repetition rate of a given accelerator.
(See also *Supercycle*)

- ❑ **D/A:** (1) In hardware this refers to a Digital-to-Analog Converter. It's a device that accepts a binary-coded number from the control system and converts it to a corresponding output voltage. Such a device is the complement of an **A/D converter**. In many instances the output is used to program the output of a power supply. Most D/A's in the accelerator control system output voltages are in the range of -10 to +10 volts.

(2) In software D/A refers to the *setting* value for a given controllable device.

- ❑ **DABELL (Database Batch Editing Language):** A **batch** program that implements device changes in the system **database**. For an on-line user's guide, see:

www-bd.fnal.gov/controls/database/dabbell/dabbell.html

- ❑ **Daisy Chain:** Describes a scheme wherein a number of devices are connected in series (this is supposed to conjure up visions of a daisy-petal-like configuration). The idea being that if the loop is broken, so goes the circuit, like removing one Christmas tree light bulb from a string. It's a technique often used in **interlock** schemes.

THE CONTROLSOPEDIA

- D -

Controls Rookie Book

- ❑ **Database:** A list of properties for each device on the ACNET system. These properties include mnemonics, alarm limits, descriptor texts, and scaling information — just about everything you ever wanted to know about a given device. Currently this list is maintained in a commercial Sybase database program. (*See also: DBM, Wine Cluster.*)
- ❑ **Datalogger:** A real-time data acquisition system that is used by Operators and other fine folks to look back on the trend of a specific database parameter. Of course one must have the foresight to initiate logging of the item of interest, so the desired data has a chance of being there when the request is made sometime later. (See also Lumberjack.)
- ❑ **Daughterboard:** Any board that plugs into a **backplane** or motherboard. And, don't forget, daughters can be mothers too.
- ❑ **Debug:** The process programmers go through to remove coding errors (*bugs*) and other unpopular features from their code.
- ❑ **DEC:** The not-so top secret code name for *Digital Equipment Corporation*, manufacturers of the PDP and VAX series computers, and perpetrators of the UNIBUS. Now owned by Compaq.
- ❑ **Decoder:** Any device that responds to the detected specific peculiarities of an input signal. See **Clock Decoder**, for example.
- ❑ **Dedicated Display:** See *Comfort Display*.
- ❑ **Device Index (DI):** This is a unique number assigned by DABBEL to each device entered into the database. This number becomes its software digital reference, sort of like your social security number is to the government.

Controls Rookie Book

Did you know that, on a parameter page, if you interrupt on a blank line below a device, its sibling will be displayed or (if none is defined) the next sequential DI? This is handy if, say, a ring full of devices was entered into the database in some logical order.

❑ **Device Name:** What's in a name? The answer is: As much as we can fit in 8 characters. All kinds of clever schemes have been incorporated over the years, but the only part that has remained relatively conventional are the first two characters — a letter and a colon. The letter is an accelerator system designator that indicates the system-association of the device, not necessarily the crate or node to which the device is connected. Below is a list of some of the designators and the system they refer to. I've also included how many devices are defined for each as of September 2004.

A:	Accumulator	4945
B:	Booster	7584
C:	Collider	4255
D:	Debuncher (and Pbar transfer lines)	5562
E:	Experiment-related	4829
G:	General	5746
L:	Linac	8039
I:	Main Injector	20666
M:	P2 & P3 lines and previous Main Ring stuff	2660
R:	Recycler	6857
S:	Switchyard	5019
T:	Tevatron	38331
X:	Accumulator Deceleration – Ramping Devices	498
Z:	Test devices (cannot be saved)	7306

I should point out that there is movement underfoot to increase this rather limited number of characters in a device name; however it is not a trivial pursuit. Most (if not all) of our console and utility programs expect no more than 8-characters. Too, having names that are too long would be painful to enter. The answer will likely come in the form of having both, a long name and a short name. Time will tell.

❑ **Digital Status:** See *Basic Status*.

Controls Rookie Book

❑ **Disk:** See *Hard Disk*.

❑ **Diskette:** The most common are known as “floppy disks”; these act like hard **drives** except they’re portable... unfortunately, they also are able to hold a pittance of the same information. More popular (and more expensive) are *optical diskettes* or even a combination of the two which are known as *floptical diskettes*.

❑ **[FIX] DMACS:** A proprietary software programming package sold by Intellution®. It allows the programmer to monitor and adjust a **PLC**-controlled system. In our environment it is used in a few closed-loop cryogenic systems, which hopefully Operators will not have to deal with on a first-hand basis. (*See also LabView®.*)

❑ **Download:** The forced transfer of data from another computer to your computer. Often viewed in a more precise context of a master vs. slave, wherein a transfer from master-to-slave (big-to-small) is a download and slave-to-master transfer (small-to-big) is an *upload*.

❑ **DPM (Data Pool Manager):** A task on the console computers which accesses the ACNET device **database** in order to organize request lists for setting and reading data.

❑ **Dumb [Module/Terminal/...]:** This is not referring to a non-responsive not-too-bright entity somewhere, but rather an entity that has no built-in processor. Whether it’s a module, a terminal or your neighbor’s dog, if there’s no built-in ‘intelligence’ which allows it to make impressive decisions based on current conditions, it’s a dumb module, dumb terminal, dumb dog, etc.

Controls Rookie Book

- ❑ **Electronic Logbook:** A ledger or journal maintained in a software medium.
- ❑ **Electronic Mail:** The mechanism that allows individuals to send and receive messages over a communications network. At Fermilab, Operators are provided with an account on the central FNAL mail server, which gives them an email address of <username>@fnal.gov. The mail server, then, forwards incoming messages to their destination-of-choice. The most common destination is one of the **IMAP** servers.
- ❑ **Encoder:** Any device that generates a signal unique to a given condition. The intent being that the signal will be detected, interpreted and acted upon by any **decoder** receiving the signal. See **Clock Encoder**, for example.
- ❑ **Energy Conservation [mode]:** Through the use of a one-of-a-kind 477 card (a modified 377), a clever technique is employed in Switchyard wherein the power supplies are disabled from ramping when there's no significant beam available to extract — kind of makes sense when you think about it. The beam intensity is monitored at Main Injector flattop and at Tevatron injection and if it's less than a set threshold (currently $\sim 5 \times 10^{11}$), the 477 is disabled. It is re-enabled on the "*Set On Event*" Event (*SOEE*) in time for the next cycle.
- ❑ **Epicure:** An ACNET-like system used in the fixed-target areas of the accelerator. An on-going effort is being waged to ACNETize that part of the Epicure that falls in the jurisdiction of the Beams Division. Things like device name length make it a rough go.

Controls Rookie Book

❑ **ERM (Event Request Modules):** These are the CAMAC **175** modules which encode events onto the Tevatron clock. They are located in the MAC room.

❑ **Error Codes:** These are a locally standardized list of codes which are supposed to tell you why you didn't succeed at what it is you just attempted to do. A typical code tells you where the message is coming from and a numeric representation of the error, 2 -1, for example. At any ACNET console, hit the F14 key to open the Utility window and enter 2 -1 in the Error Help field. You'll find that the "2" refers to FileSharing and that the -1 means "No working buffer."

Most previously anticipated errors that might be generated by, say, the incorrect use of a program option, are stated in plain-text English in the "messages" box provided at the bottom of most console pages. In the *unanticipated* situation like the one above, (a program bug) or where space is at a premium, you may get the ugly numeric representation.

❑ **Ethernet:** An access method standard (IEEE 802.3) developed in 1976 by Xerox et al. (DEC and Intel) for connecting computers into a local area network (**LAN**). The most common form of Ethernet is called 10BaseT, which denotes a peak transmission speed of 10-Mbps using copper twisted-pair cable. Faster (100-Mbs and 1-Gbs) Ethernet networks are already being tested. See the chapter on *Networking* for more information on this and other access methods.

❑ **Event:** See *Clock Event*.

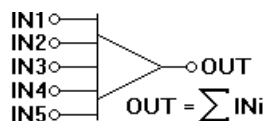
❑ **External Beamline Internal Clock:** Used when the main accelerator is off or is in some other non-compatible mode. This timing system, which is internal to the experimental beamlines area, is occasionally put into use

Controls Rookie Book

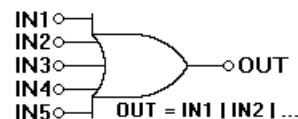
to simulate standard operating conditions. It may be used for power supply testing or some other off-line experimental use. It is controlled by the MCR.

- ❑ **Fail-safe:** A concept designed into safety-related systems wherein a signal or a constant mechanical force is required to achieve a permit. A power or component failure, for example, will cause the permit to be lost because the signal is lost or the mechanical force is removed, thus *failing* in a *safe* condition.

- ❑ **Fan-in:** A board or other device that gathers multiple signals from a variety of devices and combines (multiplexes) them for processing (as



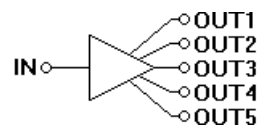
Linear = adds the amplitudes of the input signals together, output equals the algebraic sum of the inputs



Logic/Digital = sets the output to logic high whenever one of the inputs is logic high (logical OR function)

with a **MADC**).

- ❑ **Fan-out:** A board or other device that receives a single signal input and provides multiple duplicate outputs.
- ❑ **Far Front End:** A term used to refer to any front end remote from the rest of the herd in the computer room.
- ❑ **FASTBUS:** A sophisticated data acquisition system standard developed by the U.S. NIM Committee et al., to keep features of older important standards while extending the capabilities of data acquisition systems. FASTBUS provides for a more densely packed system, dramatically reducing the



THE CONTROLSOPEDIA

- F -

Controls Rookie Book

per-input cost. It is intended to be an extension of a computer **backplane**. See **Crate** for an illustration of a FASTBUS crate.

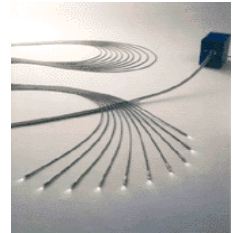
❑ **Fast Time Plot (FTP):** The standard plot generated on a console display from most any console page. (*See also Fast Time Plot Program.*)

❑ **Fast Time Plot Program:** This comprehensive program has evolved with ACNET and is designed to handle any number of user options from the simple to the complex. For those who are curious about all its most up-to-date features, feel free to visit the Web URL: See the chapter on *Networking* for more information on this and other access methods.

`www-bd.fnal.gov/controls/micro_p/ftp.txt`

❑ **Fiber Data Distributed Interface (FDDI):** This is a high-speed networking access method that incorporates a double token-passing ring system. (See also the chapter on *Networking*.)

❑ **Fiber-optic [link]:** Uses light shining through strands of glass as a communications medium (not unlike those fancy lamps one can spend gobs of money on). It's not as fast as copper, but has no significant signal power loss to deal with, which means it's the ultimate in efficiency as a transmission medium (though expensive to install). Probably the oldest fiber-optic link in our environment exists between the PreAcc ground station and the 750Kv dome. This nonmetallic *light-link* provides a perfect medium to keep the difference of potential between these two areas safely isolated.



❑ **FileMaster:** A turnkey (no-password) account that assists programmers in managing their FileSharing files. (*See also FileSharing*)

Controls Rookie Book

- ❑ **FileSharing:** When dealing with the vast number of programs required to operate the various accelerators, it becomes essential to provide a storage and retrieval area for internal data — save files, for example. This system not only allows files to be shared with other programs, but also provides a one-location repository. There are over 4000 files in this repository. Are they all being used? No, but it's not trivial to safely determine which ones can/should be removed — so we don't.

It should be mentioned that with the advent of the new Sybase **database** system, we're turning to *it* as our shared-data repository and no new files are being added to FileSharing. Elimination of the system, however, will be a long and slow evolution. (*See also: FileMaster*)

- ❑ **Finite State Machine (FSM):** Basically, an FSM is a program that changes its logic based on current conditions. An example of one such system is our cryogenic refrigerators. As the magnets cool from room temperature to -400°F (or likewise during quench recovery), the system goes through various phases, each requiring significant changes in the controlled settings. An FSM recognizes the passing from one phase to another and alters these settings accordingly. You can visit our interface to this system on the *Finite St Machines* page, F13 or the *F_S_M Viewer* page, F24.

- ❑ **Firewall:** A hardware or software system designed to prevent unauthorized access to or from a private network.

- ❑ **FIRUS (FIRe and Utility System):** A site-wide fire, trouble and utility monitoring system. Just about anything that could cause significant loss to the Fermilab infrastructure is monitored by FIRUS. Power failures, fire detectors, sprinklers, secured areas, panic alarms, temperatures, pressures, levels, and a host of other things you wouldn't believe.

Controls Rookie Book

The FIRUS system is an entity onto itself, having no dependence on other systems to do its job — it is not part of the ACNET system. Even its power is backed up by a gas-turbine generator in the basement of Wilson Hall. Most all conditions that directly effect machine operation are monitored separately (from FIRUS) within the ACNET system and reported back via the **ALARMS** system. In fact, the only significant connection between Controls and FIRUS is that the system is maintained by our Controls Department. The FIRUS electronics is housed in its own room (XGC119) below the MCR.

By default, the FIRUS monitor in the MCR is instructed to report only those problems relevant to the Beams Division. The Wilson Hall security office monitors all messages. In an emergency the MCR console could be told to monitor the whole site. (*See also FIRUS Mini.*)

- ❑ **FIRUS Mini:** This unit is a summation point for all types of FIRUS alarm monitoring in a given area. For example, there is a FIRUS mini in every third Tevatron Ring service building, each of which monitors alarms for three buildings. (*See also FIRUS.*)
- ❑ **FORTTRAN (FORMula TRANslator):** Developed by IBM way back in the 1950's, it's the oldest of the high level programming language. It has kept pace over the years by evolving into FORTRAN IV and FORTRAN 77. Once the language-of-choice in our environment, it has lost that distinction to **C** and **Java**.
- ❑ **Front end:** An intelligent interface between a standardized communications system (like ACNET) and the myriad of hardware components making up our accelerator. The front end recognizes the programmer's standardized dialect on one hand and a cacophony of other standardized dialects on the other. There may well be a sub level of front

Controls Rookie Book

ends within a given system to facilitate comprehensive local control. Most often, however, Operators think in terms of one of the computers in the CAMAC crates located in the west-most racks of the computer room, which handles the communication of data requests for a given accelerator system (Booster, Pbar, TeV, etc.). These front ends are interfaced to the various hardware crates on one hand, and to the consoles and central computers on the other.

- ❑ **FSHARE:** See *FileSharing*.
- ❑ **FTD:** When you're not in the mood to send flowers, this is a mnemonic for *Frequency Time Descriptor*. It is a term used in numerous data gathering schemes to indicate to the front end how often it should read information from a module. The frequency could be in seconds or at a specified **clock event**.
- ❑ **FTP (Fast Time Plot):** See *Fast Time Plot*.
- ❑ **FTP (File Transfer Protocol):** A communications technique used to transfer files from one computer to another over the Internet.
- ❑ **Full Duplex:** Simultaneous two-way independent transmission on the same medium. (*See also Half Duplex.*)
- ❑ **Fuzzy Logic:** A type of logic which recognizes more than just true and false, going more into the realm of various levels of truth, as might be incorporated in a **Finite State Machine**. A simple example to visualize is a spelling checker — a utility near and dear to most everyone's heart — which goes beyond simply telling you that your word is spelled wrong; it goes a step further and lists options which are close.

Controls Rookie Book

THE CONTROLSOPEDIA

- G -

- ❑ **GAS** (GHASP Advanced Software): A version of the **GHASP** protocol that has been added to list-processing and single-shot support. GAS performs much of the software that would otherwise be done by the front end. Usually information is exchanged between the front end and an **080 module**, but in the case of vacuum control the front end talks to a **170 module**.
- ❑ **GHASP** (General Host And Subsystem Protocol): The protocol used to communicate between **CAMAC** and **GAS**-speaking modules.
- ❑ **[Power] Glitch**: Any short-duration interruption of electrical line voltage is referred to as a power glitch. A long-duration interruption is referred to as a *power outage*.
- ❑ **GPIB (General Purpose Interface Bus)**: An interface (usually slow) between a hardware device and ACNET. The interface is available commercially from Hewlett-Packard. Examples include spectrum analyzers, scopes, and the DCCT current monitors (as used in the antiproton source).

THE CONTROLSOPEDIA

- H -

- ❑ **Half-duplex**: This is a data transmission scheme that is characterized by transmission in both directions in an alternate one-way-at-a-time fashion. (*See also Duplex.*)
- ❑ **Half-high NIM**: This is a nice compact (5.25”) *Nuclear Instrument Module* crate that is used in a few locations around the accelerator where size is a concern or simply more efficient. The most obvious of these are the consoles in the MCR — the crate that contains beam switch, abort button, link repeaters, etc.

Controls Rookie Book

- ❑ **Hard Disk:** A **nonvolatile memory** device used to store data magnetically.
- ❑ **Hardwire:** This term is used in the context of referring to a physical electrical condition. It is always implemented where the protection of personnel or equipment depends on the electrical deactivation of a dangerous potential. It is the basis for our beam abort and tunnel access systems, for example.
- ❑ **Heartbeat:** In our software world this is a term used to describe a yes-I'm-still-alive-quit-worryin' periodic communication. One type of heartbeat might represent itself as a visual change in an otherwise relatively inactive console display (i.e., a vacuum display). Another type is implemented by our behind-the-scenes mission-critical software — like our central services — which often incorporate an active handshake type of heartbeat so as to permit timely notification should something go awry (and the heartbeat stops). These latter tasks are registered and viewed on the *Heartbeat Users* page, D130.
- ❑ **Help Facility:** An on-line tool that can be accessed via a Web browser. This facility has literally replaced the endless (and always out-of-date) paper trail of software documentation. Operators are encouraged to not only reference information on the current application page (via the HELP key), but also add their own helpful hints, comments and updates.
- ❑ **Hemorrhoid Box:** Jargon referring to the **Beam Switch Sum Box**.
- ❑ **HERMES:** The central process that is the software interface, which provides data and information to Channel 13 and Notify Channel.
- ❑ **Hypertext:** A system in which one is able to interactively link objects (text, pictures, programs, etc.) to each other via a *hyperlink* or *button*.

Controls Rookie Book

THE CONTROLSOPEDIA

- I -

- ❑ **IMAP (Internet Message Access Protocol):** Similar to **POP**, IMAP is a protocol for retrieving e-mail messages but with better remote handling features. Typically, with IMAP, you deal with your messages right on the server, whereas with POP your mail is downloaded to your machine and manipulated locally — all fine and dandy if you always access your mail with the same machine. Both IMAP and POP are available at Fermi.
- ❑ **Index Page Editor (IPE):** A utility, available through **MECCA**, which allows for the care and maintenance of the various console index pages.
- ❑ **Indicating Fuse Holder:** A clever technique wherein an indicator is an integral part of a fuse holder and lights when fuse blows. Another technique is a button that pops out when the fuse blows. Don't trust 'em.
- ❑ **Instance:** A term used to describe a console's topological relationship. In the scheme of things, one CPU can service two or more consoles. The console directly connected to the CPU is the master and is referred to as *Instance zero*; the slaves will be recognized as *Instance 1*, *Instance 2*, etc. You can determine which instance any console is by looking at the *Console Peek* page, D15. The instance 0s are listed first as numbers 0–99, instance 1s and 2s are the 100s and 200s, respectively. For example: *Instance 0* of console 6 is located on the east side of the MCR, *Instance 1* is 106 and drives the **comfort display** above console 6, and *Instance 2* is 206 and drives a desktop console out in the boonies somewhere.
- ❑ **Interlock:** An electronic system that is designed to protect equipment or personnel. For example, a power supply's side panel might be interlocked with a **microswitch** to trigger a response if the panel is removed. The response may be the removal of power from the supply, the sounding of an alarm, the triggering of a remote monitor, etc.

Controls Rookie Book

❑ **Internet:** A global *web* of over 1 million computers and 50 million users in over 100 countries. It's a vast collection of interconnected files and programs spanning the globe and retrievable via a client-*server* system utilizing **hypertext**. Browser programs access the Web (e.g., Netscape or Internet Explorer). (See also Intranet.)

❑ **Intranet:** A TCP/IP-protocol network (**internet**) designed to operate within the confines of an organization (like Fermilab). Often the information is only made available to the organization's members.

❑ **Internet Rack Monitor (IRM):** An intelligent data acquisition system used in the Booster and Main Injector HLRF systems, the Linac and elsewhere. For



A Linac IRM

For further detail see your Linac Rookie Book and/or visit the address:

www-linac.fnal.gov/irm/WriteupPage.html

❑ **Interrupt:** In a console context, an interrupt is a sort of software trigger. The most common four are: page initialization, a click of the console **interrupt button**, at a periodic rate (such as when updating page information), and page termination.

❑ **Interrupt Button:** A button next to the keyboard (usually part of the mouse or trackball) at each ACNET console that allows one to manually trigger the current page options.

❑ **IP (Internet Protocol):** This is described more fully in the chapter on *Networking*, but the short of it is that it's a technique for passing messages around a network. (See also *TCP/IP*.)

Controls Rookie Book

❑ **ISDN (Integrated Services Digital Network):** This is an international communications standard for sending voice, video, and data over digital telephone lines. It supports data transfer rates of 64-Kbps (64,000 bits per second). Most ISDN lines offered by telephone companies give you two lines at once, called B channels. You can use one line for voice and the other for data, or you can use both lines for data to give you data rates of 128-Kbps — four or five times the data rate provided by today's fastest modems. ISDN is becoming a popular means for maintenance and staff personnel to monitor their respective Lab systems from their homes.

THE CONTROLSOPEDIA

- J -

❑ **Java:** An **object-oriented** programming language much like a simplified version of **C++**. Small (transportable) Java applications are called Java *applets*. Java *beans* are objects that conform to Java standards (established by Sun Microsystems) and therefore can be used by any application that recognizes the Java standard.

An on-going project at the time of this writing is the replacement of all console and central applications and services with Java code.

THE CONTROLSOPEDIA

- K -

❑ **Key Logger:** A stand-alone computer and program used to control the distribution and registration of accelerator keys. It is not part of ACNET.

❑ **Klixon:** A thermally activated switch which is used to trigger a protective sequence in the event a piece of equipment gets too warm. Since these little fellas are relatively inexpensive and have been known to fail, it's not unusual for more than one to be at a given location.

❑ **Knob:** Located in each console's **half-high NIM**, these clever devices allow one to *crank* the selected **D/A** up or down. An absolute necessity

Controls Rookie Book

when the final value of a device is being “tuned” as opposed to being set to a specific value.

❑ **LabView®**: A single-breath way of saying *Laboratory Virtual Instrument Engineering Workbench*, it’s a proprietary graphical software programming language. It consists of a set of software building blocks that represent I/O ports, basic numerical functions, and other simple wiring tools for connecting all of the pieces. The user interface is built with virtual buttons, knobs, switches, indicators, and graphs. Developers (like our Instrumentation Department) like it because it allows them a relatively easy mechanism to add a meaningful interface to a project early in its development — long before it reaches the stage of tying into a global networking scheme (like ACNET).

❑ **LAM**: This acronym stands for *Look At Me*. It is a flag used by many accelerator control cards to indicate to the front end that it has information ready to be passed.

❑ **LAN (Local Area Network)**: A fairly short-distance computer network. It’s not unusual for a number of LAN’s to be connected together to form a *Wide Area Network* (WAN).

❑ **LED (Light Emitting Diode)**: A solid state device used extensively for digital information display on control card front panels, exits signs, your bread maker and elsewhere.



❑ **Lex (Lexidata)**: This reference is still found within a number of applications pages and, in context, refers to any one of the console graphics windows (those whose titles begin with Gx...). For the curious: these software windows were once separate (hardware) *Lexidata®* graphics monitors — there were only two per console.

THE CONTROLSOPEDIA

- L -

Controls Rookie Book

❑ **Light Link:** See **fiber-optic**.

❑ **Link:** At Fermilab, links are the systems of cables and **repeaters** that carry information to and from various points in the accelerator. This is such a trendy topic, we've devoted an entire chapter to it.

❑ **Link Driver:** A device that organizes and transmits digital information on a link. In the context of a link driver for a console or front-end computer, this device performs a parallel-to-serial conversion on data to be transmitted or received over a link.

❑ **LOTO (Lockout/Tagout):** An *unofficial* definition is: a safety procedure involving the removing of energy and the active placing of locks and tags by the properly-trained person who will be performing work in the affected area. [For an *official* definition, see your local Safety Dept.]

❑ **Lumberjack:** A program residing on multiple VAX Stations which allows for the automatic and regular acquisition of data from an assigned set of devices, each with its own acquisition FTD. Why the name "Lumberjack"? It's a datalogger system — logger ⇔ lumberjack, get it? Hey, don't blame me! For more specifics, see:

www-bd.fnal.gov/acnet/datalogging.html

❑ **Machine, The:** The jargon most often used to refer to our entire conglomeration of accelerators and all of its infinite parts as a single entity. I'd have chosen something more descriptive, like Kicksy-Whimsy.

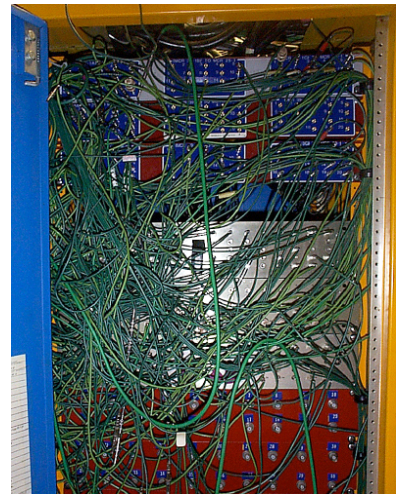
Controls Rookie Book

- ❑ **Machine Language:** The lowest-level programming language, but really the only one understood by a computer. Since it consists entirely of binary commands, programmers resort to high-level languages like Java, C or **FORTRAN**, to write their code, which is then compiled (translated) into machine language.
- ❑ **Machine Reset:** This can refer to the end (or, some perceive it as the beginning) of the repetitive cycle of one accelerator or the whole ball-o-wax — see *Supercycle Reset*.
- ❑ **MADC Multiplexed Analog-to-Digital Converter):** At Fermilab, these are local stations capable of accepting up to 64 analog inputs and converting each of them into a digital signal expressed in units of voltage. They consist of a single **analog-to-digital converter** with an analog input multiplexer to select one of the many possible input signals for conversion. Since only one channel may be digitized at a time, many commands are needed to digitize all possible input channels. This gives rise for the need of an MADC controller to coordinate its activities and issue the necessary commands.
- ❑ **MADC Volts:** The voltage output of an MADC, ranging from -10.24 V to + 10.24 V. The least significant bit is 2.5-mv. The voltage represents a twelve-digit bit pattern generated from an analog signal. Conversion factors to engineering units are kept in the **database**.
- ❑ **MakeReport:** A **turnkey** VAX account that is used as an Operations general-purpose area. Among other things this is where the Early Bird reports are generated and where department-distributed email is archived. So, if you lost an email message that was sent to the department distribution list, odds are, you can find it here.

Controls Rookie Book

❑ **Main Frame:** If it was spelled as all one word, this would refer to a large (and expensive) central computer system — a reference we never use. In the context as it's spelled here, it refers to our MCR cabling system. This system is located in the **back racks** and is a centralized distribution point for all hardwired signals entering and leaving the MCR. All connections to each console's **miniframe** are hardwired to the back of a corresponding patch panel located at the top of the main frame. Below these are **fan-out** modules (for routing the same signal to multiple locations) and below these are hardwired all connections to the 'outside world' (again, to the back). The front of the main frame, then, is a patch panel wherein signals can be patched to one or more consoles in the MCR. Even signals generated at one console can quickly be fed to another.

Although it looks like the rat's nest it is, it actually takes very little time to trace, add or remove a signal-to-console connection as compared to climbing through cable trays and wire bundles. In fact, one of the worst things a well-meaning soul can do is to try and be neat by placing a nice tight cable-tie around a bundle of main frame cables, or even



worse, run them through some plastic channels — they've just defeated the whole purpose and made it impossible to trace any of those cables without cutting the tie or removing the channeling.

❑ **Manchester Coding:** This is a standardized coding scheme for digital information on a high frequency RF carrier. It is used by the Tevatron

Controls Rookie Book

clock, the Linac serial link, and other systems. For example, TCLK operates at a 10-Mhz rate, and byte-sized signals can be sent out at a 1-Mhz rate. That allows for 10 100-nsec frames. The clock is structured so that if a phase transition takes place in the middle of a frame, it is interpreted as a “1,” otherwise it is interpreted as a “0.” Being a “self-clocking” mechanism, time errors do not accumulate as clock events are superimposed on the pattern. There is also a start bit (0) at the beginning of each clock event and a parity bit at the end. The 10-Mhz pattern is transmitted around the accelerator on an RF carrier frequency of 50-Mhz, and is demodulated and retransmitted at each house. (See also the chapter on *Links*.)

- ❑ **MDAT (Machine Data):** This refers to a signal which transmits around the accelerator such things as the programmed values for the Tevatron current and dI/dT , the measured values for the Tevatron current and dI , dT , and the measured Main Injector current. MDAT (pronounced em-dat), in the Tevatron, is carried on one of the 19-conductor cables at a 720-Hz rate. This information is then decoded and used by several devices, such as the ramped correction elements.

See also: www-bd.fnal.gov/controls/camac_modules/c166.html

- ❑ **MECAR (Main Injector Excitation Control And Regulation):** The microprocessor system that sends phasing information to the Main Injector’s main power supplies.
- ❑ **MECCA (Management Environment for Controls Console Applications):** Our software code-capture and proofing system. Being the transient community that we are, where programmers come and go like Halloween candy, this software librarian ensures that mission-critical programs — the most obvious being the console pages — are held on to.

Controls Rookie Book

In fact, MECCA goes so far as to not only make sure the code compiles (syntactically correct), but also makes sure it conforms to a certain core set of requirements. Need to modify a program that hasn't been touched for 4 or 5 years? No problem. Extract the code from MECCA, make the necessary changes and resubmit it.

For a more complete nonprogrammer's overview of how our console software comes to be, see the chapter on *Console Applications*.

❑ **MIBS Clock:** The *Main Injector beam sync clock* operates at a near 7.5-Mhz rate, so that (based on MI RF cycles) there are exactly 84 clock ticks per turn (159 for **TVBS**). It is used wherever synchronization to the beam has to be especially accurate.

❑ **Microcomputer:** A small computer based on a microprocessor chip.

❑ **Microprocessor:** See *Central Processing Unit*.

❑ **Microswitch:** Heavily used around the accelerator; their long lever-arm makes them ideal for implementing in a door or surface panel **interlock** circuit. Almost as if that's what they were designed for.



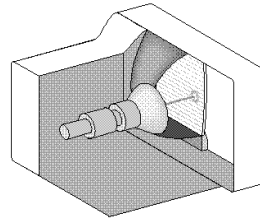
❑ **Miniframe:** A narrow patch panel located on the back side and at the top of each MCR console. Signals from the main frame arrive at this panel and can then be distributed, as needed, to appear on the **signal**



Controls Rookie Book

distribution panel at the front of the console.

- ❑ **Min/Max:** The limits of tolerance for a particular device. An alarm (and sometimes a beam inhibit or abort) is generated when the value of the device falls outside these limits.
- ❑ **Modem:** An acronym for *MO*dulator/*DE*Modulator, this is a device that translates computer signals into a format that can be communicated over telephone lines.
- ❑ **Monitor:** These hybrid television-like units are generally graphics-capable and, since they can directly process a composite video signal, they present a clean, crisp, display. Terms to brush up on if you want to bedazzle your neighbors are things like *gray-scale*, *resolution*, and *multi-sync* or *multi-scan*.



- ❑ **MOOC: (Minimal Object Oriented Communications):** A scheme that is intended to standardize front end coding. It makes it possible to reuse code which implements the functionality common to all front ends, like RETDAT/SETDAT, echoing settings to the central database, and handling the FTP and snapshot plot protocols. If you're interested in getting into the real nitty-gritty of MOOC, see:

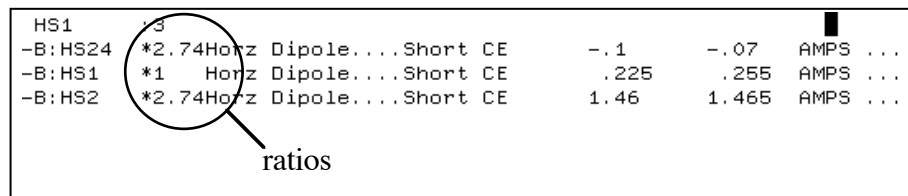
www-bd.fnal.gov/controls/micro_p/mooc_project/kickoff.html

- ❑ **Mountain Range Display:** An oscilloscope plot, usually generated in the MRRF building, which displays bunch structure vs. time. The value of this type of display is that it can be triggered at some frequency to show information (such as bunch structure) for a set number of triggers, with each trigger having a different vertical offset. If the frequency is chosen

Controls Rookie Book

to be the revolution frequency, one can watch the time evolution within a window, of the same bunch as it circulates in the accelerator.

- ❑ **Mult:** On a console parameter page, a user-specified link between two or more devices that facilitates the aggregate adjustment of the whole according to specified ratios (as depicted here in the mult 3).



HS1	*3						
-B:HS24	*2.74	Horz	Dipole...	Short	CE	-.1	-.07 AMPS ...
-B:HS1	*1	Horz	Dipole...	Short	CE	.225	.255 AMPS ...
-B:HS2	*2.74	Horz	Dipole...	Short	CE	1.46	1.465 AMPS ...

An actual Mult (from console page B8)

- ❑ **Multibus Crate:** A set of standard signal lines used to interconnect the modules of various microprocessors, such as the Z80 and Motorola 68000. Physically it is a back plane that the modules plug into. The Multibus crate is capable of supporting twenty address lines, sixteen bi-directional data lines, and eight parallel interrupt lines, as well as bus control signals, data control signals, and power distribution lines.
- ❑ **Multibus II:** In the early 1980s, a bunch of potential customers (along with Intel's leadership) came together and specified what was to become IEEE standard 1296 — Multibus II:
 - Eurocard board sizes and DIN connectors
 - self-test capability
 - software jumpers
 - fast 32-bit bus
 - functional partitioning (using LAN concepts)
 - reliability
 - inter-operability

Controls Rookie Book

- ❑ **Mux (Multiplexer):** A device that receives inputs from a number of sources and cleverly shuffles them all together into one signal. A user can elect to examine one of the multiplexed signals (like the channels on a TV) and presto, it'll get unshuffled (unmultiplexed).

- ❑ **NBS Clock:** Our National Bureau of Standards clock. WWV in Boulder Colorado broadcasts time-of-day synchronization signals. The MCR picks up these signals via two receivers, one via open air transmission, and the other via satellite. The signals are used extensively throughout the lab, even going so far as synchronizing time-of-day on our desktop computer systems.



Time-of-day accuracy is more important than one might at first think. Performance statistics, for example, must correlate downtime in the various parts of the accelerator with machine activities in other parts of the accelerator (that might even be — gasp — in other states). In point of fact, at one point in time it was necessary to change operating modes because of electric company imposed power demand limits that went into effect at 9am. Standing around and comparing watches didn't cut it.

- ❑ **NETPROCESS:** A process on each central computer that runs the communications software we know as ACNET.

- ❑ **Network:** Two or more computers that have been given a communications link.

- ❑ **NIM (Nuclear Instrumentation Module):** Standard crate equipment (predating CAMAC), with a defined protocol, for some devices in the

THE CONTROLSOPEDIA

- N -

Controls Rookie Book

accelerator. “Half-high” crates in the MCR are only used for power and the modules do not communicate with each other. (*See also Half-high NIM.*)

❑ **Nineteen-conductor cable:** See *19-Conductor Cable*.

❑ **Node:** A processing point in a computing network.

❑ **Nominal:** In the context of values being presented on the typical parameter page, this is the last known ‘good value’ of a given device. When tuning, this value is preserved adjacent to the ‘new value.’ Clicking on it “restores the nominal” to its original *setting*.

❑ **Nonvolatile Memory:** Memory not normally susceptible to power failure. (*See also Hard Disk.*)

❑ **No Q:** A response generated when a GAS-speaking module, such as an 080- or 170-card, fails to respond (return Q) because the data’s not yet available (and perhaps never will be).

❑ **No X:** An error code generated when a CAMAC module fails to acknowledge its function codes. It can mean that the user is trying to address a nonexistent module or that an incorrect address is being used or the module is busted.

❑ **Object-oriented [programming]:** It’s inevitable that around the accelerator you’re going to hear various object-oriented acronyms, like **MOOC** — indeed, the OO words seem to come and go like cicadas. In an object-oriented scheme, programmers define both the data structure and also the types of operations (functions) that can be applied to the data structure. In this way, the data structure becomes an object that includes both data and functions. In addition, programmers can create

Controls Rookie Book

relationships between one object and another. For example, objects can inherit characteristics from other objects. It really has become quite the rave.

One of the principal advantages of object-oriented (over procedural) programming techniques is that they enable programmers to create modules that don't need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify. As an Operator, do you care? Not much.

- ❑ **Open Access Client (OAC) ["Oak"]:** These rather powerful applications act as a *virtual* front end for modeling purposes (an OAM) or as a general purpose *real* front end (an OAF).

Perhaps you have just created an ACNET page and wanted to see if all the status bits work, you could use the Model Redirection page, (D128) and activate a "mirror" which creates a pseudo page that permits status changes without effecting the real devices. Ah, but this OAM example is just one of many applications currently taking advantage of this mechanism. For a complete list see:

www-bd.fnal.gov/acnet/open_access_fe.html

For a more in-depth explanation of OACs see:

www-bd.fnal.gov/www/controls/user_libraries/ul_oac/intro_oac.html

- ❑ **Operator Account:** An account which allows Operators to perform certain rebooting tasks and other functions on the ACNET system from a Central Console located in the main computer room

Controls Rookie Book

THE CONTROLSOPEDIA

- P -

❑ **Packetter:** Among others, we seem to be hopelessly mired in two different networking topologies: **Ethernet** and **token-ring**. Not that that's necessarily bad, but there exists a packet-size incompatibility between the two. The packetter takes care of this little inconvenience electronically by breaking large transmissions into smaller chunks and consolidating the small transmissions into larger ones.

❑ **Parallel Data:** In terms of data transmission this term means that all bits of a character or byte are transmitted simultaneously. This requires a multiconductor cable or bus with each conductor carrying a single bit. A 16-bit word requires 16-conductor cable (plus ground) for transmission. *Ribbon cable* is often used for making parallel connections.



Controls Rookie Book

❑ **Parameter Page:** Any of a number of pages with a standard format listing devices on the ACNET system (see below). The device mnemonics, descriptive texts, D/A settings, and A/D readbacks and other information are displayed for each device. Parameter pages also support the standard plotting package (see *Fast Time Plot*), and a multitude of display, copy and print options. There are a few variants that are defined when a parameter page is first created which allow it to have up to 99 *numbered* subpages and/or numerous *named* subpages. The example below depicts that the user is on subpage 1 (-< 1>+ on line 4) of subpage HSXX:3 (the all caps item on line 5).

```
B8  HORIZONTAL 3-BUMPS          SET  D/A  A/D  Com-U ♦COPIES♦
-<FTP>+ *SA♦ X-A/D  X=TIME      Y=M:QFB6 ,Z ECARCI,Z ECFILV,
COMMAND ---- Eng-U  I= 0      I=-10   , 0      ,-20    , 0
-< 1>+ s_MR AUTO  F= 1.6      F= 10    , 2      , 5     , 1
debcher 400blms HSXX:3 bulk ps vlxx:3 400mev damllrf linac..
HS1      :3
-B:HS24  *2.74Horz Dipole....Short CE    -.1      -.07    AMPS ...
-B:HS1   *1   Horz Dipole....Short CE     .225     .255    AMPS ...
-B:HS2   *2.74Horz Dipole....Short CE     1.46     1.465  AMPS ...

HS2      :3
-B:HS1   *2.74Horz Dipole....Short CE     .225     .255    AMPS ...
-B:HS2   *1   Horz Dipole....Short CE     1.46     1.465  AMPS ...
-B:HS3   *2.74Horz Dipole....Short CE     1.51     1.48    AMPS ...

HS3      :3
-B:HS2   *2.74Horz Dipole....Short CE     1.46     1.465  AMPS ...
-B:HS3   *1   Horz Dipole....Short CE     1.51     1.48    AMPS ...
-B:HS4   *2.74Horz Dipole....Short CE     .92      .958    AMPS ...

HS4      :3
-B:HS3   *2.74Horz Dipole....Short CE     1.51     1.48    AMPS ...
-B:HS4   *1   Horz Dipole....Short CE     .92      .958    AMPS ...
-B:HS5   *2.74Horz Dipole....Short CE     -.86     -.848    AMPS ...

HS5      :3
-B:HS4   *2.74Horz Dipole....Short CE     .92      .958    AMPS ...
-B:HS5   *1   Horz Dipole....Short CE     -.86     -.848    AMPS ...
-B:HS6   *2.74Horz Dipole....Short CE     .245     .261    AMPS ...
```

A typical parameter page

Controls Rookie Book

❑ **Pbar Front End:** This is the Antiproton front end, a VME crate, located in the Computer Room. It is used to interface and control devices on the Antiproton Serial CAMAC Link.

❑ **Periodic Interrupt:** This is a software term used to describe an aspect of program execution timing. For example, all of the console computer processors have a 15-Hz interrupt which signals the TVMgr, and hence the applications program, to execute some section of code at this rate.
(See also *Interrupt*)

❑ **PIOR (Programmed Input/Output Retrieval):** PIOR is standard hardware for receiving data from the CAMAC links around the accelerator.

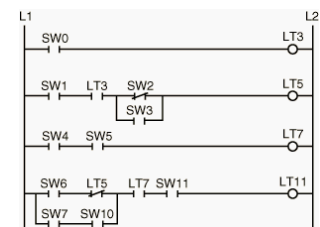
❑ **PIOX (Programmed Input/Output Transmission):** PIOX is standard hardware for transmitting data to the CAMAC links in the accelerator.

❑ **Pixel (Picture Element):** A single point on a computer monitor; the smallest unit used to construct characters or images.

❑ **Platform:** The basic comprehensive definition of a computer system. This would be its specific CPU and its specific version of operating system (OS) software. This basic information is what allows for the development of products and interfaces that (almost) can't possibly fail.

❑ **PLC (Programmable Ladder-logic Controller):** Intelligent devices that are becoming more and more popular (as prices drop) for closed-loop control of valves, switches, robotics, and other controls circuits.

A software package, like **DMACS**, can provide a user interface to these devices, but is not required for normal PLC operation.



A typical electrical ladder diagram

Controls Rookie Book

- ❑ **PLD (Program Loader):** A managing program on the console computers that works in conjunction with CPLD to download applications programs from the VAX. See *Central Program Loader*.
- ❑ **POP (Post Office Protocol):** A protocol used to retrieve electronic mail from a mail server.
- ❑ **POPmail:** A client/server e-mail package. The POPmail client allows you to read e-mail on your local machine, but in order to receive e-mail you must have an account on a host machine that is acting as a POPmail server. (*See also: Electronic Mail, IMAP, POP.*)
- ❑ **Pop-up Window:** A window wherein the options are hidden until a titled field is clicked on. On any parameter page you will find the ♦COPIES♦ example of pop-up window. (*See also Pull-down menu.*)
- ❑ **Predet:** A hardware box that delays a timing event by a predetermined amount. Switches on the front of the box set Predets manually. They can be thought of as hardwired alternatives to the CAMAC 177 modules. They are used in the Pre-Acc for triggering devices in the domes and in the LINAC, as well as delays for scope triggers in the MCR.
- ❑ **Primary Application (PA):** Applications programs that allow interaction with a console user. Behind the scenes every named application has a unique PA number. These numbers are used about as often as your SSN is used, and for the same reasons — as an internal-use unique identifier. The same PA can appear on multiple pages. A couple of the most popular are the parameter pages (PA0052) and the index pages (PA0001).

Let's clear up another oddity while we're here. When is a PA not a PA? When the day came that we could display 3 Primary Application windows at once, the respective window titles weren't named PA1, PA2

Controls Rookie Book

and PA3, but rather, **Primary Application A**, **Primary Application B** and **Primary Application C** or (sadly) PA, PB and PC. Sometimes the deck is really stacked against the rookie operator.

- ❑ **Property Index (PI):** This is a **database**-recognized number which identifies a device's software property references such as name, descriptive text, alarm limits, basic control, etc.
- ❑ **Protocol:** From a networking perspective it's the set of rules by which different nodes communicate. See the chapter on *Networking*.
- ❑ **Pull-Down Menu:** A menu wherein the options are hidden until the menu's title is clicked on. On any index page you will find the ♦Cmnds♦ and ♦Pgm_Tools♦ examples of pull-down menus. (*See also Pop-up Window.*)
- ❑ **Pulse Train:** A sequence of electronic pulses generated by a device. The frequency is interpreted as a numerical value. Examples include the intensities for the various beam lines appearing on Channel 13, and the hardwired loss monitors for the Switchyard.
- ❑ **QPM Link:** A 50-ohm RG-8U coaxial cable located in the Tevatron tunnel that carries information essential to the quench protection system. TECAR, the Tevatron power supply controller, refreshes the information at a 720-Hz rate. The link carries information about the current in the Tevatron bus that the QPMs decode locally in order to evaluate conditions inside the magnets.
- ❑ **QXR Link:** The link that transmits the programmed waveform to the Quad Extraction Regulator crates. It originates at the QXR microprocessor in the MAC room (TeV) or MI-60 (Main Injector).

Controls Rookie Book

❑ **Reboot:** See *Boot*.

❑ **[DAQ] Redirection:** A term used to describe how data-acquisition save-data (from a D1, SDA or lumberjack file) can be viewed at your console. One could go, say, to the SAVE/RESTORE page (D1) and wade through the structured levels of data, or take a more meaningful approach and redirect that saved data to appear on the actual parameter page where you're used to seeing it. A yellow slash across the page reminds you that the page is in this alternate mode.

Here's how to do it: Go to page B8, subpage 1 of 400-MEV and hit the UTIL keyboard button (F14). Select DAQ Redirection in the pop-up window, then click on Directory to the right of "SR File". Click Booster, then select one of the most current save files. You'll then pop back to the redirection window where you should look at the top line and select the screen that is currently displaying B8 (it's on the window's title bar). Presto! Your parameter page is now displaying the latest save file information. To cancel redirection, go back to the redirection window and repeat your last interrupt.

This, by the way, is another example of an **Open Access Client** at work.

❑ **Repeater:** A module that is used in long transmission paths to boost a degraded signal. It receives information from a link and retransmits it to points further downstream.

❑ **RETDAT:** A program present on all front ends which receives request lists for data from consoles and organizes those lists to be sent out to the appropriate CAMAC crates in the field. RETDAT stands for RETurn DATa.

❑ **Rhone:** See **Wine Cluster**.

THE CONTROLSOPEDIA

- R -

Controls Rookie Book

❑ **Ring [Networking] Topology:** See **Topology**.

❑ **Ripple:** Used most often in the context of power supplies, it's an unwanted signal (of a higher frequency structure) that's gotten introduced onto the desired output, usually because of an internal component failure in the filter of the supply.

❑ **Router:** A networking device slightly more intelligent than a bridge, it routes messages based on internal road maps called routing tables. Operators tend not to get into heavy conversations on routers, but see the chapter on *Networking* for more exciting details.

❑ **Sample and Hold:** This is a circuit, used throughout the accelerator which, upon command, records the (analog or digital) value of a device in a buffer and holds this value until it receives a command to reset its buffer. This allows data to be accumulated for, say, the course of an entire cycle, then uploaded in one transmission when the cycle ends. The circuit would then be reset (cleared) for the next cycle.

❑ **Sample Time:** The time, unique to each accelerator, when data is loaded into buffers and sampled by the front ends.

❑ **SCADA (Supervisory/System Control And Data Acquisition):** A system used to acquire remote information, especially on power systems. It has datalogging capabilities and a cutesy graphical interface. We use it to monitor most of our power utility equipment, like 13.8Kv feeders.

❑ **Save File:** This term generally refers to a FileSharing file containing some type of data or information relating to accelerator operation. These files are generally temporary. The purpose of these files is to record current

Controls Rookie Book

operating conditions in nonvolatile memory in the event they need to be restored at a later date.

❑ **Scaling:** The conversion of MADC voltages into engineering units, like furlongs-per-fortnight, which humans can understand. *Scale factors* are a predefined set of algorithms, defined within the database. When it's initially defined, the proper algorithm is specified for a given device.

❑ **SCR:** (1) Our **Save/Compare/Restore** program which reads the current value of D/A settings, A/D readbacks, nominal values and alarm status and stores this information into a **save file**. One can then compare or restore the current information with that previously stored.

(2) An acronym for *Silicon Controlled Rectifier*. A solid-state gated diode used extensively within power supplies.

❑ **Screen Image Editor (SIE):** A primary applications program, currently found on D10, which allows editing of the initial image displayed by any console program.

❑ **Scripting:** Perhaps the easiest way to describe a script is a list of manual operations to be executed automatically. Cursor movement, keyboard input and delays may be recorded and saved. The script can then be executed, whenever necessary, to replicate the sequence, whether it be for a tutorial, or as a time-saving operation. Compared to **sequencer** operation, a script is much less complex and was designed for a much more single-focus task.

❑ **SDA (Shot Data Acquisition):** A central process that captures scalar and snapshot data during collider operation, for use in shot analysis.

❑ **Secondary Applications (SA):** An application that is intended to run as sort of a remote display (such as the ever-popular cryo temperature

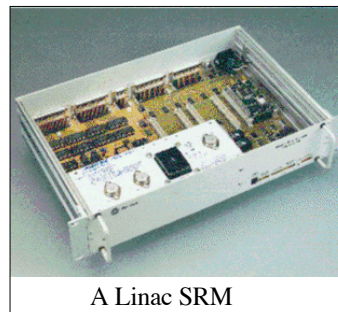
Controls Rookie Book

display). One typically uses an associated PA to establish an SAs optional *parameters*, then perform the launch. An SA continues to update its display with fresh data, although one is free to leave the initializing PA once the SA is launched. (See also Primary Application.)

- ❑ **Serial:** Unlike in **parallel** transmission, data on a serial link is transmitted one bit at a time. Many of the links at Fermilab are serial links. Normally this data must be reconstructed into a parallel format before it can be used at its destination. Of course the advantage is that the link can be made with a single twisted-pair wire rather than the multi-conductor monstrosity necessary for parallel transmission.
- ❑ **Serial Link Driver [SLD]:** A VME CAMAC module that acts as a parallel-to-serial converter between a given front end and its associated serial link.
- ❑ **Server:** See *Client/Server Architecture*.
- ❑ **SETDAT:** A program present on all of the front ends; it organizes lists of D/A setting values before they are sent to the hardware.
- ❑ **Set Point:** (1) In the context of feedback systems, it is the nominal value toward which the system strives. (2) See *Nominal*.
- ❑ **Sequencer:** This is a program designed to be used in an operational procedure where a precise progression of steps is imperative. It will base its progression on real-time readings, accelerator timing, and interactive human response. Unlike a **script** that tends to have a very simple and limited scope, a *sequence* can be quite lengthy and involve many page changes and built-in delays.

Controls Rookie Book

- ❑ **ShowBiz:** A VAX **turnkey** account that replicates the **Channel 13** display. It's a handy medium for anyone with a **modem** to log into, whether from a conference or grandma's house.
- ❑ **Sibling [device]:** A device that is logically or sequentially related to other devices. — for example, two consecutive horizontal correction dipoles.
- ❑ **Signal Distribution Panel:** One or more of these are located at an MCR console and sport various hardwired signals that Operators feel the need to monitor, usually on a scope. (*See also: miniframe, mainframe.*)
- ❑ **Smart Module:** A relative term used to designate a module which can do some amount of internal data processing or error checking. All of the microprocessors, such as the BPMs, QPMs, and refrigerators, can do this as well as many smaller units such as 080, 160, and 165 cards.
- ❑ **Snapshot:** In general terms, this refers to data taken at some specific time over a short interval. The data obtained may or may not be representative of events over a longer time interval. A specific use is the Beam Position Monitor (BPM) data averaged over a number of turns and stored in a **circular buffer**. Another important use is for diagnosis of problems; for example, when the ramp trips a snapshot of the voltage-to-ground is taken and may be placed in a circular buffer to be analyzed at a later time.
- ❑ **Software Documentation Memo (SDM or SODOM):** A once-popular publication distributed by the Controls department that is intended to clarify operational aspects of software usage. This is primarily used for descriptions of major software utilities, control system theory, and not for application program documentation.



Controls Rookie Book

(See also Help Facility.)

- ❑ **SRM (Smart Rack Monitor):** An intelligent data-gathering system used in the Linac controls system.
- ❑ **SSDN (Sub-System Device Number):** Each database entry in a given subsystem (e.g. Linac, Tevatron, etc.) has a unique number assigned to it. This number is used by the front ends and describes characteristics such as module type, location etc. The format of these numbers is different between front ends and different for each property of a device.
- ❑ **Staging:** This term is used to refer to the downloading of a program to your console; see **CPLD** for more details.
- ❑ **Star [Networking] Topology:** See **Topology**.
- ❑ **Status Display Monitor:** See *Alarm Screen*.
- ❑ **Stepping Motor:** A motor that rotates a finite amount in response to an encoded signal. In switchyard, for instance, it allows beam-line elements to be moved precise distances.
- ❑ **Supercycle:** The Supercycle time is the amount of time required to complete all the different machine cycles, which have been defined to operate periodically as a unit. This implies that the Supercycle time is periodic and that its length is dependent on the number and duration of different machine cycles defined to occur within its period. Sort of like our calendar is made up of numerous cycles of hours, minutes, days, weeks and months, they all make up a **time-line** of events called a year — viola, the essence of a Supercycle.

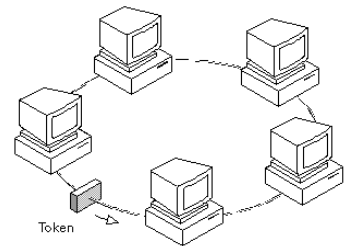
Controls Rookie Book

- ❑ **Supercycle Reset:** The signal, or clock event that triggers the end of the current **time-line** or **Supercycle** and the beginning of the next. From the previous (Supercycle) analogy, we'd call it midnight on New Year's Eve. (*See also Machine Reset.*)
- ❑ **Switchyard Front End:** The Switchyard front end is a VME crate located in the Computer Room. It is used to interface and control devices on the Switchyard Serial CAMAC Link (see the *Links* chapter).
- ❑ **Sybase®:** Our primary database management engine, it is the registered trademark of Sybase, Inc.
- ❑ **TCLK:** Pronounced T-clock, the name implies that it is only used for Tevatron, however it is essentially the master clock for the accelerator. It operates at a 10-Mhz rate. For more information on accelerator clocks, see the *Timing* chapter.
- ❑ **TCP/IP (Transmission Control Protocol / Internet Protocol):** The protocols used to connect hosts to the internet and, for that reason, have become the defacto standard for data networks. See the chapter on *Networking* for more information.
- ❑ **TECAR (Tevatron Excitation Control And Regulation):** The microprocessor system that sends phasing information to the Tevatron's main power supplies.
- ❑ **Terminal:** The point at which the user interacts with the computer system — a keyboard console.

THE CONTROLSOPEDIA
- T -

Controls Rookie Book

- ❑ **TeV Front End:** The Tevatron front end is a VME crate, located in the Computer Room. It is used to interface and control devices on the Tevatron Serial CAMAC Link.
- ❑ **Tevatron Serial Crate Controller [TSCC]:** A set of modules that occupy the two right-most slots in a CAMAC crate. This is true, even in non-Tevatron crates — the name has stuck only because the TeV had first dibs. The purpose of the TSCC is to recognize messages addressed to it and then to perform a serial-to-parallel conversion on the incoming data and finally to direct it to the proper card in the associated crate.
- ❑ **Timeline:** A sequence of significant events. See *Supercycle*.
- ❑ **TLG (Time-Line Generator):** Operators control the **timeline** via the *ACCELS TIME-LINE* page (D69), therein defining when all of the significant events are to occur in the current **Supercycle**. The associated TLG hardware (including a backup) is supported in a VME crate in the Mac Room (racks 17 and 18).
- ❑ **Token Ring:** A network access method in which all the computers are arranged (schematically) in a circle (ok, *ring*). A token, which is a special bit pattern, travels around the circle. To send a message, a computer catches the token, attaches a message to it, and then lets it continue to travel around the network. See the chapter on *Networking* for more information on this and other access methods.
- ❑ **Tolerance:** The amount that the readback of a device can deviate from its **nominal** value before going into an alarm state.



Controls Rookie Book

❑ **Topology:** In a networking sense, topology is a description or drawing of how one's system is laid out. The three most popular topology types are star, ring and bus. For a more in-depth look at network topologies, see the chapter on *Networking*.

❑ **Touchpanel or Touchscreen:** These are referenced within a few applications pages and, in context, refer to special on-screen windows made available for the purpose of displaying a program-independent interactive control panel. In the past these were actually small monitors with touch-sensitive screens. Touch the panel with your finger, elbow, book, coffee cup, fist, or what-have-you and exciting things would happen — seldom the thing you wanted. So, troublesome buggers that they were, the separate monitors were abandoned; they're now point-and-click windows — a much more reliable approach.



❑ **T-Times:** Because of the myriad of ramping schemes this term has lost meaning within the Beams Division (except in its golf context), however it is still a buzz-word within the external beamlines area. It refers to the major power supply ramp events, as denoted in the figure.

❑ **Turnkey Account:** A no-password computer account.

❑ **Two-tier Architecture:** See Client/Server Architecture.

❑ **TVBS:** The *Tevatron beam-sync* clock. It is derived from a subharmonic of the Tevatron RF frequency.

Controls Rookie Book

❑ **TVMgr:** This process (which runs on every console) takes care of the console X-windows character management. One of its tasks, for instance, is dealing with those character cells in the upper left corner reserved for page numbers.

THE CONTROLSOPEDIA

- U -

❑ **UCD (Universal Clock Decoder):** This device generates a 15-Hz signal from the Tevatron clock for internal use and is used to decode specific clock events as needed.

❑ **Upload:** See *Download*.

❑ **UPS (Uninterruptible Power Supply):** A power supply that continues to provide electricity despite the loss of AC input power. This is made possible by means of a backup battery and a DC-AC inverter or DC-DC converter.

THE CONTROLSOPEDIA

- V -

❑ **VMEbus:** (VersaModule Eurocard) An architecture that was developed in 1981 to fill a need to support the Motorola 68000 processor family. This open architecture design is widely used. For those who keep track of such things, this is IEEE standard 1014.

❑ **VMS:** The VAX operating system software.

❑ **VXIbus:** A standard bus architecture considered better for low signal-level applications than VME. (*See also Crate.*)

❑ **VxWorks:** A fast, multitasking kernel with preemptive scheduling and fast interrupt response, extensive intertask communications and synchronization facilities, efficient UNIX-compatible memory management, multiprocessor facilities, a shell for user interface, symbolic

Controls Rookie Book

and source level debugging capabilities, performance monitoring and an I/O file system. Now, if someone asks (and they won't), you can tell them all about it.

- ❑ **WAN:** A Wide Area Network usually refers to the network that ties all of the local networks. (*See also LAN.*)
- ❑ **Word:** In computerese this refers to data size and is dependent on the particular computer. For a computer incorporating a 16-bit CPU, a word is 2-bytes or 16-bits; a *long-word* would be 32-bits or 4-bytes; a *quad-word* is 64-bits.
- ❑ **Wine Cluster:** This is a set of computers that can be viewed as ACNETing (the next generation). It's part of an on-going effort to move us away from the VAX platform. These things tend to come out of nowhere, probably because (like so many things) they start as an experiment and then just continue to hang around once the experiment pans out. The ones that have been around the longest are *Rhone* and *Chianti*; both are Pentium Pro PCs running UNIX. Rhone is our device database server and Chianti is our application database server. Physically, these have taken up residence in the racks located at south end of the west wall in the Computer Room. Does this mean UNIX is our path to the future? So far no one's committing themselves.
- ❑ **World Wide Web (WWW):** The *Web*. See *Internet*.

THE CONTROLSPEDIA

- W -

Controls Rookie Book

THE CONTROLSOPEDIA

- X -

- ❑ **X-Windows:** Standard set of display-handling routines developed at MIT for UNIX workstations; they allow the creation of hardware-independent graphical user interfaces. What makes them particularly attractive is they're free. All of our console displays are via X-Windows.