



UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE



INTERNATIONAL
COMMITTEE FOR
FUTURE ACCELERATORS

ISTITUTO **INFN**
NAZIONALE
DI FISICA NUCLEARE

H4.SMR. 394/11

SECOND ICFA SCHOOL ON INSTRUMENTATION IN
ELEMENTARY PARTICLE PHYSICS

12- 23 JUNE 1989

Computerized Data Acquisition Systems in Particle Physics
Experiments

H. BEKER

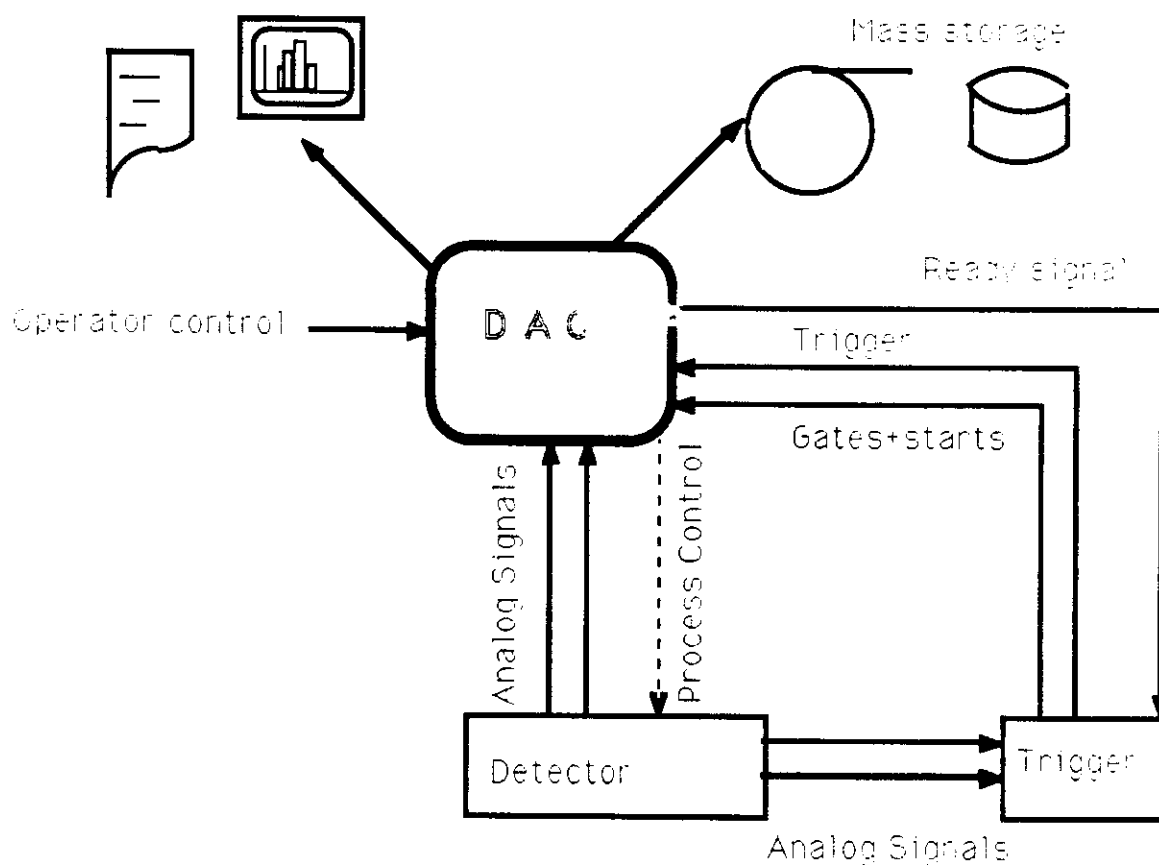
CERN, Geneva, Switzerland

These notes are intended for internal distribution only.

Following fields will be discussed:

- General layout of DAQs
- Digitizing electronics (sensors) and actuators
- Peripheral bus systems
CAMAC, (FASTBUS, VME)
- Interfaces
- Acquisition Computers
- Storage media
- Software aspects
- Advanced software concepts
 - data reduction
 - filtering
 - 2 level software trigger
 - parallelism
 - pipelineing
- Examples
Valet +, SPA³W, HELIOS

Performance checks



Definitions:

Trigger system: views the detector state continuously using a subset of the analog signals, to define the event time and decide whether the event is 'interesting'. Through the gating logic it provides a 'snap shot' of the detector state. Provides a digital signal to the DAQ to cause the read out of the detector. It has to be inhibited during the read out. It is notified by the DAQ (clear signals) on end of read out.

'Analog' Signals: Coming directly from the detector, amplified and shaped in an appropriate way.

'Digital' Signals: These signals are used in the trigger, and for the communication between the trigger and the DAQ. They have binary values so that boolean algebraic combinations can be performed on them. Unlike in the classic boolean algebra the time domain of this signal is important. Here we name only signals which interface the data acquisition with the trigger and gate logic.

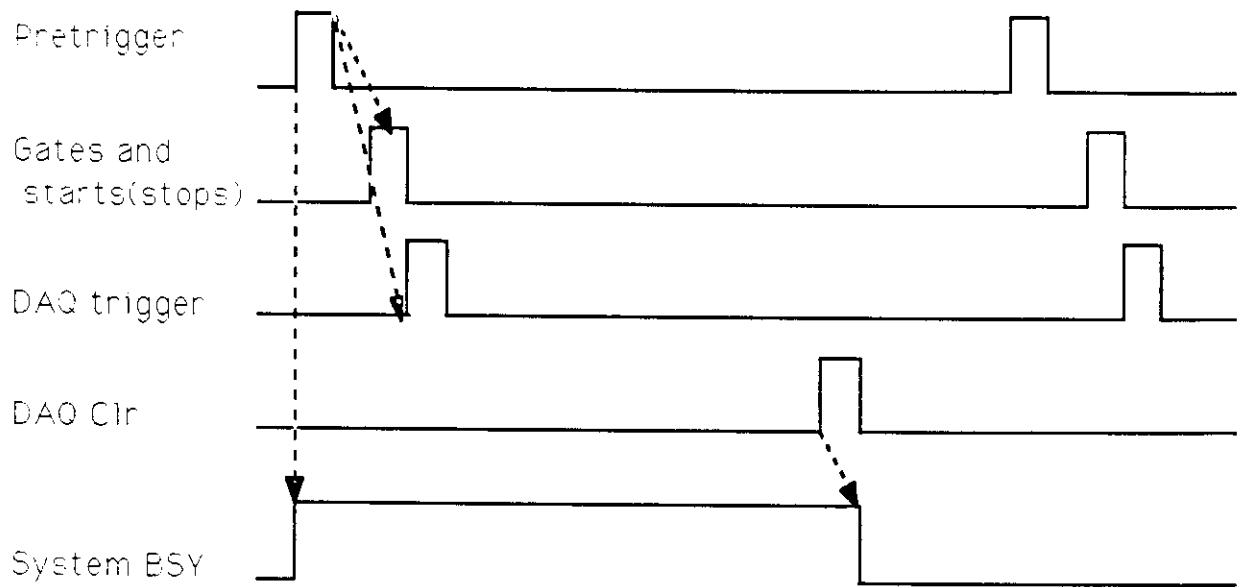
Trigger signal(s): Tells the read out machine that an event has been recognized and been 'frozen' on the inputs of the digitizing electronics. The trigger signal can either be a simple pulse or can contain more detailed information on the cause of the trigger.

Gate: A pulse of a fixed width with a fixed delay to the occurrence of an 'interaction'. Used to define the exact observation time of the detectors in the digitizing electronics(ADCs)

Starts and Stops: similar to gates however they define the reference for time and combined time/pulse height measurements. (TDCs, FADCs). Therefore only the leading edge of this signal is of interest.

Fast Clear: Gates, starts and stops are usually caused by a very low trigger level. They initiated the internal digitization in the electronics. In case a higher trigger level decides to discard the event it has to issue a fast (i.e. hardware) clear to prepare the read out electronics for a new event without (slow) software intervention.

Clear: Issued by the the DAQ to indicate that is has finished the read out. This signal restarts the trigger and can be used to clear the digitizing electronics.



Performance numbers:

Dead time: fraction of total time the system cannot accept physics events.

There are several levels of dead time:

detector dead time = total signal collection time (10's of nanoseconds to several μ secs)

trigger: decision time (several 10 nseconds to 10's of μ sec)

DAQ: digitization + read out time (100's of μ sec to 100's of msec)

Another (easier) way of estimating the dead time is:

$$t_{\text{dead}} = \frac{N_{\text{events recorded}}}{N_{\text{events occurred}}}$$

This performance number adds up all levels of dead time.

How to calculate the necessary read out dead time for a given physics process: (assuming that the read out dead time is dominating)

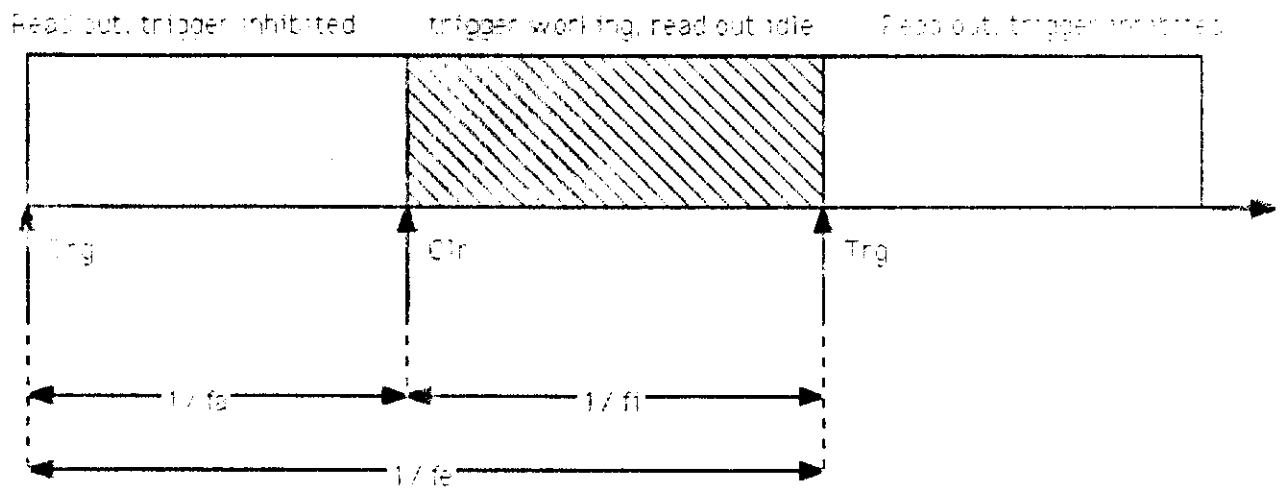
f_i frequency of process

f_amaximum acquisition frequency

f_eeffective acquisition frequency

$$f_e = \frac{f_i f_a}{f_i + f_a}$$

If we intend to exploit 80% (i.e. 20 % dead time) of the given physics process, i.e. $\frac{f_e}{f_i} = 0.8$ then f_a has to be 4 times f_i . That means for a physics process as infrequent as 10 Hz we have to aim for no more than 25 msec read out time.

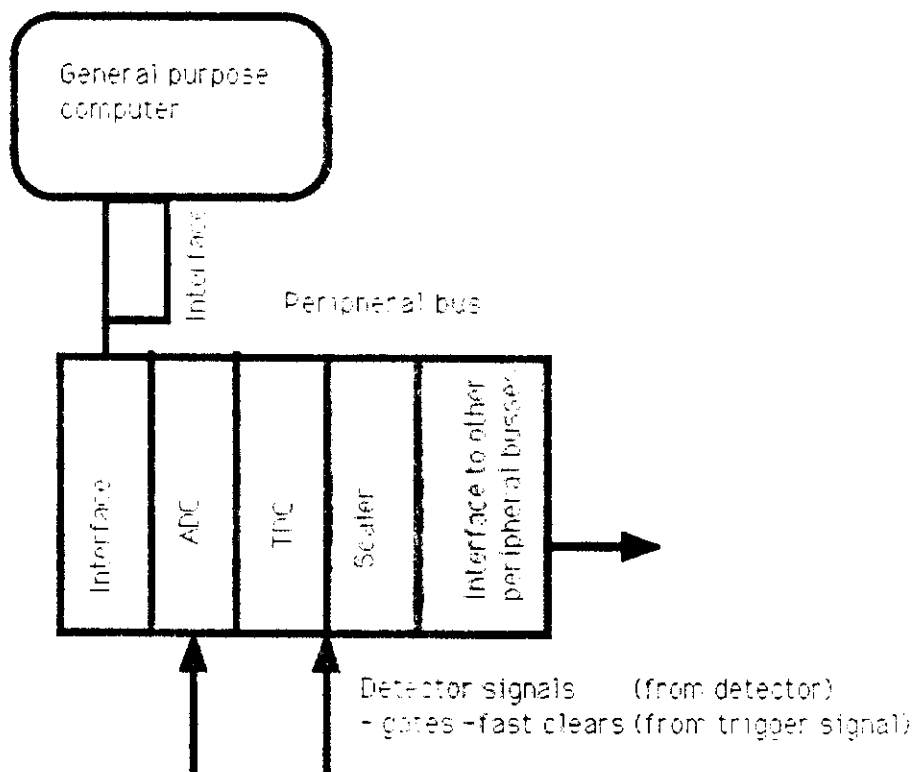


Life time = (1-dead time)

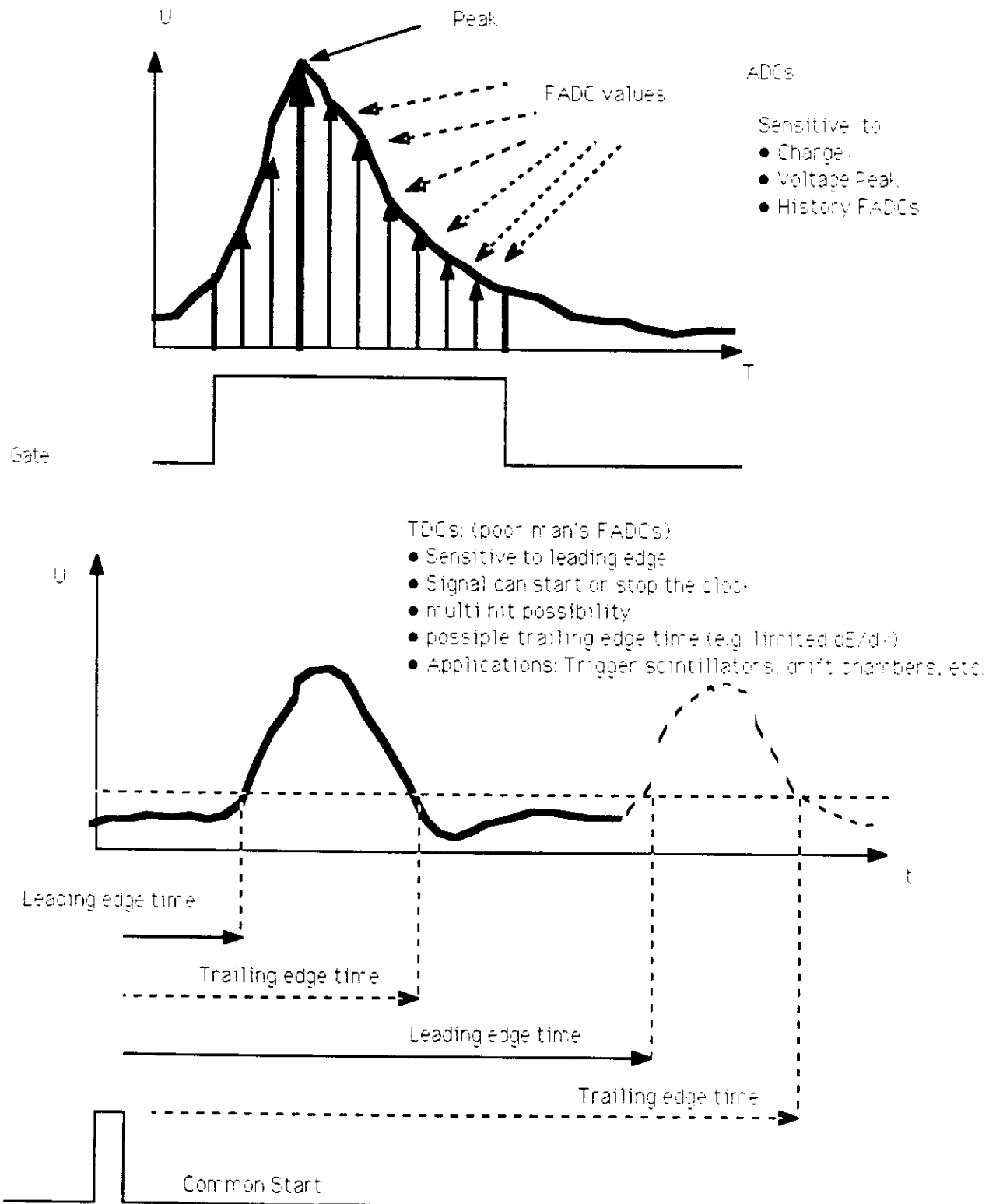
Throughput:

The amount of data recorded per unit of time = average event length * f_e

Closer look at DAQ:



Sensors:



Scalars:

Simple counter. they are usually ungated and reset at regular intervals. Typically discriminated detector signals are scaled as well as signals in the trigger logic. They can give you an idea on the absolute cross section of the observed process as they also record events in which the whole detector is not read out.

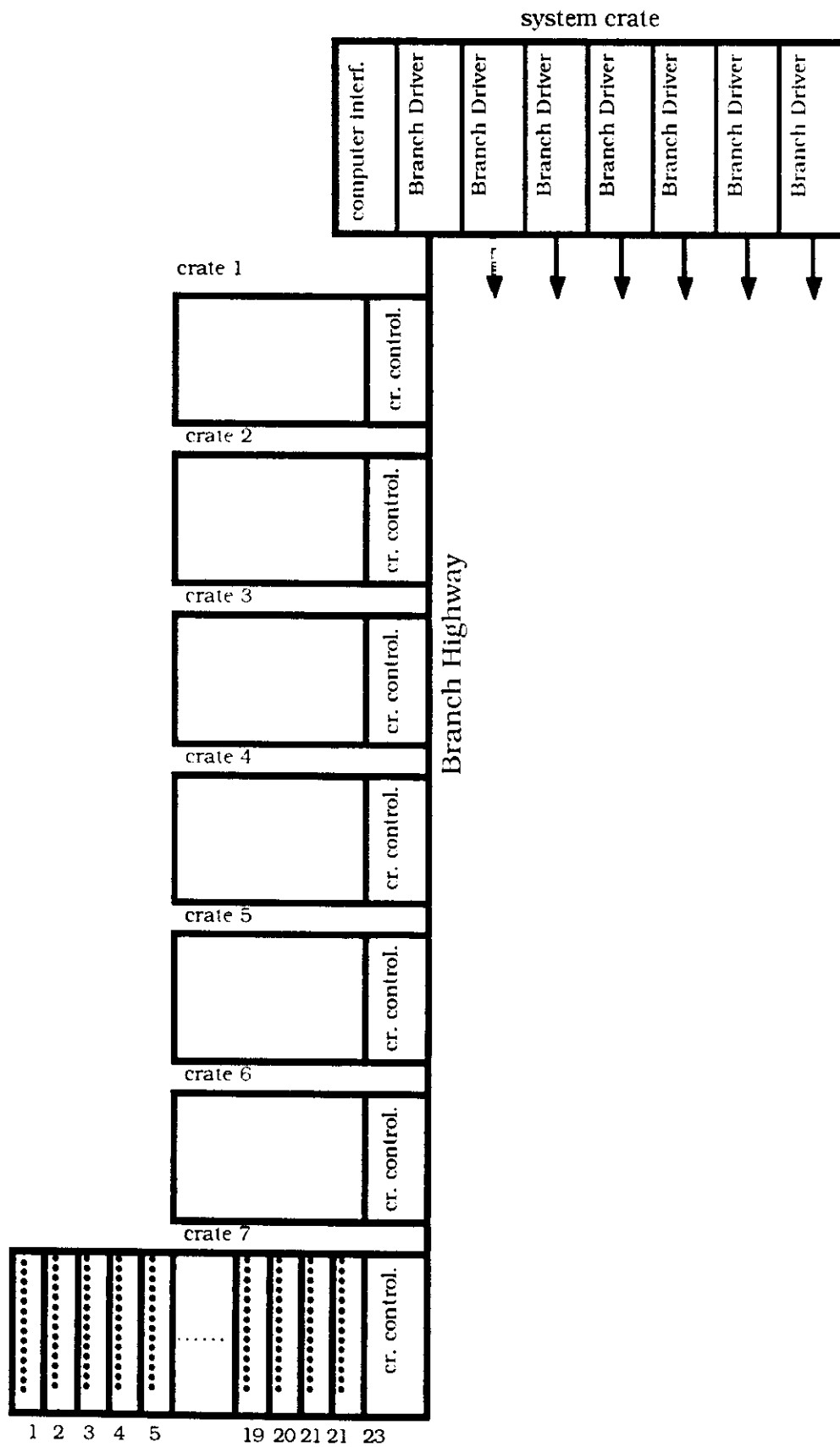
Actuators:

Can be used to control (i.e. influence) the observed process.

e.g. DAC
 Output register
 Switches, Relays
 High voltage controls
 Programmable Logic Gate Arrays
 etc.

Peripheral Bus systems

CAMAC



It's a peripheral bus system developed for the needs of particle physics by the beginning of the 70's, allowing the treatment of the experimental equipment in a general way independent of the host system (computer). By now the standard has also been widely accepted by industry and interfaces to numerous real time machines are available. It offers most probably the widest range of digitizing electronics and control modules among all peripheral bus systems.

The position of a CAMAC device is defined by following geographical address:

BRANCH	up to 7
CRATE	"
SLOT	23 physical slots + 9 virtual
SUBADDRESS	16 usually defining the input on the board

A complete CAMAC bus can therefore contain up to 49 crates with a total of 1000 module and roughly 15000 data channels.

The geographical address is used by the host computer to connect to a device.

Each device can respond to up to 32 possible functions:

F 0..F 7	read functions
F 8..F15	test or control (dataless) functions
F16..F23	write functions
F24..F31	test or control (dataless) functions

The data which can be exchanged with the devices can be either 16 or 24 bit wide words.

The success of each access on the bus can be checked in a hardware status register. There are two levels of command acknowledgement:

- The X-response tells whether the addressed module was present
- The Q-response tells whether the applied function could be executed successfully.

The Q-response very often depends on the state of the device, e.g. an ADC will issue a positive Q-response to a read function only if the internal digitization has terminated, a multihit TDC will issue the Q-response only when there are still more hit times to be read out etc.

CAMAC also offers interrupt capability with a limited prioritization scheme. Every module can raise a LAM (Look-At-Me) spontaneously.

This LAM can be polled using special read functions. Most computer interfaces also can pass a LAM as an interrupt to the host computer, which services this request in a routine connected to this interrupt source. Such interrupts are referred to as GLAM(graded LAMs) as they can be prioritized. However a CAMAC interrupt can never be interrupted by another GLAM. All LAMs arriving in the mean time, however, are ordered according to there interrupt level. (0..23)

Within certain limits CAMAC lends real times features to the computer it is interfaced to, even though this machine was not designed for real time applications without impairing the user friendliness of the general purpose machine.

There is the possibility to have one auxiliary bus master in each crate. arbitration can be provided. The branch high way is usually not arbitrated and allows only one bus master.

Many computer interfaces provide a DMA mode for read and write functions. Several address increment algorithms are possible and several different DMA termination conditions are available:

QIGNORE QSTOP QREPEAT QSCAN

With modern microprocessors and interfaces the time gain over the single word transfer is very often negligible and these functions are emulated in software.

Typical transfer speeds in DMA mode are slightly under 2 μ secs/word (independent of data width) which results in a minimum read out time of a 10 kword event of some 20 msecs. As already mentioned this might not be sufficient in a modern high energy experiment. As will be shown later on there are ways to overcome this limitation by applying parallelism in the read out.

For future high rate experiments, however, CAMAC will not suffice any longer. It will nevertheless prevail in many small experiments and in the slow control even of complex experiments for another few years to come, due to the relatively low cost, the simplicity of its use and above all the commercial availability of a huge number of components.

There are interfaces to following computers:

VAX + PDP(UNIBUS, Q-bus, BI)	NORD	VME (VALET+)
FASTBUS	IBM-PC	MacIntosh
CAVIAR	HP	

Example:

ESONE routines:

CDREG(IEXT,IB,IC,IN,IA) : declare CAMAC register
 CFSR/CSSA(IFUN,IEXT,IDATA,IQ): execute a CAMAC command

CCOZ : generate dataway initialize
 CCCC : generate crate clear
 CCCI : set/clear dataway inhibit
 CTCI : test dataway inh bit
 CCCC : enable/disable crate demand
 CTCD : test crate demand enabled
 CTGL : test crate demand present

CDLAM(LAM,B,C,N,M,INTA): declare LAM
 CCLM(LAM,L): enable/disable LAM
 CCLC(LAM): clear LAM
 CTLM(LAM,IA): test LAM
 CCLNK(LAM,SUB): link LAM to a service procedure
 CCRGL(LAM): re-enable LAM interrupt
 CCULK(LAM,SUB): cancel link between LAM and service procedure
 CCLWT(LAM): wait for LAM

CDCHN(CHAN,NL,S,F): define end of QSCAN
 CFUBC(F,EXT,INTC,CB) Qignore or Qstop
 CSUBC(F,EXT,INTT,CB)
 CFUBR(F,EXT,INTC,CB) Qrepeat
 CSUBR(F,EXT,INTT,CB)
 CFMAD(F,EXTB,INTC,CB) Qscan
 CSMAD(F,EXTB,INTT,CB)

1.

USE OF LAM, BUT NOT GRADED LAM:

```

C
PROGRAM TLAM1
  IMPLICIT INTEGER (A-Z)
  DIMENSION INTA(2)
  LOGICAL L,T
  T=.TRUE.
  PRINT 10
  READ 20,B,C,N
  INTA(1)=0      ; NO GL
  INTA(2)=0      ; NO EVENT FLAG
C  ASSIGN CHANNEL
  CALL CDSET(ICN,0)
C  DEFINE ADDRESS
  CALL CDREG(EXT,B,C,N,M)
  CALL CDLAM(LAM,B,C,N,M,INTA)
C  CLEAR LAM
  CALL CCLC(LAM)
C  ENABLE LAM
  CALL CCLM(LAM,T)
C  RESET TRIGGER
  CALL CSSA(16,EXT,1,Q)
  EVNR=1
C  TEST LAM
1000 CALL CTLM(LAM,L)
1001 CONTINUE
      IF(L.NE..TRUE.) THEN
            CALL READOUT
            CALL WRITETAPE
C            CLEAR LAM
            CALL CCLC(LAM)
            CALL CSSA(16,EXT,1,Q)
            IF (EVNR.LT.EVMAX)GOTO 1000
            STOP
      ENDIF
      GOTO 1000
C
C ...
10
+FORMAT(' ENTER BRANCH,CRATE,STATION,SUBADDRESS')
20 FORMAT(3Z8)
END

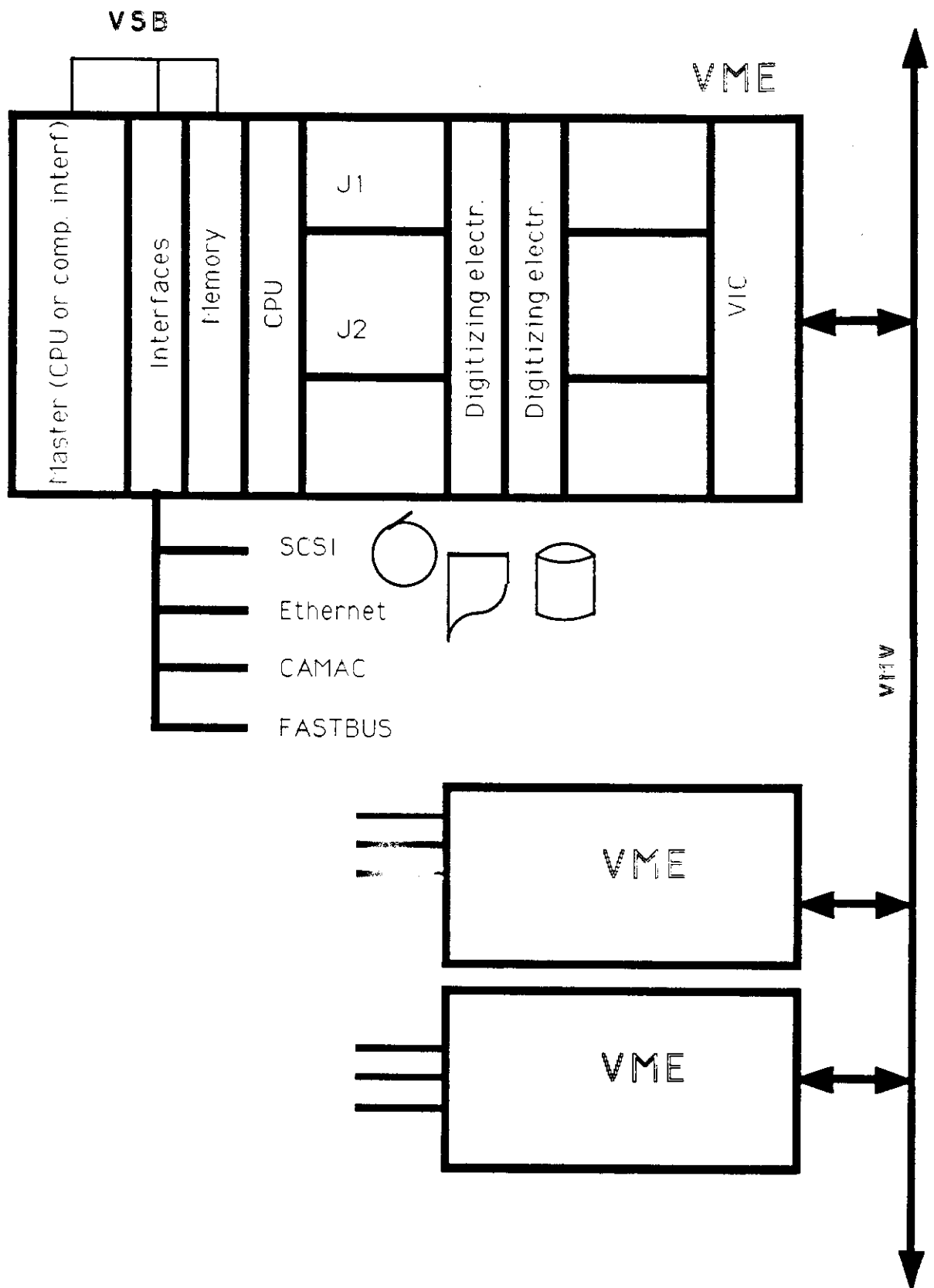
```

```

2.      USE OF GRADED LAMS (INTERRUPT DRIVEN)
      PROGRAM TLAM2
      IMPLICIT INTEGER(A-Z)
      DIMENSION INTA(2)
      COMMON/DDD/INTA,EXT,LAM,EUNR,EUMAX
      LOGICAL L,T
      EXTERNAL READOUT
      T=.TRUE.
      100 PRINT 10
      READ 20,B,C,N,M,INTA(1),INTA(2),EUMAX
      C   ASSIGN CHANNEL
      CALL CDSET(ICN,0)
      C   DEFINE ADDRESS
      CALL COREG(EXT,B,C,N,M)
      CALL CDLAM(LAM,B,C,N,M,INTA)
      C   CLEAR LAM
      CALL CCLC(LAM)
      C   LINK GL TO EVENT FLAG AND BOOK LAM
      CALL CCLNK(LAM,READOUT)
      C   ENABLE LAM ON MODULE
      CALL CCLK(LAM,T)
      EUNR=0
      CALL SLEEP
      10 FORMAT('BRANCH,CRATE,STATION,SUBADDRESS,GL,EVENT FLAG,EUMF
      20 FORMAT(7I4)
      END
      SUBROUTINE READOUT
      COMMON/DDD/INTA,EXT,LAM,EUNR,EUMAX
      INTEGER*4 EVENT(10000),CB(2)
      C   CLEAR LAM
      CALL CCLC(LAM)
      CB(1)=500
      CALL CSMAD(2,EXT,EVENT,CB)
      C   RE-LINK LAM
      CALL CCAGL(LAM)
      C   ENABLE LAM ON MODULE
      CALL CCLM(LAM,T)
      EUNR=EUNR+1
      WRITE(10,1000)(EVENT(I),I=1,CB(2))
      1000 FORMAT(16(' ',24))
      IF(EUNR.GE.EUMAX) THEN
          CALL CCULK(LAM)
          STOP
      ENDIF
      END

```

VME:



This bus system has been developed by the beginning of this decade by Motorola, and was tailored to processors of the 680xx type, but is by no means limited to these. (transputers, NEC, PCs etc)

Contrary to CAMAC or the IEEE bus it is not a mere peripheral bus. Because of its data rate and functionality it is much better to compare it to the system bus of micro and mini computers. Aside from that a certain number of truly peripheral cards are available in this bus system. However, the higher bus speed (40 MHz) causes problems to incorporate high accuracy digitizing electronics.

Compared to CAMAC VME today is in very wide spread use in many fields of industry and particle physics represents only a very minor user group. Therefore a huge number of modules are available commercially at very moderate cost. The standardization has advanced enough to be able to build up one's tailor made system from modules of many different companies.

The VME standard describes the mechanical measures, the number and use of lines on the back plane as well as the used protocols. No software standard has been realized yet.

The VME bus supports data transfers at a speed of 40 MHz at data widths of up to 32 bits. The address space is also 32 bit (4 Gigabyte) deep. The system bus is physically separated into two back planes. The second one is optional. If it is not present the address space is limited to 24 bit (16 Mbyte) and the data width to 16 bit. The latter configuration matches the addressing capabilities and data width of the 68000 and 68010 processors. The bus has only data address space and no I/O address space in compliance with the 68000 philosophy of memory mapped I/O.

The addressing is only logically as opposed to the geographical scheme in CAMAC. There a module can be put in any free slot without having to tell the software where it sits. Usually the base address can be set on the card using jumpers or the like. The system designer, however, has to take care that no address is doubly assigned.

VME is a multi **master** bus. One of the masters in a crate has to be equipped with an arbiter. The bus requests of different masters can be prioritized on four levels. On one level the priority goes from left to right in the crate. The arbitration is done bus access by bus access i.e. word wise so that the user usually does not have to care about the arbitration logic (unlike FASTBUS!!) The TAS function of the 68000 (indivisible read and write function) is, however, supported across the bus. (Semaphores)

The system has been designed as an one crate system, i.e. sofar the standard does not define an inter-crate connection. An inter-crate connection most likely to be compatible with the standard in preparation is already available. It allows to connect any number of VME crates with various topologies.

Both masters and slaves can issue service requests(interrupts). Masters can respond to them. There are 7 interrupt levels often directly looped through to the 68000 processor interrupts. Passing an 8 bit vector do determine the cause of the asynchronous request is optional.

Bus and address errors are handled independently of service requests.

The VSB Bus

Many VME modules are **dual-ported** supporting the VSB bus on the second bus. By these means the VME bus can be off-loaded. Any subset of VME modules in one (or different) crates can be interconnected by a private VSB bus which is realized only by a flat cable plugged to the P2 connector.

Dual ported masters decide whether to address the VME bus or the VSB bus by either waiting for a bus error on the VME bus before accessing the VSB end or reserve a certain portion of the address space of the processor for the VSB bus.

The functionality of the VSB bus as compared to VME is somewhat limited(arbitration etc.). Still interrupts and multi master operation are possible.

FASTBUS:

This bus was designed for the growing demands with respect to acquisition speed and number of data channels by the beginning of the 80's. It covers both the needs of the digitizing electronics and the intelligent processors which have to process the data stream. In that respect its design is somewhat similar to that of the VME and thus they will be contrasted here.

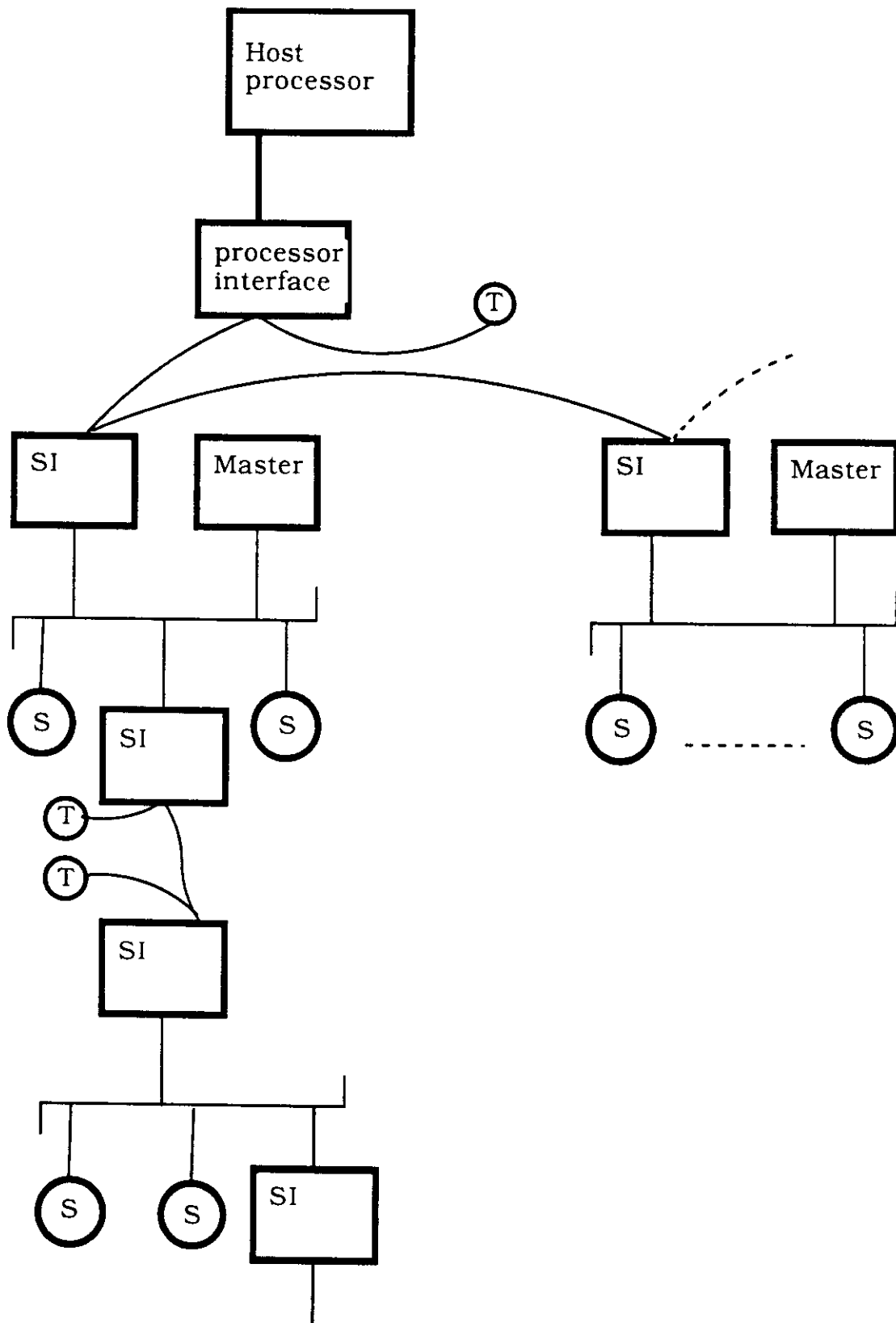
Where as in VME the emphasis is definitely on the processing side, FASTBUS stresses more the integration of the digitizing electronics.

Since the applications of FASTBUS sofar are mainly limited to the needs of particle physics for which it was designed, only a relatively small number of modules is commercially available. Many boards have been developed, however, by institutions connected to high energy physics. One big advantage of FASTBUS is the increased board size as compared to the previously discussed bus systems (4 times) and the higher power consumption each board is allowed. This fact facilitates the integration of a high number of channels in just one crate (typically 1000 - 2000).

FASTBUS encourages the use of crates for speed reasons, nevertheless the protocols can be run on flat cables. The use of ECL logic largely increase the transfer speeds (10 MHz). Again data widths and address occupy 32 bit. Both logical and geographical addressing is supported. Aside from the word by word, point to point transfers block transfers with a single addressing cycle, with and without handshake, as well as broadcast transfers are defined in the standard. Multiple masters on one cable segment are supported. Arbitration is usually achieved on the block rather than on the word by word level increasing throughput.

A cable segment, usually a crate, can be interconnected with another using SIs which form a crate to crate, point to point connection (versus a bus concept). The interconnection topology can be setup up in way that optimizes time-critical data paths. (Segments exchanging a lot of data can be put close to each other in the network)

Another supported technique of linking segments is the use of buffered interconnects and segments extenders which conceptually constitute an extension of the fast bus back plane over several segment (crates).



Acquisition computers:

mini computers - main frames:

Traditionally data acquisition systems were run on quite big and expensive general purpose computers. They offer a comfortable development environment and the necessary services. (disks, printer, tape stations; software packages).

Almost of all of them, however, are **not** real time machines. Therefore in all recent experiments intelligent and fast front end processors have to be used. (VME, FASTBUS). In the last two years, however, both the software services (OS9) and the hardware components are available already in the peripheral/logical bus as with VME; almost invariably at a fraction of the cost of an equivalent piece of hardware on a main frame.

With fast tape drives on SCSI it seems it is no longer necessary to keep these machines in the main experimental data flow. They will still be needed for some time in complex experiments as a central storage device, to provide access to the system to a big number of users. However, with less and less emphasis, on the actual online/real time aspect of data acquisition systems, as microprocessors are intrinsically superior in this field.

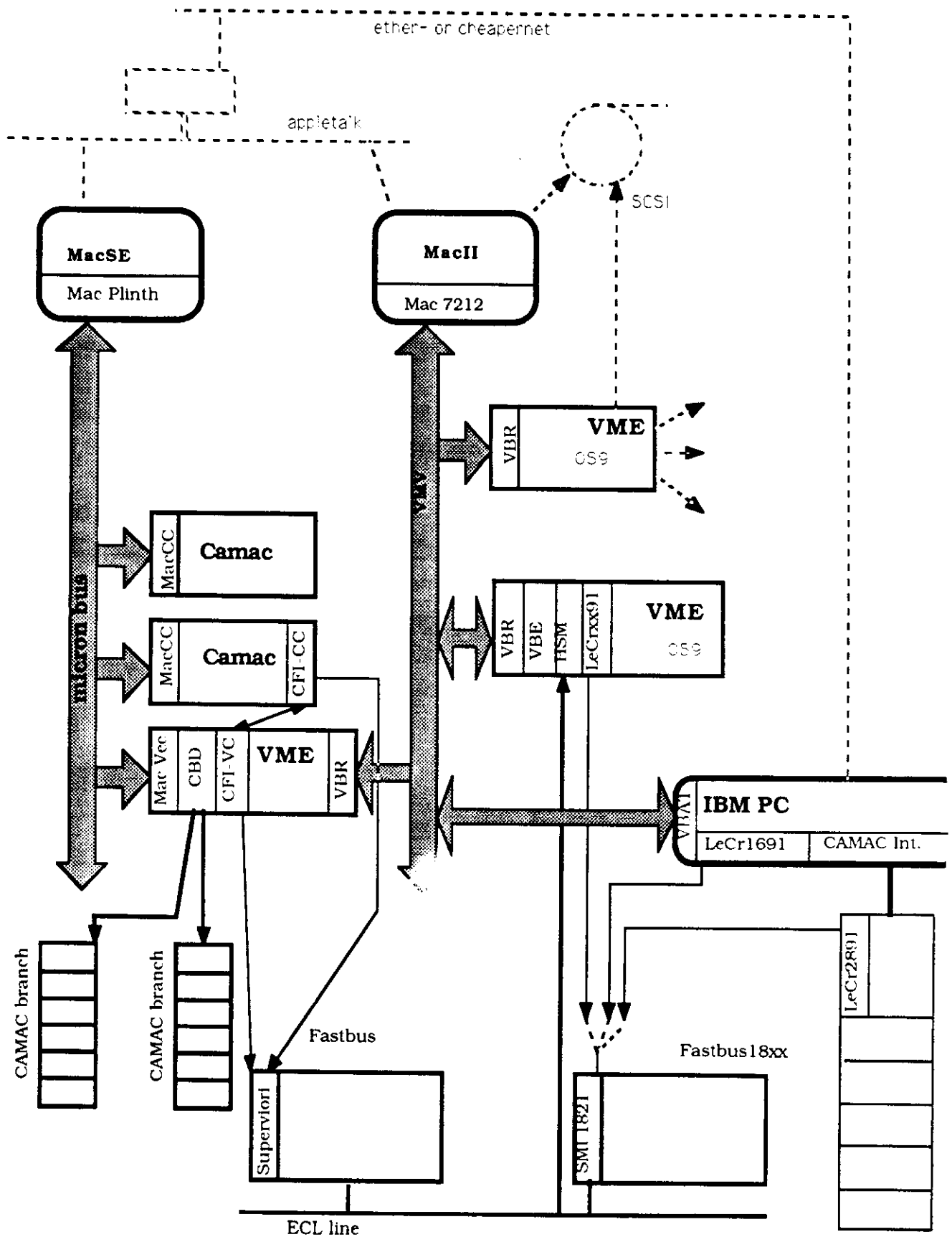
PCs: (IBM PC, Mac, Apollo, Sun ...)

Almost all modern PCs have now an open architecture and hence can be connected to the peripheral busses of interest. They all offer a similar basic computing power which is sufficient for a wide range of experiments. The software environment still is somewhat limited but with UNIX being implemented on the high end of this machines this will not be a problem in the near future. Even by now already a lot of data acquisition specific software exists on these machines. They are very successfully being used in the control of big experiments together with VME and FASTBUS based front end processors(UA1). Due to the huge market volume the necessary standard hardware is usually very cheap and maintenance fees cause a negligible cost.

Entirely bus based systems:

It is possible to control an experiment totally from processors sitting on the peripheral bus. This is especially true for VME where access to almost all the standard hardware of minicomputers is now available(laser printer, tapes, cartridges, disks, networks,)

In VME following solutions were taken: VALET+, OS9, stand alone programs.



Interfaces:

a) computer <-> computer (computer peripherals)

ethernet, (appletalk, LAN, token ring...), SCSI(computer->periphery)

b) transparent(high speed) computer <-> bus interfaces:

	VME	VMV	CAMAC	FASTBUS	18xx
MacSE	1		2		
MacII	3	4	5		
PC (Apollo)		6	7		8
Q-Bus(μ Vax)	9	10	11	12	13
VME(Valet+)	VMV,VSB	VBR,VBE	14,15	12,15	16,17

1..... MacPinth- MacVee	8..... LR 1691	15... CFI-CC CFI-VC
2..... " MacCC	9..... CES and others	16... Superfiori-CFI-CC
3..... micron - MacVee	10..... CES	17.... LRxx91
4..... Mac7212	11.... CES, Kinetics	18.... CES HSM
5..... micron - MacCC	12.... CHI	
6..... VMV-AT	13.... ECL-IO channel	
7..... Kinetics and others	14.... CBD 8210->A2 contr.	

c) Nontransparent interfaces:

RS232, centronics etc.

d) Bus<->Bus interfaces:

	CAMAC	FASTBUS	FB18xx	VME
CAMAC	-	1	2	3,4
FASTBUS		5	5	6,7
FB 18xx			5	8,9
VME				10,11

1	CFI-CC Superfiori	7	CFI
2	LR 2891 - 1821	8	HSM
3	CBD 8210	9	LRxx91
4	CFI	10	VSB,VMV,.....
5	ECL cable segment		
6	CHI		

Storage media:

Following media are used for storage of experimental data:

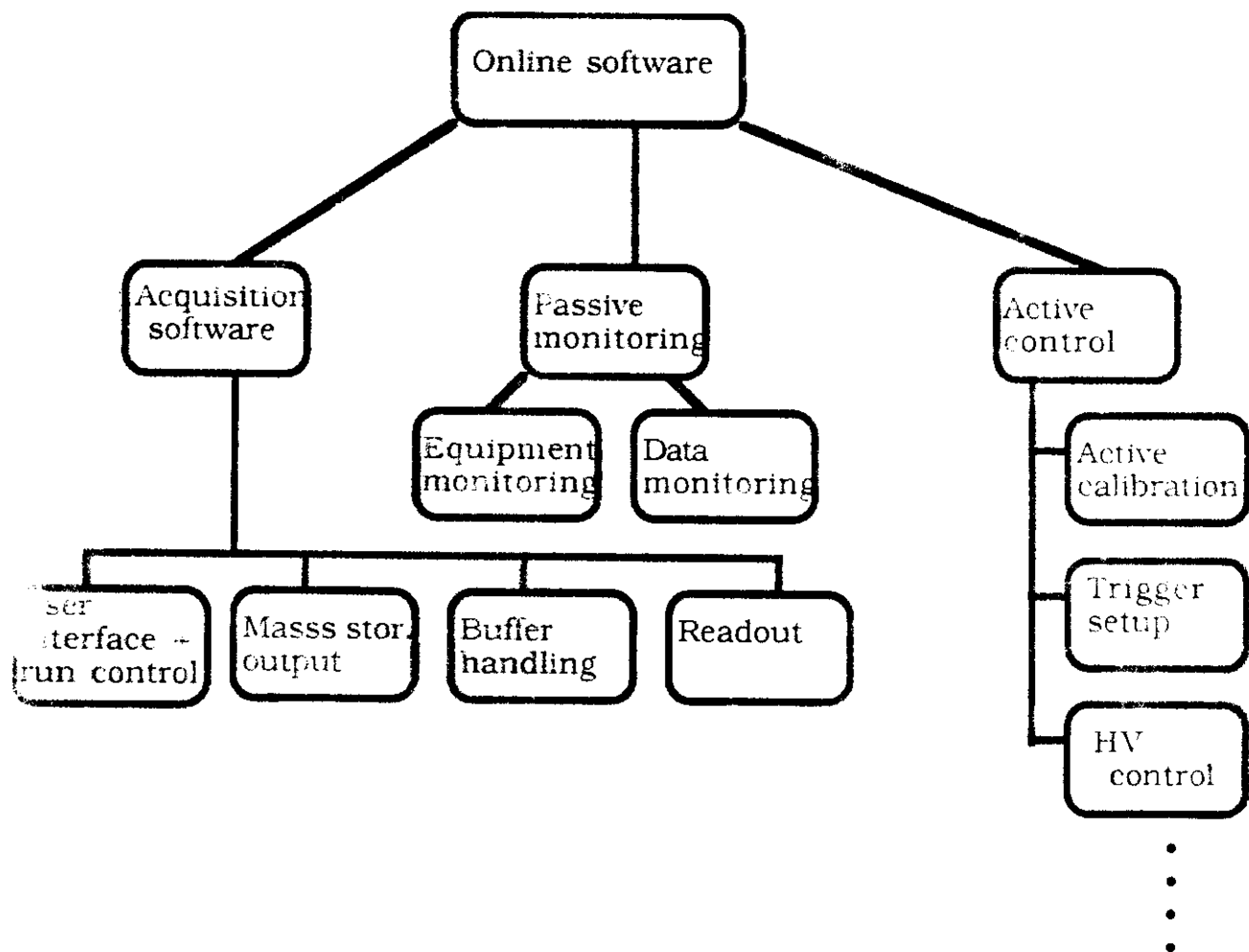
native disk system(or through SCSI)
 diskettes
 streamer cartridge tapes
 rotating head tape stations (video technology)
 9 track tape
 IBM cartridges

	write speed	capacity	data reli.	medium (Sfr/Mbyte)	device
disk	<1 Mb/sec	<100Mbyte	short term	not appl.	500-1K
diskette	30 Kbyte	1 Mbyte	months	3	200-400
streamer	50-100 kB	40 Mbyte	sev. years	0.5	1-2 K
9track tape	150-600 KB	40 or 160MB	10 years +	0.125(0.6)	15-40 K
video	250 Kbyte	2-3 Gigaby.	?	0.01	5 K
IBM cartr.	up 3 Mbyte	200 Mbyte	10 years +	0.1	60 K

Only 9 track tapes are fully portable between mainframes. IBM cartridges are likely to be in the near future. It should be hoped for that VCR type tape support is widened in the future, too. They are very cheap and have outstanding performance figures.

Laser (CD technology) disks have not been mentioned as it seems that they are inferior to video tape technology (speed 50 Kbyte, non erasable, medium price etc.). The direct access possible with laser disks is not very important as tapes will be written and analysed sequentially.

Software aspects



Advanced concepts:**Data reduction:**

Very often only a small part of all your data channels sees an effective signal. Instead of putting all read data words in the event only the ones above a certain threshold are passed on together with their 'address'. By recording only these the output band width can be considerably reduced. For TDCs this operation is trivial, time overflows are simply not recorded. If the pulse height is to be recorded usually a threshold of pedestal + $n \sigma$ is applied.

These methods will introduce a small overhead on the actual read out time, if applied in the read out routine, but significantly reduces the mass storage band width. Both parallelism and pipelineing can be applied.

FADCs give 250-1000 data words per real channel resulting most of the time in unreasonable event sizes. Again only counts above a preset threshold have to be recorded. The pulse samples can also be replaced by their parametrization (integral, centre of gravity=time, peak etc.)

Aside from that standard non-physics oriented reduction methods can be applied (Huffman coding and others) traditionally used in massive volume data transfers (satellites). These can yield surprisingly high reduction factors (>60 %). Especially for imaging detectors many algorithms are likely to be usable from other fields (telephone transmission of TV pictures)

Filtering:

The events gets read out completely and causes the complete read out dead time. Before it gets written to mass storage, however, a complex physics algorithm is run on it. On a positive decision the event is passed on to the output medium else discarded, thus reducing the necessary band width of the mass storage and at the same time the offline computing power.

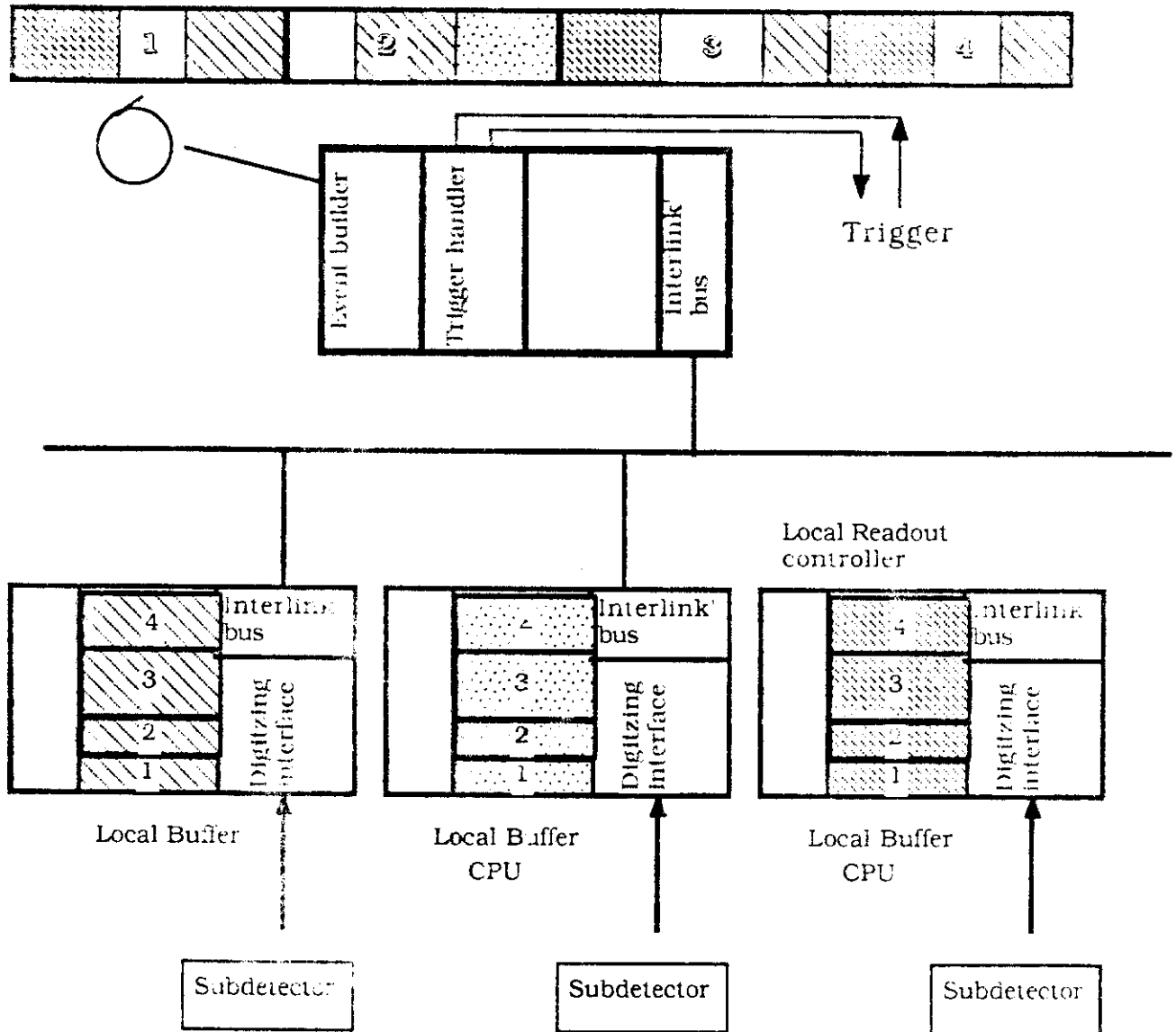
The algorithm has to fulfil 2 requirements simultaneously:

Fast execution time (see pipelineing), high rejection rate (> 50 %)

2-level (software) trigger:

The event is partially read out from the digitizing electronics and from this data subset a trigger decision is deducted. If it was negative, the read out is pre-pre-empted immediately and the trigger system is reset. In this way the effective read out dead time can be reduced. The time constraints are even more stringent as this operation cannot be pipelined (limited parallelism however is possible). The rejection rate should again be higher than 50 %.

Parallelism in read out:



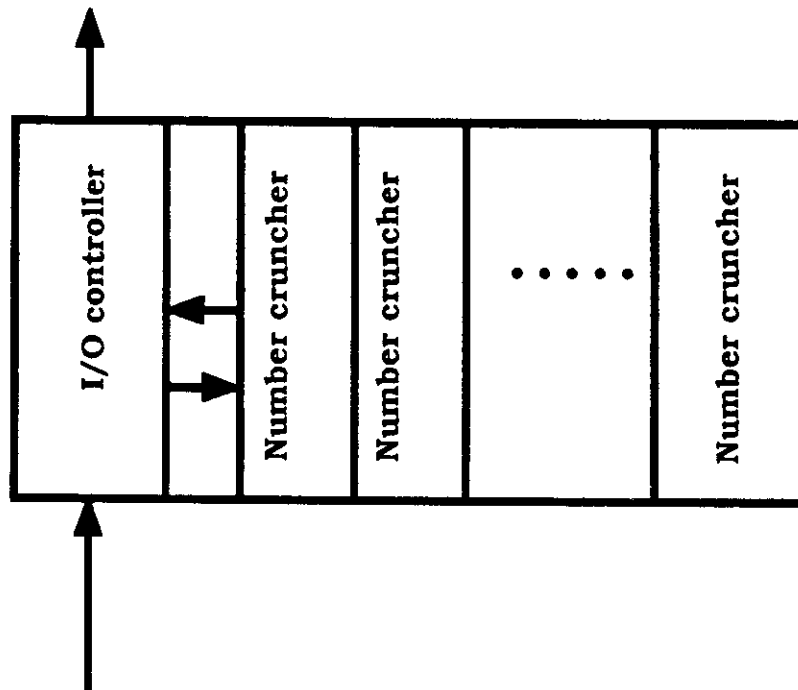
Motivation:

- The exponential time distribution of the event arrival and possible bursts make it sensible to be able to respond to the trigger system at a rate higher than the average sustainable rate.
- Usually the acquisition of data from digitizing busses is quite time consuming as compared to mere block moves on the e.g. VME bus. Therefore given a suitably high degree of parallelism pays off the overhead of intermediate storage of the data and concatenating them once they have been fully acquired.
- Having parallelism allows to apply effective data reduction mechanisms which increase the effective output rate and offload the offline computing budget.

Pipelineing:

This concept can be used in highest level filtering and data reduction processes. Any complexity of algorithms can be achieved by applying an appropriate number of processor. The communication overhead is almost constant however high the number of processors is. The number of processors required is calculated as follows:

$$N = T_{\text{algorithm}} * f_a$$



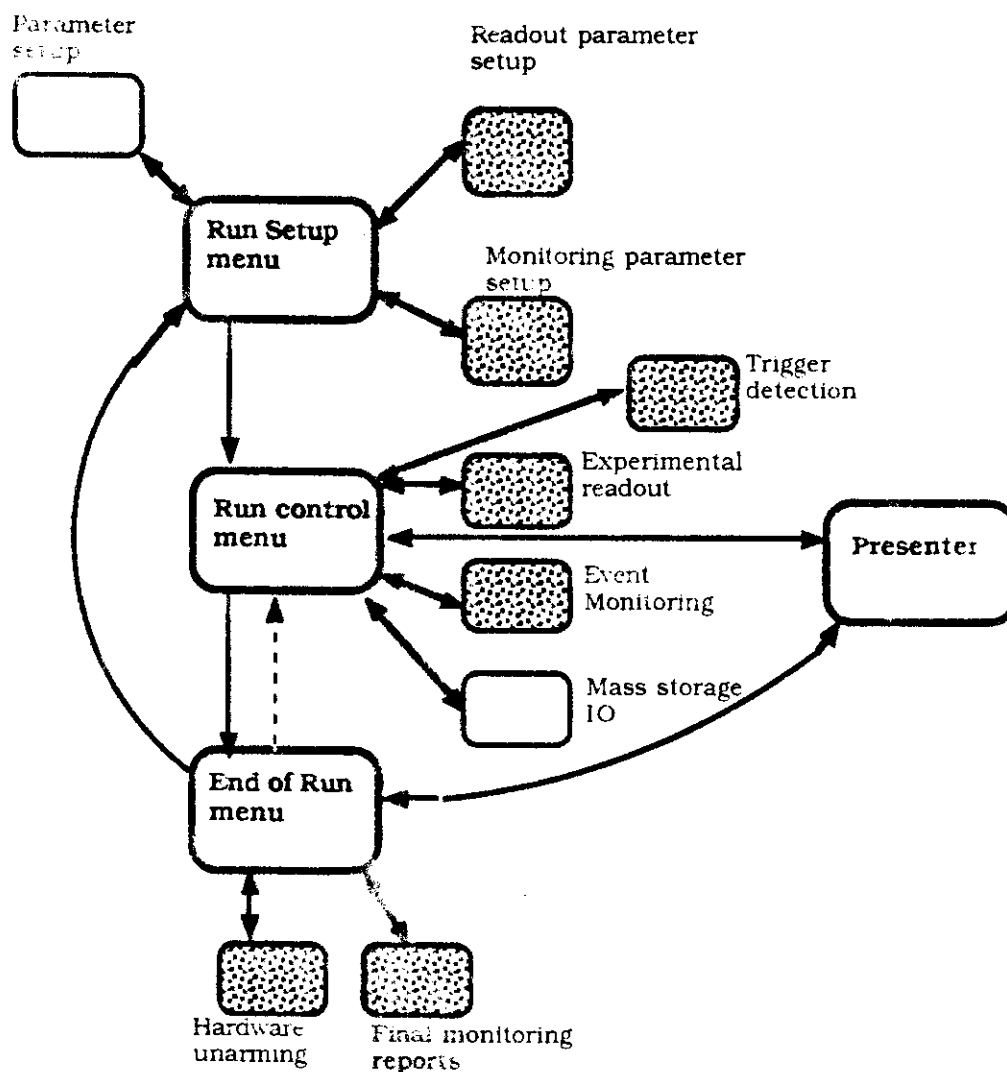
Number crunchers in use:

Farms of: 3081E (168E)

68020 up to 4 on 1 VME board (1 crate = 150 μ VAXes!!!)

ACP on BI bus

Appendix



The SPA³W System

- A modular Macintosh based hardware independant Data Acquisition system
- Initial aim: An inexpensive Data Acquisition System for test beam and small experimental setups. Limited analysis capacity.
- Used as a subsystem in VME based multiprocessor systems.
NA34/3, NA44

Programming environment:

MacSys by UA1

FORTRAN

Standard libraries

VME-specific libraries

DAQ package SPA³W

Start Control Board

Run

327

Date

2/ 2/1989

Time

11:52:14

Input



Camac



Tape



Disk

Output



Tape



Disk



None

Monitoring



Off



On

Number of Events in Monitor

Disk

Events

Max. number of events

999999

User Defined Parameter Value

86

Add comment

Presenter

Quit

Start Run

Messages

02/02/1989 11:52:13

***** DAQ started *****

206 Bytes

5 Lines

Top 1

Cursor 1



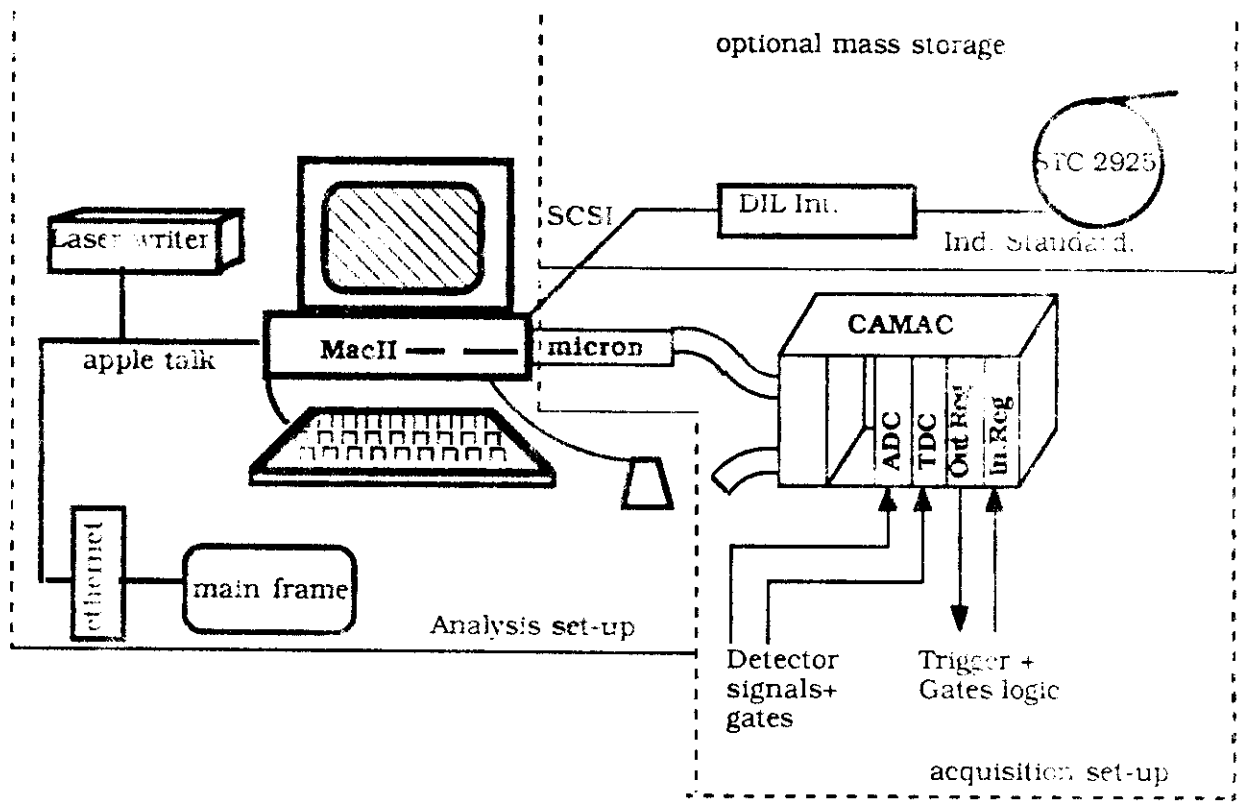


Figure 5. Hardware set-up with MacII, CAMCC, appletalk + ethernet

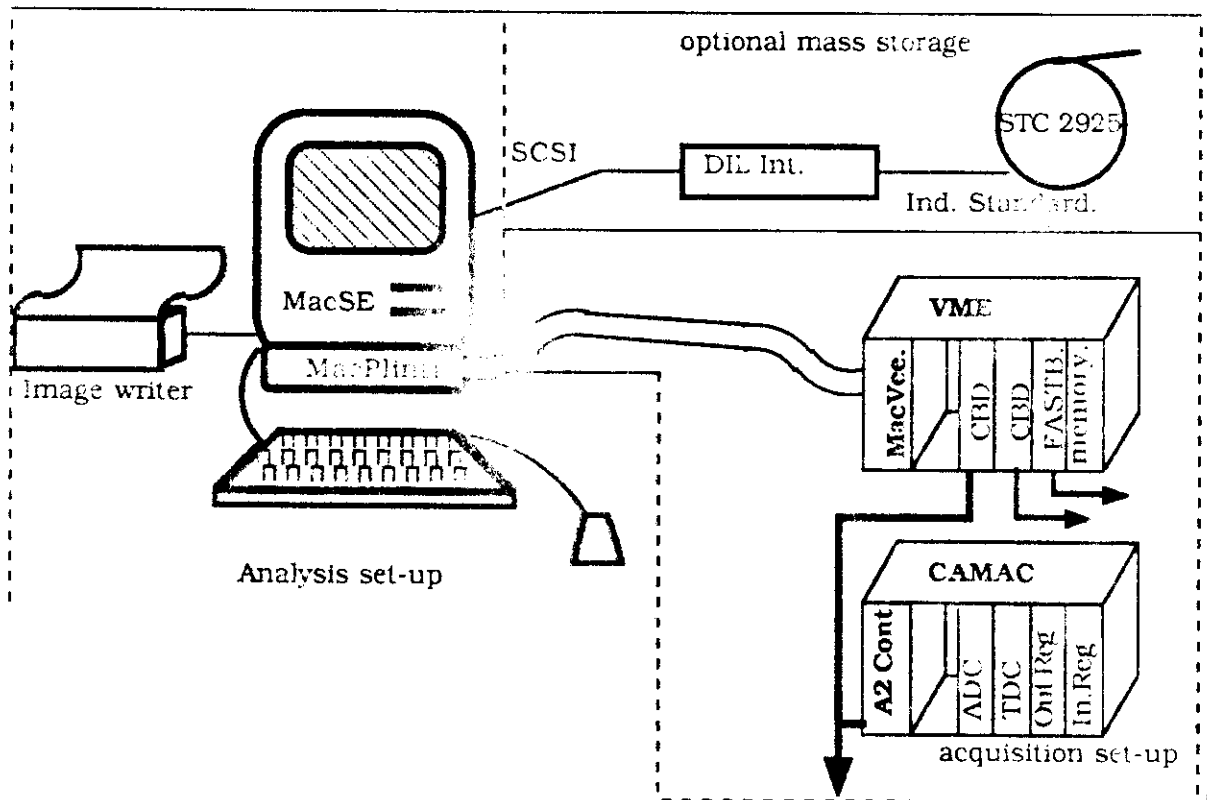


Figure 6. Hardware set-up with MacSE, VME to CAMAC interface, FASTBUS

The VALET + System

- A modular VME based micro-computer system, developed by CERN
- Initial aim: Testing of Electronics
- Recently development of imbedded multiprocessor Data Acquisition Systems.(CP/Lear)

Programming environment:

Monitor primitives: MoniCa

PILS: Basic like language

Standard libraries

VME-specific libraries

DAQ package SPIDER

Cross development software on Vaxes

