European Centre for Medium Range Weather Forecasts

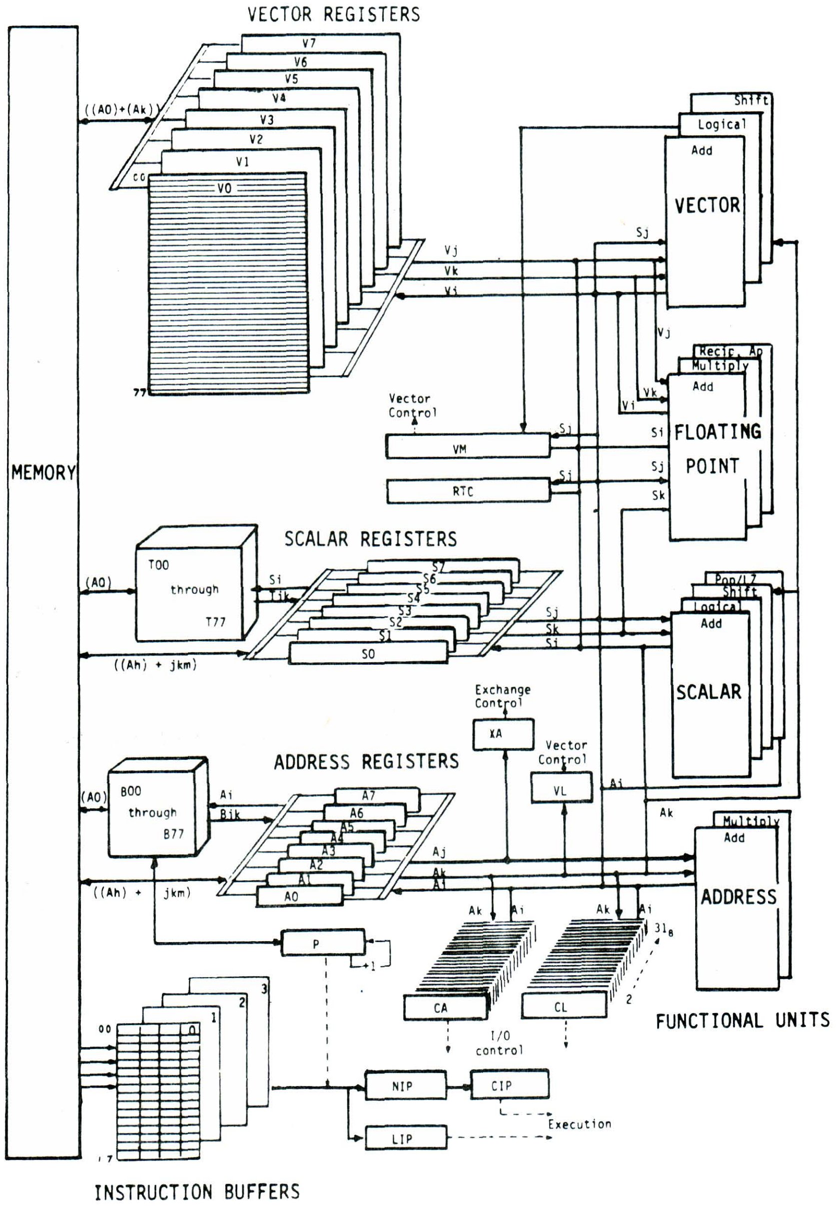
TECHNICAL NEWSLETTER

Shinfield Park, Reading, Berkshire RG2 9AX, England

Reading (0734) 85411 Telex 847908

Number 3 - June 1979

REGISTER STRUCTURE OF THE CRAY-1 CPU



IN THIS ISSUE:

METEOROLOGICAL

COMPUTING

*NOTE: These articles directly concern the computer service, we recommend computer users read them all.

COVER: A block diagram of the register structure of the CRAY-1 CPU. This is figure 1 for the article "The CRAY-1 architecture" on page 10.

An apology: We omitted the date and number from the front cover of the last ECMWF Technical Newsletter. We hope this has not caused too much confusion!

This Newsletter is edited and produced by User Support, for the Operations Department of ECMWF.

The next issue will appear in August.

STATUS OF ECMWF'S METEOROLOGICAL OPERATIONAL SYSTEM

Technical Newsletter No.1 contained a basic description of the planned ECMWF Meteorological operational system (EMOS), and the individual subsystems making up EMOS. The present notes review progress in development and implementation of EMOS.

Several years of intensive planning and work are now culminating in the final steps in the development of EMOS, the integration of each of the individual component subsystems into a complete operational suite. About 90% of all the necessary subsystems are working individually. Integration tests are well advanced, especially for the preprocessing*. About 25 full days of data in Global Telecommunication System format have been processed. Data has been decoded, quality controlled and stored in the Reports Data Base. About 5% of the GTS bulletins are rejected at the decoding level because of format errors which are not at present automatically corrected by the decoding system. Many of the important bulletins are manually corrected from a visual display unit the corrected bulletins are reprocessed to feed the Reports Data Base. Statistics are being gathered regarding the quality and coverage of G.T.S. data by building up statistic files in the Data Base. The Analysis system has been integrated also. However INITIALISATION, FORECAST and POST-PROCESSING remain to be fully integrated, but each of these does already work satisfactorily individually.

Development of the Operational Watch Programs has been going ahead in a most encouraging way and all the basic facilities to monitor the Operational Suite (Display and corrections (if necessary) of Bulletins and Reports, and display/plotting of Data Coverage, Statistics, Tephigrams, contours on various Geographical areas) are working quite satisfactorily.

Optimisation and improvement of some programs is being carried out.

In conclusion, the first phase of EMOS should have been fully completed by July, and a starting date in July for daily medium-range forecasting on a preliminary operational basis is expected to be maintained.

* (See article on EMOS in Newsletter No.1 for the definition of the subsystems).

- Joel Martellet

* * * * * * * * * * * * * * * *

AN INTRODUCTION TO THE ECMWF ANALYSIS AND DATA ASSIMILATION SYSTEM

The objective of ECMWF to which top priority has been given is the operational production of medium range weather forecasts, that is, forecasts up to a period of 10 days ahead. This will be carried out using numerical models which predict the evolution of the atmosphere from a given initial state. Thus an essential pre-requisite for success must be an accurate definition of that initial state, in the way required by the model. The input data for the determination of the initial state is all the observational data made available by the World Weather Watch system from a variety of observing systems; the extent and accuracy of this observational data determines the accuracy of the specification of the initial state. The ECMWF analysis and data assimilation system has been built for the purpose, inter alia, of using the available data which are essentially randomly distributed in as effective a way as possible and defining the required initial atmospheric state of winds, temperatures, geopotential heights and humidity on a regular grid to be used as the basis of the forecast.

The scheme used at ECMWF is based upon a six hours analysis-forecast cycle; the overall system is illustrated in Figure 1.

The principal steps are the Analysis, the Initialisation and the 6-hour Forecast. The two "transformation" stages ("pressure to sigma" and "sigma to pressure") are necessary because the analysis is performed in a co-ordinate system (pressure) that is different from that used by the initialisation and the forecast (sigma =pressure normalised with respect to surface pressure).

Every six hours, all the available observational data gathered, checked and quality controlled for a six-hour period (i.e. for three hours either side of the hour being analysed) are used to "correct" a set of First Guess Fields (the forecast from the previous cycle, see Figure 1) so as to produce the Analysis Fields - this is the "Analysis" step.

These fields usually contain, as well as much useful information on the real atmospheric evolution in the past 6 hours, small scale imbalances between wind, temperature and geopotential that we believe are more harmful than useful to the forecast model; we therefore proceed to remove these imbalances, trying to leave unaltered the mainstream information; this is the "Initialisation" step.

The medium-range forecast can now start from these "initialised fields". In order, however, to be able to carry out the next analysis, six hours later, we run a special six-hours forecast that will provide us with First Guess Fields for the next cycle - this is the "Forecast" step.

This whole procedure will be carried out four times a day, thus progressively "assimilating" all the observations taken during the course of the day. It is planned that the main daily operational 10-day forecast will be run starting from one of the four sets of initialised fields produced during the day (probably either the 12 or 18 Z fields).

We now examine in more detail some of the principles of, and methods used during, the Analysis step. For convenience, the Analysis is considered in two parts: the wind and geopotential analysis (often referred to as "mass and wind" analysis) and the heterogeneous set of "secondary" analyses (humidity, sea surface temperature, etc.).

Our method of analysis is based on the following basically simple procedure. Given one of the points of the required rectangular grid at a certain latitude, longitude and height, all the relevant observations (i.e. containing useful information) within a certain three-dimensional "radius of influence" are selected. In order to analyse, say, wind we use also the information implicit in the neighbouring observations of pressure and temperature, and not only the wind observation; this is termed a "multi-variate" method, expressing that more than one variable is used in the analysis of a single variable.

The First Guess Fields, known at all required regular grid-points, are interpolated to all the observation points and subtracted from the observed values, providing us with a "correction" value at every observation point. The final grid-point analysed value is the sum of the first-guess value plus a "grid-point correction" which is just a weighted average of all the "observation point corrections" that fall within the radius of influence of that grid-point.

The complications arise in computing the various weights that have to be attributed to each single observation "correction" for the analysis value at each single grid-point. These weights can be computed simply by taking into account the estimated observation error (compared to the first-guess error) and the distance between the observation and the grid-point (as is done in the secondary analyses, humidity, sea surface temperature, etc); this is usually called the "correction method". The weights can, however, take into account the statistical (and dynamical) relationships that, to the best of our knowledge, exist between the various meteorological parameters (pressure, temperature, wind, etc.) so as to make the analysis statistically "optimal" (and dynamically consistent) - in other words, to minimise the mean analysis error. This second method, called "optimum interpolation", although far more expensive in terms of computer usage, gives better results, and is therefore used for the analysis of the mass and wind fields, for which the meteorological statistical (and dynamical) relationships are, at least, partially known to us.

This description should shed some light onto the reasons why the ECMWF Assimilation Cycle is sometimes described as an "Intermittent, six-hours, forward scheme"!!! To tell the truth, things are even more complicated.......

- S. Tibaldi

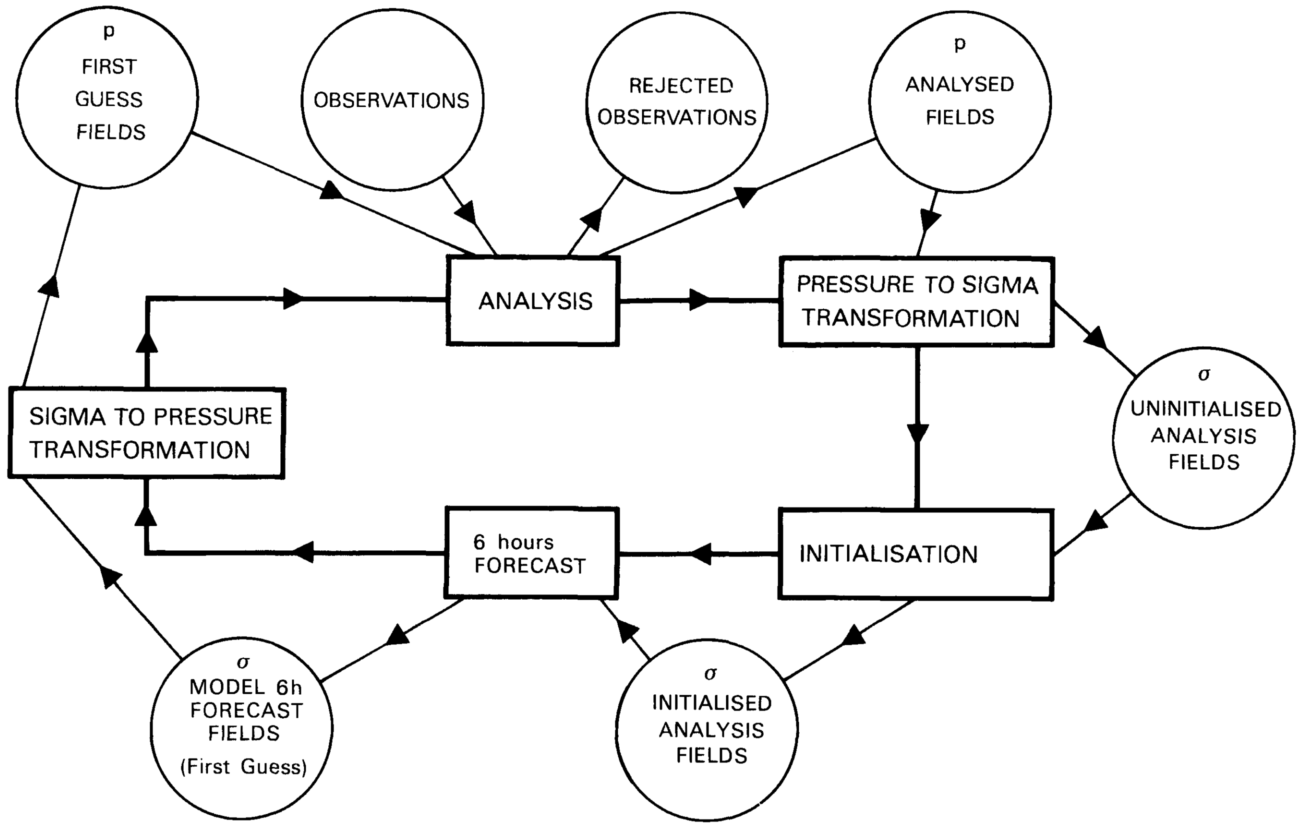


Fig. 1 Interrelation of the main functional components (square boxes) and data file (round boxes) in the ECMWF assimilation cycle. p ≡ pressure file σ ≡ sigma file

* SPECIAL SESSIONS

At the last meeting of the Computer Steering Group, new schedules for special computer sessions were discussed and agreed.

Under the new schedules, 3 types of special session will exist.

1. Dedicated Session. This type of session will require use of the Cyber and/or Cray in a dedicated manner and no service will be available to users. All dedicated sessions will take place before 0900 or after 1800 on Mondays to Fridays, or at anytime on Saturdays and Sundays, except one session from 0900 - 1200 each Wednesday morning.

2. NFEP Testing. Some NFEP testing will take place during dedicated sessions. However, soon we expect to be in a situation where it is possible to provide a full service except Intercom whilst testing the NFEP. These special test sessions are scheduled to take place between 1200 and 1300 on Mondays, Tuesdays, Thursdays and Fridays. During these sessions no Intercom facilities will be available.

3. Operational Suite Integration. During the sessions when the operational suite is being tested, that suite will have top priority in the machine. Other Cyber and Cray users will then have a reduced turnaround and there may be some restriction of private pack and tape mounts. However, the full service facilities will be available during these sessions albeit at a reduced level. These sessions are scheduled from 1400 to 2300 on Mondays, Tuesdays, Thursdays and Fridays during May, and every evening and overnight in June.

Details of actual timings of these sessions have been circulated to division heads in Operations and Research, and displayed on the notice board in the User Area (by room 022).

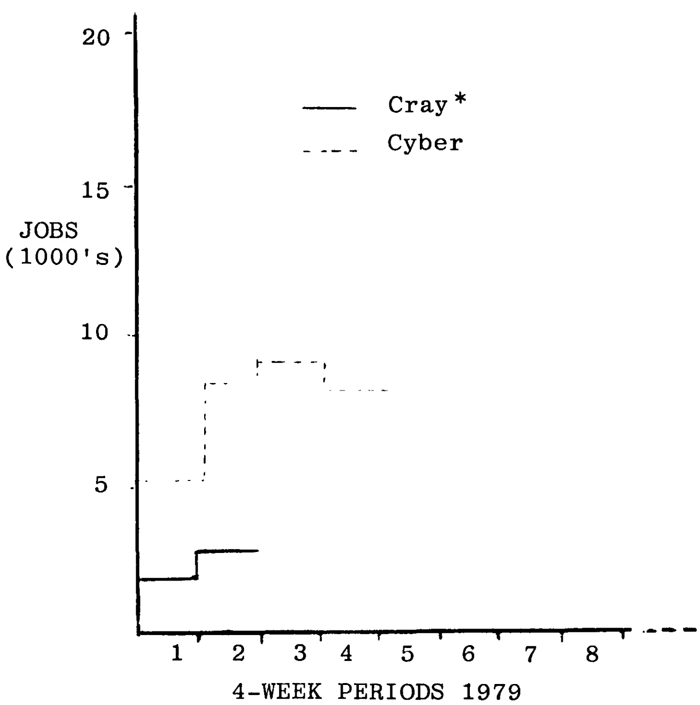
- Peter Gray

* * * * * * * * * * * * * * * * * * *

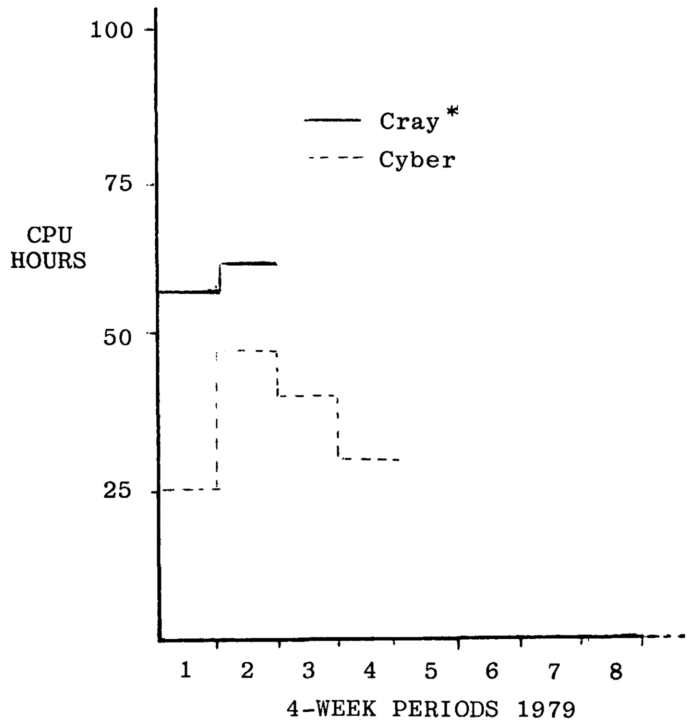
STATISTICS

The tables below show the weekly average for the number of jobs and CP time used for both systems. They are presented as averages over 4-week periods, to smooth out random week by week variations.

Average number of jobs per week within each period



Average CPU hours used per week within each period



* CRAY statistics for periods 3 and 4 are not yet available

NEW FEATURES IN COS 1.0.5

The next release of COS (and CFT- see next article) has been received at the Centre. It is known as level 1.0.5. We hope to install it sometime in the next 2 months.

i. Job RERUN/NORERUN

The restarting of a job from its beginning is referred to as "rerunning a job". Conditions causing the system to attempt to rerun a job are :

- operator RERUN command
- uncorrectable memory error
- uncorrectable mass storage error on the roll-file
- system restart.

Certain operations performed by the job may make it non-rerunnable such as :

- writing to a permanent dataset
- saving, deleting, adjusting or modifying a permanent dataset.

Ordinarily any of these functions would make the job non-rerunnable. The program may, as a user option, declare the job to be rerunnable or not rerunnable thus overriding the system. As released, the system never makes a job rerunnable automatically, however this may be altered on site to make the default RERUN.

The control statements and macros involved are :

$$\text{NORERUN} \begin{Bmatrix} \text{ENABLE} \\ \text{,DISABLE} \end{Bmatrix} . \quad \text{and} \quad \text{RERUN} \begin{Bmatrix} \text{ENABLE} \\ \text{,DISABLE} \end{Bmatrix} .$$

(CFT-callable subroutines will be produced also)

NORERUN, ENABLE	tells the system to monitor the job to declare it non-rerunnable if any non-rerunnable functions are performed.
NORERUN, DISABLE	tells the system to stop monitoring the job for non-rerunnable functions, it does not make an already non-rerunnable job rerunnable again.
RERUN, ENABLE	instructs the system to consider the job to be rerunnable, regardless of what functions have been executed previously.
RERUN, DISABLE	instructs the system to consider the job to be not rerunnable, regardless of what functions have been executed previously.

The RERUN statement/macro in no way affects the monitoring of the user job for non-rerunnable functions, this is done by the NORERUN statement/macro.

ii. Memory-resident datasets

By specifying a parameter 'MR' on the ASSIGN statement the job may define a dataset to be memory-resident. This means that the system I/O routines will write the buffers to disc only if they become full. One consequence of this is that a memory-resident dataset cannot be SAVED. At the time of going to press this feature has not been tested at ECMWF and I would therefore advise against its use until it has been evaluated, since it appears to pose several possible problems.

iii. New DUMP parameters

The new parameter 'FORMAT=f' allows the DUMP to be displayed in various formats, the options are:

O Octal integer amd ASCII character as at present

D Decimal integer and ASCII character

X Hexadecimal integer and ASCII character

G Floating point/exponential and ASCII character

P 16-bit parcels.

The new parameter 'CENTER' causes a dump of 64 words on each side of the address held in the p-register of the exchange package. The format for this is P.

iv. New AUDIT parameter

The "X=mm/dd/yy {:'hh:mm:ss'}" parameter lists all datasets that have expired as of the specified date and time. The date may be specified alone. The default date and time are "now" if 'X' alone is specified,

e.g. AUDIT (X=02/21/79: '01:30:00')

AUDIT (X=03/18/79)

AUDIT (X)

\- Neil Storer

* * * * * * * * * * *

CFT 1.0.5 and DEBUG 1.0.5

With the release of the new COS (1.0.5) a new CFT will also be released. This has available a "MANTRAP-like" package called DEBUG. Before the system is generally released these are available under TEST and may be used as described below. Note that the new $FTLIB must be used with the new CFT. Any user who does I/O from CAL should contact either User Support or myself before using the new $FTLIB.

```
ACCESS (DN=CFT,PDN=CFTTEST)
ACCESS (DN=$FTLIB,PDN=$FTLIBTEST)
CFT (ON=Z)
LDR.
EXIT.
DUMPJOB.
ACCESS (DN=DEBUG,PDN=DEBUGTEST)
DEBUG (TRACE,BLKS)
```

If the call to load the program and execute the program are in different jobs (i.e. LDR(BX,AB=....)) then the dataset $DEBUG should be saved in the first job and accessed in the second.

Users of the following public libraries - ECLIB, CONVLIB - must use special test versions when trying CFT(1.0.5). They are ECLIBTEST, CONVLIBTEST respectively. Note that ACCESS cards will be required.

New CFT

This is a short description of the new features of CFT 1.0.5, for more details contact User Support or myself, or see the new CFT reference manual (to be distributed later).

Features marked '**' are in the new ANSI-77 Fortran standard, they will also be supported by a new Fortran compiler due to be released by CDC later this year.

Support of low bound in array dimensions

Lower bounds can be specified in array dimensions e.g.

DIMENSION IARR (0:10, -1:5)

is a two dimensional array consisting of 11 x 7 elements.

Assumed size array declarator

The last bound of an array declaration in a subroutine of an array passed as an argument may be a * to indicate that the size of the array passed is used, e.g.

```
SUBROUTINE SUB (X,Y)
DIMENSION X(*), Y(10, 0:2,*)
```

NAMELIST support

NAMELIST I/O is supported via a NAMELIST statement and new options on READ, WRITE, PRINT and PUNCH statements. In general, Namelist processing is the same as on the Cyber.

The form of the NAMELIST statement is:

NAMELIST/$groupname_1$/var_1,var_2,var_3.../$groupname_2$/var,.....

Groupname can only be used as in the following I/O statements.

```
READ(unit,group name[,ERR=sn, END=sn])
WRITE(unit,group name[,ERR=sn]
READ  group name
PRINT group name
PUNCH group name
```

[Note that in the new CFT manual there is a misprint which shows an extra ',' in the last 3 of these statements]

The input for namelist can consist of an echo flag, comments, and namelist groups and values. An echo flag is an E in the 1st column of any input record. If an echo flag occurs all records read are echoed to $OUT. If a namelist group is found on input that is not being looked for it is skipped and a message put on in the LOGFILE.

Namelist output consists of an "&", the group name and the variable names. Note that the default output separator (&) is not compatible for Cyber input. A control subroutine can be called however to change this, see the example program.

Example: Namelist Program, Input and Output.

Program.

```
CRAY FORTRAN COMPILER VERSION 1.05 03/29/79
COMPILATION DATE AND TIME          05/16/79  -  16:36:35

      PROGRAM TESTNL
      LOGICAL ALLDONE
      NAMELIST/INPUT/A,B,C,ALLDONE
      NAMELIST/OUTPUT/RESULT,A,B,C
      DATA A,B,C,ALLDONE  /1.0,1.0,1.0,.FALSE./
      CALL WNLDELM (1HS)-this causes the output separator to be a $
   10 READ INPUT
      IF(ALLDONE) STOP
      RESULT = A*B*C
      PRINT OUTPUT
      GO TO 10
      END
```

Input

```
 A COMMENT CARD

$INPUT; THIS IS A COMMENT - USE DEFAULT VALUES$
$INPUT A=2,;THIS IS ALSO A COMMENT - KEEP C THE SAME
 B=3$
$INPUT A=4,B=5,C=6.0$

$NOTTHERE A=4,B=7$
$INPUT A=-1.5,B=-2.5,C=-3.5$
$INPUT ALLDONE=.TRUE.;THATS THE LOT $
$INPUT B=-1; THIS SHOULD NOT BE READ$
```

Output

```
$OUTPUT RESULT = 1., A=1., B=1., C=1., $END
$OUTPUT RESULT = 120., A=4., B=5., C=6., $END
$OUTPUT RESULT = -13.125, A=-1.5, B=-2.5, C=-3.5, $END
```

Alternate returns from subroutines

To use alternate returns the call statement must contain the return statement number in the form *sn e.g.

```
      CALL SUB (A,B,*30,*20)
20    ...
30    ...
```

The subroutine statement must contain a * whenever a label is expected.

e.g. SUBROUTINE SUB (A,B,*,*)

To return to an alternative return the format of the return statement is RETURN n where n is the number of the * in the subroutine statement to return to:

e.g. with the above statements

RETURN 2
Would go to statement 20.

External vector functions

If a CAL user function exists which can handle vector arguments it may be declared as such with a VFUNCTION statement. This will then allow loop vectorisation with external references.

New compiler option P

This controls use of double precision. Specifying OFF=P on the CFT control card will convert all double precision items (constants, variables, subroutines and format elements) to single precision.

- Gary L. Harding

* * * * * * * * * * *

* DISPOSE - WAIT BECOMES THE DEFAULT

Some changes to the COS operating system have been made and tested, and were incorporated into the production system on Monday 21 May.

The main change that affects users is that 'WAIT' on DISPOSE has been made the default. If you do not wish to wait you must code 'NOWAIT' on the control card.

e.g. To WAIT code:
DISPOSE(DN=X,ID=XYZ,DC=ST) or DISPOSE(DN=X,ID=XYZ,DC=ST,WAIT)

Not to wait code:
DISPOSE(DN=X,ID=XYZ,DC=ST,NOWAIT)

For those users who internally dispose datasets there is a change to the PDD macro to enable them to specify WAIT or NOWAIT. This is the parameter WAIT=(ON)
(OFF).

'ON' means WAIT & is the default 'OFF' means NOWAIT
'WAIT=ON' sets a bit 'PMWAIT' in the PDD

'WAIT=OFF' unsets this bit.
e.g. to WAIT code :
e.g. X PDD DN=X, ID=XYZ,DC=ST,WAIT=ON
or X PDD DN=X, ID=XYZ,DC=ST

not to WAIT code
X PDD DN=X, ID=XYZ,DC=ST,WAIT=OFF

- Neil Storer

* * * * * * * * * * * * * * * *

LIBRARIES ON THE CRAY-1

The following libraries are being maintained on Cray-1 disks so that users need take no action to ACQUIRE them.

NAGLIB
NCARLIB
CRAYLIB
CONVLIB
ECLIB
ECMWFLIB,ID=DUMPOO

ECLIB and ECMWFLIB have identical contents. ECMWFLIB,ID=DUMPOO will be retained for a limited period to give users time to change to ECLIB.

Note that it is no longer necessary to ACCESS these libraries (with the exception of ECMWFLIB). It is sufficient to list the library names on the LDR statement.

EX :

LDR(LIB=ECLIB:NAGLIB,...)
causes ECLIB and NAGLIB to be searched for unresolved references.

Please remove ACCESS statement for ECMWFLIB as soon as possible and add ECLIB to the LDR statement.

- David Dent

* * * * * * * * * * * * * * * *

* Cray Station Bug

A problem has been uncovered in the station which results in a jobfile being incorrectly formatted for the Cray. It occurs when an EOF control word is the first word of a block. The effect of the error is to lose the end of file mark and hence 2 job sectors become merged together. A version of the station with this problem fixed will be released within a few weeks. In the meantime, if you suffer from this symptom, try adding a comment control card to your job deck and the error should go away.

- David Dent

* * * * * * * * * * * * * * * *

THE CRAY-1 ARCHITECTURE

This is the second article describing the CRAY-1 architecture in some detail. It has been adapted from "The CRAY-1 Computer System" by Richard M. Russell (Cray Research Inc) published in the Communications of the ACM (January 1978)†

- David Dent

Functional Units

There are 12 functional units, organized in four groups: address, scalar, vector, and floating point. Each functional unit is pipelined into single clock segments. Functional unit time is shown in Table 1. Note that all of the functional units can operate concurrently so that in addition to the benefits of pipelining (each functional unit can be driven at a result rate of 1 per clock period) we also have a parallelism across the units too. Note the absence of a divide unit in the CRAY-1. In order to have a completely segmented divide operation the CRAY-1 performs floating-point division by the method of reciprocal approximation. This technique has been used before (e.g. IBM System/360 Model 91).

Registers

Figure 1 shows the CRAY-1 registers in relationship to the functional units, instruction buffers, i/o channel control registers, and memory. The basic set of programmable registers are as follows:

8 24-bit address (A) registers
64 24-bit address-save (B) registers
8 64-bit scalar (S) registers
64 64-bit scalar-save (T) registers
8 64-word (4096-bit) vector (V) registers

Expressed in 8-bit bytes rather than 64-bit words, that's a total of 4,888 bytes of high-speed (6ns) register storage.

The functional units take input operands from and store result operands only to A, S, and V registers. Thus the large amount of register storage is a crucial factor in the CRAY-1's architecture. Chaining could not take place if vector register space were not available for the storage of final or intermediate results. The B and T registers greatly assist scalar performance. Temporary scalar values can be stored from and reloaded to the A and S register in two clock periods. Figure 1 shows the CRAY-1's register paths in detail. The speed of the CFT Fortran IV compiler would be seriously impaired if it were unable to keep the many Pass 1 and Pass 2 tables it needs in register space. Without the register storage provided by the B, T, and V registers, the CRAY-1's bandwidth of only 80 million words/second would be a serious impediment to performance.

Instruction Formats

Instructions are expressed in either one or two 16-bit parcels. Below is the general form of a CRAY-1 instruction. Two-parcel instructions may overlap memory-word boundaries, as follows:

Fields	g	h	i	j	k	m
	0-3	4-6	7-9	10-12	13-15	16-31
Bit positions	(4)	(3)	(3)	(3)	(3)	(16)
			Parcel 1			Parcel 2

The computation section processes instructions at a maximum rate of one parcel per clock period.

Table 1. CRAY-1 functional units

	Register usage	Functional unit time (clock periods)
Address function units		
address add unit	A	2
address multiply unit	A	6
Scalar functional units		
scalar add unit	S	3
scalar shift unit	S	2 or 3 if double-word shift
scalar logical unit	S	1
population/leading zero count unit	S	3
Vector functional units		
vector add unit	V	3
vector shift unit	V	4
vector logical unit	V	2
Floating-point functional units		
floating-point add unit	S and V	6
floating-point multiply unit	S and V	7
reciprocal approximation unit	S and V	14

For arithmetic and logical instructions, a 7-bit operation code (gh) is followed by three 3-bit register designators. The first field, i, designates the result register. The j and k fields designate the two operand registers or are combined to designate a B or T register.

The shift and mask instructions consist of a 7-bit operation code (gh) followed by a 3-bit i field and a 6-bit jk field. The i field designates the operand register. The jk combined field specifies a shift or mask count.

Immediate operand, read and store memory, and branch instructions require the two-parcel instruction word format. The immediate operand and the read and store memory instructions combine the j,k, and m fields to define a 22-bit quantity or memory address. In addition, the read and store memory instructions use the h field to specify an operating register for indexing. The branch instructions combine the i,j,k, and m fields into a 24-bit memory address field. This allows branching to any one of the four parcel positions in any 64-bit word, whether in memory or in an instruction buffer.

Operating Registers

Five types of registers - three primary (A,S, and V) and two intermediate (B and T) - are provided in the CRAY-1.

A registers - eight 24-bit A registers serve a variety of applications. They are primarily used as address registers for memory references and as index registers, but also are used to provide values for shift counts, loop control, and channel i/o operations. In address applications, they are used to index the base address for scalar memory references and for providing both a base address and an index address for vector memory references.

The 24-bit integer functional units modify values (such as program addresses) by adding, subtracting, and multiplying A register quantities. The results of these operations are returned to A registers.

Data can be transferred directly from memory to A registers or can be placed in B registers as an intermediate step. This allows buffering of the data between A registers and memory. Data can also be transferred between A and S registers and from an A register to the vector length register. The eight A registers are individually designated by the symbols A0,A1,A2,A3,A4,A5,A6, and A7.

B registers - there are sixty-four 24-bit B registers, which are used as auxiliary storage for the A registers. The transfer of an operand between an A and a B register requires only one clock period. Typically, B registers contain addresses and counters that are referenced over a longer period than would permit their being retained in A registers. A block of data in B registers may be transferred to or from memory at the rate of one clock period per register. Thus, it is feasible to store the contents of these registers in memory prior to calling a subroutine requiring their use. The sixty-four B registers are individually designated by the symbols B0,B1,B2,...., and $B77_8$.

S registers - eight 64-bit S registers are the principle data handling registers for scalar operations. The S registers serve as both source and destination registers for scalar arithmetic and logical instructions. Scalar quantities involved in vector operations are held in S registers. Logical, shift, fixed-point, and floating-point operations may be performed on S register data, The eight S registers are individually designated by the symbols S0,S1,S2,S3,S4,S5,S6, and S7.

T registers - sixty four 64-bit T registers are used as auxiliary storage for the S registers. The transfer of an operand between S and T registers requires one clock period. Typically, T registers contain operands that are referenced over a longer period than would permit their being retained in S registers. T registers allow intermediate results of complex computations to be held in intermediate access storage rather than in memory. A block of data in T registers may be transferred to or from memory at the rate of one word per clock period. The sixty-four T registers are individually designated by the symbols T0,T1,T2,..., and $T77_8$.

V registers - eight 64-element V registers provide operands to and receive results from the functional units at a one clock period rate. Each element of a V register holds a 64-bit quantity. When associated data is grouped into successive elements of a V register, the register may be considered to contain a vector. Examples of vector quantities are rows and columns of a matrix, or similarly related elements of a table. Computational efficiency is achieved by processing each element of the vector identically. Vector merge and test instructions are provided in the CRAY-1 to allow operations to be performed on individual elements designated by the content of the vector mask (VM) register. The number of vector register elements to be processed is contained in the vector length (VL) register. The eight V registers are individually designated by the symbols V0,V1,V2,V3,V4,V5,V6, and V7.

Supporting Registers

The CPU contains a variety of additional registers that support the control of program execution. These are vector length (VL) and vector mask (VM) registers, the program counter (P), the base address (BA) and the limit address (LA) registers, the exchange address (XA) register, the flag (F) register, and the mode (M) register.

VL register - the 64-bit vector mask (VM) register controls vector element designation in vector merge and test instructions. Each bit of the VM register corresponds to a vector register element. In the vector test instruction, the VM register content is defined by testing each element of a V register for a specific condition.

P register - the 24-bit P register specifies the memory register parcel address of the current program instruction. The high order 22 bits specify a memory address and the low order two bits indicate a parcel number. This parcel address is advanced by one as each instruction parcel in a nonbranching sequence is executed and is replaced whenever program branching occurs.

BA registers - the 18-bit base address (BA) register contains the upper 18 bits of a 22-bit memory address. The lower four bits of this address are considered zeros. Just prior to initial or continued execution of a program, a process known as the "exchange sequence" stores into the BA register the upper 18 bits of the lowest memory address to be referenced during program execution. As the program executes, the address portion of each instruction referencing memory has its content added to that of the BA register. The sum then serves as the absolute address used for the memory reference and ensures that memory addresses lower than the contents of the BA register are not accessed. Programs must, therefore, have all instructions referencing memory do so with their address portions containing relative addresses. This process supports program loading and memory protection operations and does not, in producing an absolute address, affect the content of the instruction buffer, BA, or memory.

LA register - the 18-bit limit address (LA) register contains the upper 18 bits of a 22-bit memory address. The lower 4 bits of this address are considered zeros. Just prior to initial or continued execution of a program, the "exchange sequence" process stores into the LA register the upper 18 bits of that absolute address one greater than allowed to be referenced by the program. When program execution begins, each instruction referencing a memory location has the absolute address for that reference (determined by summing its address portion with the BA register contents) checked against the LA register content. If the absolute address equals or exceeds the LA register content, an out-of-range error condition is flagged and program execution terminates. This process supports the memory protection operation.

XA register - the 8-bit exchange address (XA) register contains the upper eight bits of a 12-bit memory address. The lower four bits of the address are considered zeros. Because only twelve bits are used, with the lower four bits always being zeros, exchange addresses can reference only every 16th memory address beginning with address 0000 and concluding with address 4080. Each of these addresses designates the first word of a 16-word set. Thus, 256 sets (of 16 memory words each) can be specified. Prior to initiation or continuation of a program's execution, the XA register contains the first memory address of a particular 16-word set or exchange package. The exchange package contains certain operating and support registers' contents as required for operations following an interrupt. The XA register supports the exchange sequence operation and the contents of XA are stored in an exchange package whenever an exchange sequence occurs.

F register - the 9-bit F register contains flags that, whenever set, indicate interrupt conditions causing initiation of an exchange sequence. The interrupt conditions are: normal exit, error exit, i/o interrupt, uncorrected memory error, program range error, operand range error, floating point overflow, real-time clock interrupt, and console interrupt.

M register - the M (mode) register is a three-bit register that contains part of the exchange package for a currently active program. The three bits are selectively set during an exchange sequence. Bit 37, the floating-point error mode flag, can be set or cleared during the execution interval for a program through use of the 0021 and 0022 instructions. The other two bits (bits 38 and 39) are not altered during the execution interval for the exchange package and can only be altered when the exchange package is inactive in storage. Bits are assigned as follows in word two of the exchange package.

Bit 37 - Floating-point error mode flag. When this bit is set, interrupts on floating-point errors are enabled.

Bit 38 - Uncorrectable memory error mode flag. When this bit is set, interrupts on uncorrectable memory parity errors are enabled.

Bit 39 - Monitor mode flag. When this bit is set, all interrupts other than parity errors are inhibited.

Integer Arithmetic

All integer arithmetic is performed in 24-bit or 64-bit 2's complement form.

Floating-point Arithmetic

Floating-point numbers are represented in signed magnitude form. The format is a packed signed binary fraction and a biased binary integer exponent. The fraction is a 49-bit signed magnitude value. The exponent is 15-bit biased. The unbiased exponent range is:

$$2^{-20000_8} \text{ to } 2^{+17777_8}, \quad \text{or approximately} \quad 10^{-2500} \text{ to } 10^{+2500}$$

An exponent equal to or greater than 2^{+20000_8} is recognised by the floating-point functional units as an overflow condition, and causes an interrupt if floating point interrupts are enabled.

Chaining

The chaining technique takes advantage of the parallel operation of functional units. Parallel vector operations may be processed in two ways: (a) using different functional units and V registers, and (b) chaining; that is, using the result stream to one vector register simultaneously as the operand set for another operation in a different functional unit.

Parallel operations on vectors allow the generation of two or more results per clock period. A vector operation either uses two vector registers as sources of operands or uses one scalar register and one vector register as sources of operands. Vectors exceeding 64 elements are processed in 64-element segments.

Basically, chaining is a phenomenon that occurs when results issuing from one functional unit (at a rate of one/clock period) are immediately fed into another functional unit and so on. In other words, intermediate results do not have to be stored to memory and can be used even before the vector operation that created them runs to completion.

Chaining has been compared to the technique of "data forwarding" used in the IBM 360/195. Like data forwarding, chaining takes place automatically. Data forwarding consists of hardware facilities within the 195 floating-point processor communicating automatically by transferring "name tags", or internal codes between themselves. Unlike the CRAY-1, the user has no access to the 195's data-forwarding buffers. And, of course, the 195 can only forward scalar values, not entire vectors.

Interrupts and Exchange Sequence

Interrupts are handled cleanly by the CRAY-1 hardware. Instruction issue is terminated by the hardware upon detection of an interrupt condition. All memory bank activity is allowed to complete as are any vector instructions that are in execution, and then an exchange sequence is activated. The Cray Operating System (COS) is always one partner of any exchange sequence. The cause of an interrupt is analysed during an exchange sequence and all interrupts are processed until none remain.

Only the address and scalar registers are maintained in a program's exchange package. The user's B, T, and V registers are saved by the operating system in the user's Job Table Area.

The CRAY-1's exchange sequence will be familiar to those who have had experience with the CDC 7600 and Cyber machines. One major benefit of the exchange sequence is the ease with which user jobs can be relocated in memory by the operating system. On the CRAY-1, dynamic relocation of a user job is facilitated by a base register that is transparent to the user.

References

1. The IBM 360/195. by Jesse O'Murphy and Robert M. Wade, Datamation, April 1970.

NOTE Figure 1 is reproduced on the front cover of this issue

* * * * * * * * * * * * * * *

8-DISC CRAY SYSTEM

We are now running the CRAY using all 8 disks. The final disk DD-19-53 came online on Friday 11 May. We may, however, at times have to degrade to a 7 disc system, in which case DD-19-53 will be removed. In anticipation of this I would ask users not to specifically assign files to DD-19-53, since any jobs which do this will not be runnable on a 7 disc system.

- Neil Storer

* CYBER SCHEDULER EXPERIENCE

Experience is now accumulating on the use of the Cyber scheduler introduced in March. The current class boundaries for time and memory are:

CP time	T20, T40, T200, T400, T1000, T3000, over
CM size	CM60000, CM140000, CM260000,, CM320000

(where all values are in octal).

As was stated in the last Newsletter we are experimenting with these boundaries in order to improve turnround and machine utilisation, hence they may well change.

Analysis of the workload shows that over 90% of all jobs use no more than 20 (octal) CP seconds, therefore we recommend you code T20 as much as possible to take advantage of the smallest (time) class (default is T40).

Reported in your dayfile is now the maximum amount of memory your job used. We recommend that you use this figure to base future CM requests on. To this figure add a few hundred words, to allow for possible error recoverage routines to be brought in should your program run into trouble. Then when rerunning that job you can code a much more realistic CM value and so improve your turnround.

The default for CM is 260000 (octal), it is proposed to shortly reduce this to 60000 The exact figure has been determined from an analysis of current job sizes. You would be well advised to begin coding the CM parameter for jobs over 60000 now. Note that the maximum CM size will be raised to 377700 (octal) in June. Because such jobs take all the (user) available memory turnround for them will be restricted to overnight and weekends.

- Andrew Lea

* * * * * * * * * * * * * * * *

* CHANGES TO THE CYBER DISC CONFIGURATION

At the beginning of the computer service on Wednesday morning 23 May, small changes were made to the Cyber disc configuration in an attempt to improve Cyber throughput. These changes should not have affected the majority of users, but any user who directs permanent files to particular packs in the system default permanent file set will be affected and will have to amend JCL or code.

The changes amended the situation where scratch space was located on one whole disc drive to a new situation where scratch space will be available on a portion of 3 different disc drives. We have discovered that the number of transfers to the scratch device has been very high whilst the number of transfers on the other devices is relatively low. This change will spread the scratch transfers of 3 devices and this should result in a throughput improvement.

In order to minimise any risk of problems during the change, the system default permanent files were reloaded onto new packs immediately before the start of service on Wednesday morning 23 May. The system default set permanent files now resides on packs SA001A, SA002L, SA009T and SA010S. Although system default set permanent files reside on 4 packs compared with the 3 packs originally, no additional permanent file space will be made available.

These changes do not have any impact on the use of private packs.

Those users who were affected by the above change should re-consider their approach in the light of the information below.

From time to time in a service environment, it is necessary to change the media used to hold permanent files. Such changes could occur due to configuration changes (such as this one), due to preventive maintenance (such as a periodic pack cleaning) or due to fault discovery (such as a flaw, head crash or loss of permanent files due to other reasons). We do not guarantee that the default permanent file set will ever reside on particular media.

Whilst we will do our best to inform users of any planned changes, there will be times when no warning will, or can, be given, and thus any jobs depending on particular physical media being used for the default system permanent file will be adversely affected.

It should also be recognised that in a multi-programming environment, the effect of placing files on particular packs to ensure separation, will not always be as great as one might think, due to interaction with other jobs which are running simultaneously. This observation is not as valid for the Cray, due to the more restricted nature of the workload on that machine. Also, it should be remembered that in a situation where the disc packs are often full (a situation encountered fairly frequently at ECMWF) placing a file on a particular device only ensures that the first Record Block will be on that device - the rest of the file may be on one or more other devices.

This work is part of the ongoing effort led by Steve Jay, to improve Cyber performance. Already, a reconfiguration of the Peripheral Processor Library has resulted in a reduction in disc transfers for PPL loads from around 11 per second to around 3 or 4 per second. Watch out for further improvements.

- Peter Gray

* * * * * * * * * * * * * * * *

SHORT TERM FILE STORAGE ON THE CYBER

A private set has been set up on the Cyber to hold files with 4 days' retention period. Each day all expired files will be deleted with a destructive PFDUMP.

The set currently comprises 1 pack identified as:

SN=TEMP,VSN=TEMP01.

Each morning the operators will perform a destructive PFDUMP of all expired files in this pack. Therefore user files which are catalogued on this set will survive for 4 days.

This set has been created to provide users with short term disc storage for medium to large size files that are to be accessed several times over a short period. All files that are no longer needed should be purged as soon as possible.

- Tony Stanford

* * * * * * * * * * * * * * * * *

THE 844 - 41 DISC PACK

These 2 articles are reprinted from the ECMWF Computer Newsletter NO. 8 (May 1978)

The 844-41 disc pack is referred to as the "double density" pack, and is the disc pack currently used on the CYBER 175.

This pack comprises 808 cylinders of 19 tracks. Each track contains 24 sectors or physical record units (PRUs) which each hold 64 60-bit words of data. The PRU is the smallest addressable unit of data.

Rotating Mass Storage (RMS) or Disc Space is assigned a record block at a time. On the Cyber 175 all record blocks currently comprise 56 PRUS.

The capacity of the 844-41 disc pack is equivalent to:

5,656	blocks of 4096 words
6,464	record blocks of 56 PRUS each
361,984	PRUs
23,166,976	words of 60 bits.

THE LIMIT CONTROL CARD

The LIMIT control card specifies the OCTAL number of 4096-word blocks or rotating mass storage (RMS) that may be assigned to a job. An RMS block is equivalent to 64 disc sectors or PRUs.

The RMS limit specified applies to RMS files actually created or extended by the job and does not include the INPUT file or attached permanent files. Any RMS file deleted by the job decreases the count of assigned blocks.

The default mass storage LIMIT is 3000_8 which corresponds to:

¼	of a disc pack (approximately)
1536	blocks of 4096 words
98,304	sectors of PRUs
6,291,456	60-bit words

Using LIMIT,13000 would set the RMS limit to approximately the capacity of a disc pack.

The maximum is LIMIT(20000).

- Tony Stanford

* * * * * * * * * * * * * * * * *

* NOS/BE (473) PROBLEM HARVEST

Some further points to note, in addition to those in the last few Newsletters.

- TRIVIAL RECORD MANAGER ERRORS CAN BE FATAL WHEN REPEATED 64 TIMES

Control of trivial Input/Output errors is done by setting the ERL field of the FIT to a value between Ø and 511. A trivial error becomes fatal after having occurred the number of times specified by ERL. The default value of ERL is zero, which stands for an indefinite number of trivial errors, but this does not happen in practice! For example, if a Fortran compiled program reads records with a list shorter than the record length, the record manager issues an error message informing that there are more data than required; if this happens more than 63 times the program aborts with an error mode. This will not happen any more with our next release of NOS/BE.

- CYBER JOB CARD NON-STANDARD PARAMETERS

An incorrect syntax for the Cyber job card may lead to a situation whereby an originally intended standard parameter looks like a non-standard one, and as such it will be ignored. This happens, for instance, if "CM=" is punched, instead of just "CM".

- WORD ADDRESSABLE RECORD LENGTH LIMIT

A "WA" record length is recorded in the RL field of the FIT with a maximum limit of 24 bits. If a record exceeds (2^{24}-1) characters, the FIT card holding RL is badly corrupted and an error occurs. By forcing the Record Manager to write a Record of Type W the problem disappears. This case is being investigated at present.

- Luigi Bertuzzi

* * * * * * * * * * * * * * * * *

INTERCOM PROCEDURE LIBRARY

A procedure library is being developed to assist users who wish to access the Cyber via the INTERCOM interactive terminals. The procedures aim to (a) reduce typing effort for commonly required operations (b) supply additional facilities not readily available from the standard operating system.

A basic design feature is that the user's identifier should be supplied only once (at the commencement of the session). Subsquently, the identifier is supplied automatically by the procedures where necessary (e.g., ID and FID parameters) unless explicitly overridden on a specific procedure call.

Many user permanent files are small and consequently waste much disc space. Therefore an important additional feature is the ability to reorganise files so that they become combined as separate decks of one permanent file. Procedures are available to extract, add, replace and delete decks from a file organised in this way.

The procedure library is still incomplete and some work remains to be carried out to improve response time. However, in its present form, the library is available on a test basis and some users are already assisting in an initial evaluation. Anyone who wishes to take part in this trial is welcome to contact me.

The completed library will be formally released for general use later in the summer.

- David Dent

* * * * * * * * * * * * * * * * *

TAPE ANALYSIS PROGRAM - LOOK 9

A misunderstanding has occurred regarding the use of this package recently acquired from the University of London Computer Centre (and announced in our last Newsletter). This is the program designed to help identify the properties and contents of 9 track magnetic tapes.

To request an analysis to be done on a given tape, please contact the Magnetic Tape Librarian. The Librarian will then run the package on the tape(s) concerned, this currently has to be done from the Cyber operators' console. The output will be returned to the user concerned. Because the interpretation of that output can be difficult for those not familiar with the package the Magnetic Tape Librarian or advisory staff can be called upon for assistance.

- Andrew Lea

* * * * * * * * * * * * * * * * *

CYBER-CRAY SPEED COMPARISONS

Fortran programs when executed on both the Cyber and the Cray exhibit a wide range of speed comparisons depending largely on the degree of vectorising achieved on the Cray.

The following is a sample of comparisons made using ECMWF programs and external standard test programs (CCA). Ratios are calculated on CP times.

Program		Cray/Cyber
Explicit grid point model - coarse grid - includes vectorised dynamics and physics		18
Spectral model - largely scalar		3
- vectorised		12
Fast Fourier Transform - vectorised		19 to 23
Processor tests (CCA)	FOPR00	2.5
	FOPR01	12.3
	FOPR02	4.1
	FOPR03	3.3
	FOPR04 (double precision)*	0.6
	FOPR12	2.3

* Hence, once again, we stress that Cray double precision arithmetic should be avoided from an efficiency point of view.

- David Dent

* * * * * * * * * * * * * * * * *

CYBER STATION SOFTWARE FOR THE CRAY-1

(OR THE LINK)

This article is based on a paper presented at ECODU (Control Data European Users Conference) held in London on 20-22 March. The paper was jointly produced and jointly presented by Peter Gray (Head of Systems, ECMWF) and Jim Lowell (CDC Project Manager for the work described here).

It describes the software project undertaken by Control Data Corporation to provide a Cray station for ECMWF. This station is resident in the Cyber 175. Note that it is also known as "the Link".

INTRODUCTION

In order to fully understand the requirements for the link software, it is necessary to provide some figures for the amounts of data which are expected to be transmitted across the link. The volume of observational input from around the globe at present consists of 1 million bytes per operational run. As the number of meteorological satellites increases, it is expected that this will increase.

The forecast process involves the passage of around 200 million Cray words of data (12.8×10^9 bits) to the Cyber. This figure includes restart data sets and it is important that it is transmitted rapidly and securely.

REQUIREMENTS

The overall requirements can be summarised as follows:-

- A link between the Cyber NOS/BE operating system and the CRAY COS operating system.
- Remote and local batch job access to the Cray system via the Cyber system.
- The ability to transfer files between Cyber and Cray at speeds of up to 6 million bits per second.
- The ability for operators on Cyber intercom terminals to display the status of jobs on the Cray mainframe.

These facilities are broadly similar to those provided by the standard CDC NOS-BE/SCOPE 2 station.

In addition, ECMWF were concerned to ensure that the station software should be able to handle and recover from error situations caused by user problems, magnetic media problems and hardware or software errors without any loss of data.

It was clear that to achieve a transfer rate of 6 million bits per second, there was the possibility that unrealistically high amounts of central memory and CPU time on the Cyber could be consumed. Thus we were careful to specify maximum requirements (which we felt to be realistic) for the use of central memory and of central processing time on the Cyber 175. More information on this topic will be presented later.

EXPERIENCE WITH THE STATION

A provisional acceptance of the station software took place early in December 1978. This acceptance test involved a demonstration of the facilities and the performance of the station, using a suite of jobs provided by ECMWF, and also involved the cyclic running of jobs using the station, at the same time as other jobs were run on the Cyber and the Cray. This acceptance test was a qualified success and allowed us to enter a 90-day period of acceptance for the link software and the full Cyber 175 system. At the time of the provisional acceptance test there had been no user exposure to the station software, and a list of around 30 deficiencies in the station software was generated, although most of these were minor in effect. It was decided that the implementation of all the facilities provided by the station software would be phased in order to avoid any likelihood of major problems. This phased implementation also allowed CDC to correct some of the deficiencies found during the provisional acceptance test, thereby preventing them from causing any user problems.

The implementation phase began by allowing job submission and output transfer between the Cyber and Cray. Prior to the use of the station job submission to the Cray had been done via a magnetic tape batching system produced at ECMWF. This batching system generally introduced handling delays of at least 30 minutes per job, over and above the normal running time of the job on the Cray. The introduction of the Cray station software reduced this handling time to zero and a considerable improvement in Cray job turnaround time was immediately noticed by our users. During this period, when the transfer of permanent files and magnetic tapes was not allowed to be used by the general user population, one or two other station problems were discovered and rapidly fixed. However, these problems did not cause any undue user difficulties.

This first phase of implementation continued for two weeks until just prior to Christmas 1978, when we felt sufficiently confident to allow the station to be operated for 24 hours per day.

The second phase of implementation began on New Year's day 1979 when the facility to transfer system default set permanent files (excluding the use of private packs and magnetic tapes) was released to the users. This facility, along with a suite of data conversion routines provided to convert binary data between Cray and CDC formats, allowed our users to readily transfer data between the Cyber and the Cray and, for example, permitted Cray jobs to send back data to the Cyber which would then be processed by graphics packages on the Cyber and eventually produce weather maps. Again, this second phase lasted for two weeks until mid-January 1979.

At that time, the third and final phase of implementation was introduced: the remaining facilities were released to our users. Acceptance of these facilities by our users was very good and few problems were found adapting existing jobs to use them.

Turning now to the reliability of the station software during the first 9 weeks of 1979, the station software achieved a meantime between failure of 51.1 hours and a serviceability figure of 99.6%.

Turning now to look at the performance of the station, at the time of the provisional acceptance test early in December 1978 there were deficiencies in some aspects of the performance, although these were balanced by better performance than expected in some other areas.

In March 1979, the following performance figures were achieved:-

- A single file binary transfer rate of 3.7 million bits per sec. to 844-41 full tracking.
- A single file binary transfer rate of 4.4 million bits per sec. to a 679-6 6250 bpi/150 ips magnetic tape deck.
- Transfer rates of up to 5.5 million bits per sec. when transferring two or more binary files.
- Central processor utilisation on the 175 of less than 1% when the link system is idle.
- Central processor utilisation of 17.8% when transferring two blocked binary files at 5.3 million bits per sec. to magnetic tape.
- Central processor utilisation of 14.2% when transferring two blocked binary files at 5.3 million bits per sec. to 844-41.
- A central memory usage of 5K decimal words (12K octal) when the station is idle.
- A central memory utilisation of 15K decimal (36K octal) plus 20K decimal (50K octal) per file being transferred whilst the station is busy.

Work is currently being undertaken by CDC and Cray to tune the station software (in the Cyber and the Cray) to achieve the CPU utilisation and transfer rate requirements.

Work will also be done to provide a station to achieve optimal operational performance. It is expected that such a station will require less memory, at the expense of a lower transfer rate.

The minimum station size achievable is:-

5K decimal whilst idle.
9K decimal when busy, plus $5K_{10}$ per active SPOT.

This situation gives a maximum transfer rate of 600 thousand bits per second.

AFTERTHOUGHTS

As would be expected when a major piece of software enters service, there are a number of small problem areas which were overlooked in the design phase and which must be solved. These are as follows:

- The lack of provision for operator control of Cray jobs from the Cyber. The station in its present form allows an intercom operator to examine the state of the queues on the Cray, but no similar facility is allowed for the DSD operator to control jobs on the Cray. Control of all Cray jobs must be exercised from the Data General Eclipse maintenance control unit which is supplied as part of the Cray configuration. It is intended that the station software will be enhanced during 1979 to overcome these problems.

- The Cray protocol allows for only up to 8 files to be transmitted simultaneously in each direction. In addition, the CDC station at presentallows only 12 simultaneous transfers in total, due to lack of memory and control points. Given this limitation we expect occasions when file transfers will be tied up awaiting operating action such as mounting of private disk packs or magnetic tapes, when other transfers could in fact take place. This problem may be solvable by having either the Cray or the Cyber limit the number of different types of transfer at any one time, in effect, implementing a means of having several classes of transfer, each class being limited to an upper limit.

- The Cyber station is presently capable of handling only one Cray print file (as opposed to permanent file transfers) simultaneously. Overall this limitation is not a problem, but there are times when very long print files are sent back to the Cyber causing holdups of some minutes in transferring other files in the Cray print queue. This problem could be overcome by changing the Cyber link software to run extra copies of the output queue SPOT when necessary.

- The transfer of coded files from the Cray involves very large CPU overheads on the Cyber. This overhead is due to the fact that the file code conversion from display code to ASCII and vice versa is carried out inside the Cyper CPU. It is not feasible to carry out this code conversion inside the drive PPU, and the provision of a helper PPU would not solve the problem, as the PPU does not have sufficient processing power to convert and re-format the 6-8 bit characters at least every 6μs. Code conversion by the link hardware would have solved the problem but it was not possible to easily achieve the required flexibility in the link hardware. In general, interfacing machines with dissimilar character sets and/or word lengths should be arranged to ensure the minimum of character set conversion takes place.

- Presently, the lack of a particular Cray facility prevents the Cyber station from sending messages back to the Cray in response to a DISPOSE command from the Cray. It is expected that Cray will provide the required facility later in 1979. This problem has caused our users some heartache due to the fact that transfers which they expected to complete satisfactorily, did not take place due, for example, to an error in the JCL used.

- Cray have recently altered the internal structure of their station control package to allow a higher throughput. Whilst this did not require any changes to the Cray station protocol, it did affect the way in which transfers took place and thus it is necessary for the CDC station software to be retuned to take account of this new Cray software. This retuning is now taking place.

- We discovered that the permanent file SPOT was using very large amounts of CPU time (≈35%) when transferring binary blocked files, whereas the magnetic tape SPOT only used half this amount. A large amount of investigative work took place before it was established that 'periodic recall' has a different meaning when doing disc transfers under NOS-BE. A stack processor 'feature' causes the CPU program to be recalled after <u>every</u> PRU (using the M.BUFPTS monitor request) rather than after the specified time interval. A means of disabling this feature should be provided since it prevents effective program tuning.

The most significant problem we have encountered in service is not particularly a problem of the software on either the Cyber or the Cray but is, rather, philosophical. The problem is essentially that when mass storage channels on the Cray or the Cyber are busy due to the load on the machine then it is impossible for the station to achieve a frequent enough transfer rate to allow the high transfer rates required to be maintained. When running our forecast model on the Cray, all of the Cray disk channels and drives are kept busy and thus it is difficult for the station software to transfer data to or from a drive due to the head movements required. Thus, although the station software on both sides is capable of transferring around 6 million bits per sec. a transfer rate of only 10% of that figure is frequently observed in a busy system. Clearly, careful placing of files and changes to input output priorities on the machines can improve this problem but it is undoubtedly true that whilst high link transfer rates are necessary to transfer files to or from an empty system, they are rarely achieved in a real, live environment. It is expected that a tuning exercise will alleviate this problem.

- Peter Gray (ECMWF)
- Jim Lowell (CDC)

NEW COASTLINE DATA BASE

For users of ECMWF's Contouring Package, there is now a new coastline data base. It is stored under the name COASTLINES, and has multi-read access (no 'MR=1' needed). It must be attached to TAPE90 as for the old data base.

ATTACH,TAPE90,COASTLINES

This database has been obtained from the Rutherford Laboratory and has been derived from data collected by Dr. A.V. Hershey of the Naval Weapons Laboratory.

Compared to the old data base (which will still be available as BACKBASE,ID=EWPLOT), it has the following features:

i) There are no State boundaries

ii) There are no ice-limits

iii) Less detail is present on the coastlines, but there are fewer errors than with the old database.

iv) Less space is required on disk for storing the new database, and a lower CP time is required for generating maps.

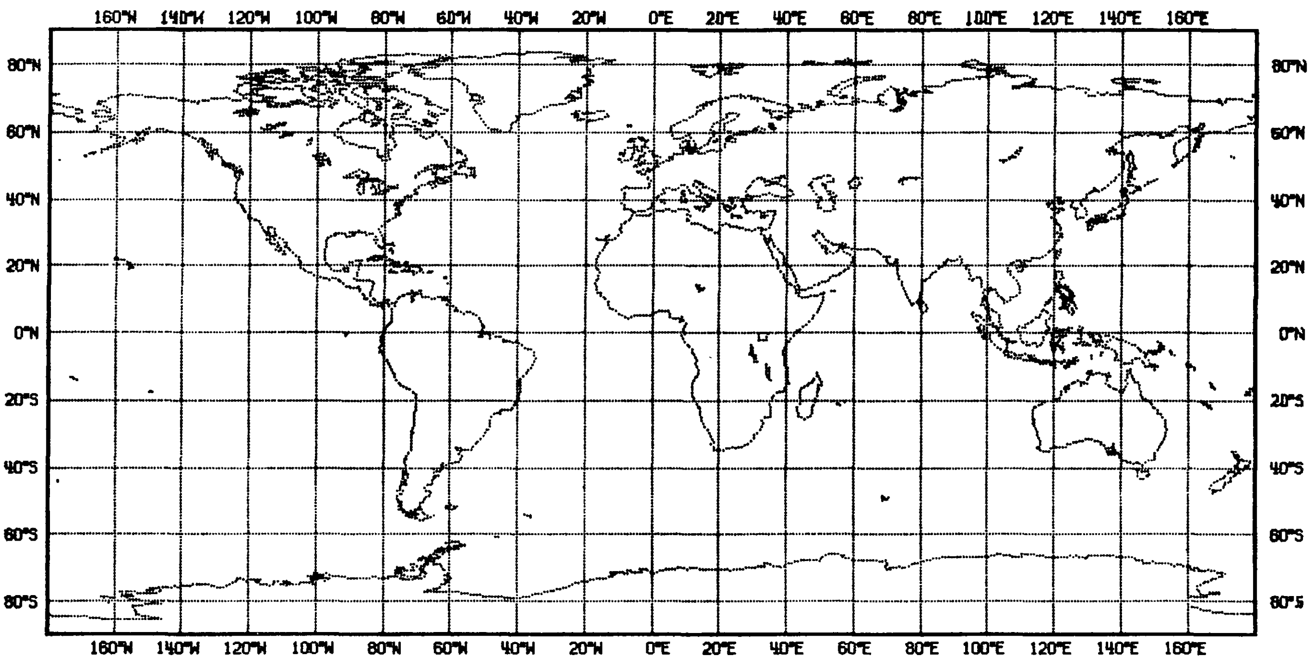
Users are requested to try the new data base and to report any problems directly to us. Some examples of the new data base are shown, these should be compared with examples C-1, C-2 and C-3 from the Contouring Package Users Guide.

- Alain Lemaire, Howard Watkins

* * * * * * * * * * * * * * *

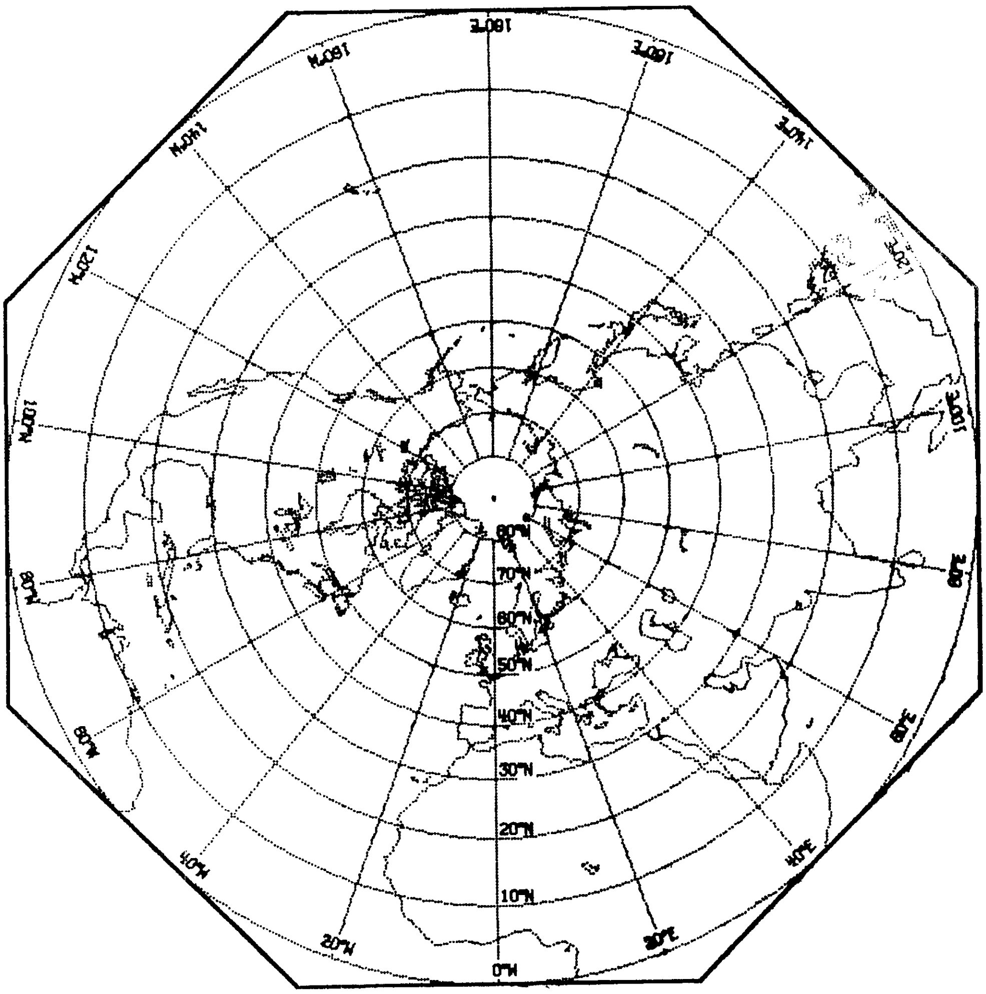
Example 1 of new coastline data base

(Compare with example C-1 in the New Users Guide)



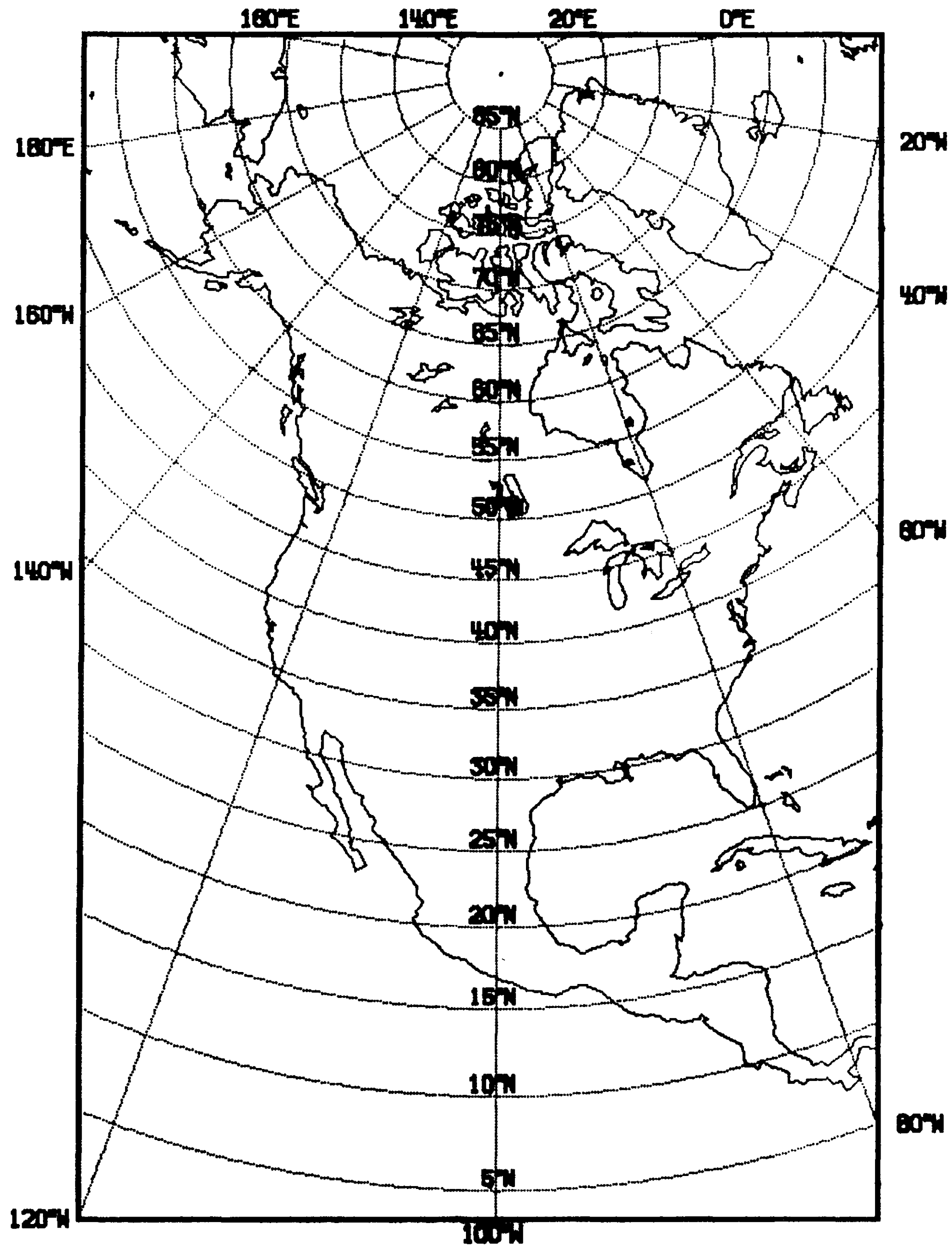
Example 2 of new Coastline Data Base

(Compare with example C-2 in the Users Guide)



Example 3 of new Coastline Data Base

(Compare this with example C-3 in the Users Guide)



* TERMINAL PROCEDURE AT ECMWF

Now that some terminals are installed in offices, please follow the guidelines outlined below:

1. To move a terminal to another office

a) Switch OFF mains.
b) Disconnect interface plug
c) Move to new office.
d) Connect interface plug - take care, the plug only fits one way, and the pins can become damaged.
e) Connect mains plug and switch ON.
f) Telephone ext. 334. State both OLD and NEW room numbers to obtain a line to the computer.

2. Reporting Terminal Faults

a) Telephone Computer Reception, ext. 332 (Judy Herring).
b) Quote the number painted on the front of the terminal (e.g. N12)
c) Describe the nature of the fault and give your name with the room number in which the terminal can be found.

- Eric Walton

* * * * * * * * * * * * * * * *

PROBLEMS WITH NEWBURY LAB VDU's

(or why your terminal doesn't behave the way you want)

This article is reprinted from the ECMWF Computer Newsletter No.8 (May, 1978)

Many users have experienced problems using our Newbury terminals (the visual display units) on Intercom. These are caused by the extra features built into the units which Intercom cannot handle.

The most common cause for complaint is the appearance of special mode character displays on the screen. There are 3 special modes for the display: blinking, inverted and protected. In blink mode the affected characters flash on and off about twice per second; in inverted mode the characters appear dark on a (local) bright background, and in protected mode the characters cannot be removed from the screen. Protected characters are especially annoying because any subsequent output only appears in the gaps between the protected characters, which do not roll off the screen when the display shifts up one line but reappear on the bottom line when they are shifted off the top.

Two other 'strange' states the vdu's can get into are 'send page' mode, where the vdu tries to output all the data from its screen to the computer, and direct cursor addressing, where a group of 3 characters from the computer or the keyboard drives the cursor (the position where the next character will be displayed) to a specific location on the screen. Send page can be very disruptive if it happens by accident, because the screen contents are unlikely to make sense to the Intercom, and any error messages sent from the computer will be mixed with the existing screen characters to produce further confusion. Cursor addressing is less of a problem because it simply appears to ignore two further characters and resume output at a strange point on the screen.

Users of these terminals may suffer such problems from one of 3 possible causes: obviously, random key depressions can produce odd effects (especially CTRL combinations); random noise on the communications line can appear to be control characters to the vdu, and can switch it into one of these modes; but the most common cause is for the vdu to receive input simultaneously from the computer and from the keyboard: the internal circuit which allows characters from the keyboard to appear on the screen then mixes up the two lots of data and effectively sends garbage to the vdu.

What can be done about these effects? Obviously, the cure for keying random characters and for entering characters while the computer is sending data is not to do it: this is not always practicable however since some output from Intercom may be unsolicited - e.g. operator messages. About line noise, please report such instances to Computer Operations.

The most practical approach is to learn how to clear a vdu to its normal operational state from any condition it may have got into. The following sequence of key operations will always work (if you find a case where it doesn't, let me know) and leave the vdu in normal state with a clear screen.

1) Put the vdu in LOCAL mode
2) Hit RESET
3) Hold down CTRL and hit S - this turns off any protection.
4) Hold down CTRL and hit F - this turns off inverted mode.
5) Hold down CTRL and hit W - this turns off blink mode.
6) Hit CLEAR
7) If the PAGE button is lit, hit the ROLL button above it, which should light (on the newer vdus the PAGE button depressed means PAGE, and not depressed means ROLL).
8) Take vdu back out of LOCAL by hitting the LOCAL key again.

Not all of these steps are always necessary, but if in doubt it is always better to be sure. The 3 control combinations can be used selectively depending on what is wrong with the display.

Finally it perhaps ought to be stated that these special facilities and modes of the terminal are not always a liability: it is possible to use all of them from a FORTRAN program by making the proper subroutine calls. Thus where a lot of output (or input) is made interactively to a single program it may be practical to make use of the special feature terminals. Any users who believe they may be able to do so should contact User Support in the first instance for advice.

- Dick Dixon

* * * * * * * * * * * * * * *

COMPUTER MANUAL DISTRIBUTION WITHIN ECMWF

Those of you who have not been here very long may like to know our system of manual distribution. All manuals are organised into the sets defined below. Each user receives a PERSONAL set (which he may take with him when he leaves the Centre) and each room (containing at least one user) is issued with a USER OFFICE SET. REFERENCE sets are located in the advisory office, the user area of the computer hall and the library. A fourth reference set is available from User Support for training purposes. At the operators console there is also a MACHINE ROOM SET for reference only. Additionally, there are special sets for system programmers, user support and computer operators.

PERSONAL SET: (CDC) NOS/BE User's Guide
FTN Reference
FTN User's Guide
(CRAY) CFT Fortran Reference

USER OFFICE set: (CDC) NOS/BE Reference
NOS/BE Diagnostic Handbook
COMPASS Reference
LOADER Reference
Basic Access Methods
Record Manager Users Guide
FTN Common Library Math Routines

INTERCOM Reference
INTERCOM Guide for FTN users
UPDATE Reference

(CRAY) CAL Assembler
COS External Reference Specs
UPDATE Reference

- Pam Prior

* * * * * * * * * * * * * * *

*DOCUMENTATION CHANGES

ECMWF Bulletins B1.5/1 (Advisory Services), B6.3/2 (Utility Programs from the CERN Library) and a reprint of B1.1/1 (Naming Conventions), now retitled 'Naming Conventions and Registration Procedure', were distributed in early May. User Support have prepared an "Introductor User Guide" to the ECMWF computer facility. It s purpose is to outline the main facilities, and how to use them, for new or inexperienced users. Draft copies have been circulated to Section Heads, plus those who attended our recent training courses, to obtain some initial reaction as to its usefulness and clarity. Comments on any aspect of this guide are very welcome, they should be sent to User Support before 1 July for inclusion in the first issue of the guide.

- Pam Prior

* * * * * * * * * * * * * * *

COMPUTER USER TRAINING COURSES - SEPTEMBER

The Centre has just sent details to Member States of the computer user training courses it proposes to run in September of this year. They are similar to those courses run during April, with a few minor changes based on experience gained.

The courses offered for September are as follows:

- Course B - "Basic Usage" (September 17-20) This is intended to introduce users to the basic Centre computer facilities, concentrating on the CRAY-1.

  An (optional) fifth day will be added (September 21) covering use of the ECMWF meteorological operational data base.

- Course C - "Cray User Course" (September 24 - 28) A detailed course going through the main CRAY-1 facilities in depth. It is aimed at those who will make substantial use of the CRAY-1.

- Course D - "Cyber User Course" (October 1 - 5) This is an in-depth course covering the main Cyber facilities, and is aimed at those who will extensively program for the Cyber, or use its file handling and data storage facilities a lot.

Nominations to attend any of these courses are now invited, and they should reach ECMWF by 13 July at the latest. Please note that nominations from Member States users must be sent through the national meteorological service of that Member State.

- Andrew Lea

INDEX

INDEX of Still Valid Newsletter Articles

This is an index of the major articles published in this ECMWF Technical Newsletter series, plus those in the original ECMWF Computer Newsletter series. Articles in the original series which are still valid will eventually all be reprinted in this Technical Newsletter, making the Computer Newsletter then obsolete. Currently, Computer Newsletter numbers 1 to 8 can be thrown away. Some back copies of Computer Newsletter numbers 9 to 11 are still available, please apply to Mrs. P. Prior (ext. 355).

As one goes back in time, some points in these articles may no longer be accurate. When in doubt, contact the author, or User Support.

INDEX

GENERAL

METEOROLOGY

*C indicates article appeared in the previous ECMWF Computer Newsletter series.

USEFUL NAMES AND 'PHONE NUMBERS WITHIN ECMWF

		Room*	Ext.**
ADVISORY OFFICE Open 9-12,14-17 daily		CB 037	308/309
Computer Division Head	- Rob Brinkhuysen	OB 009A	340/342
Disk Space and Permanent File Problems		AS FOR ADVISORY	
DOCUMENTATION Officer	- Pam Prior	OB 016	355
Libraries (ECMWF,NAG,CERN, etc.)	- John Greenaway	OB 107	354
OPERATIONS	- Console/Shift Leader	CB Hall	334
Reception Counter	- Judy Herring	CB Hall	332
Terminal Queries		AS FOR SHIFT LEADER	
Operations Section Head	- Eric Walton	OB 002	349/351
Deputy Ops. Section Head	- Graham Holt	CB 023	307
METEOROLOGICAL DIVISION			
Division Head	- Roger Newson	OB 008	343
Operations Section Head	- Austin Woods	OB 107	406
Applications Section Head	- Joel Martellet	OB 011	344
Meteorological Analysts	- Ove Akesson	OB 106	380
	- Rauno Nieminen	OB 104	380
	- Stefan Panin	OB 104	380
	- Herbert Pumpel	OB 106	380
Registration (User and Project Identifiers, INTERCOM)	- Pam Prior	OB 016	355
Research Department Computer Co-ordinator	- Rex Gibson	OB 126	384
Tape Requests	- Pauline Litchfield - George Stone	CB Hall	335/334
User Support Section Head	- Andrew Lea	OB 003	348

---

* CB - Computer Block
OB - Office Block

** The ECMWF telephone number is READING (0734) 85411