SUPERMUX
A MULTI-HOST FRONT END CONCENTRATOR SYSTEM
FOR ASYNCHRONOUS CONSOLES

T. Bruins, K.S. Olofsson, E.M. Palandri, B. Segal,
H.J. Slettenhaar and H. Strack-Zimmermann
© Copyright CERN, Genève, 1974

Propriété littéraire et scientifique réservée pour tous les pays du monde. Ce document ne peut être reproduit ou traduit en tout ou en partie sans l'autorisation écrite du Directeur général du CERN, titulaire du droit d'auteur. Dans les cas appropriés, et s'il s'agit d'utiliser le document à des fins non commerciales, cette autorisation sera volontiers accordée.

Le CERN ne revendique pas la propriété des inventions brevetables et dessins ou modèles susceptibles de dépôt qui pourraient être décrits dans le présent document; ceux-ci peuvent être librement utilisés par les instituts de recherche, les industriels et autres intéressés. Cependant, le CERN se réserve le droit de s'opposer à toute revendication qu'un usager pourrait faire de la propriété scientifique ou industrielle de toute invention et tout dessin ou modèle décrits dans le présent document.

Literary and scientific copyrights reserved in all countries of the world. This report, or any part of it, may not be reprinted or translated without written permission of the copyright holder, the Director-General of CERN. However, permission will be freely granted for appropriate non-commercial use.

If any patentable invention or registrable design is described in the report, CERN makes no claim to property rights in it but offers it for the free use of research institutions, manufacturers and others. CERN, however, may oppose any attempt by a user to claim any proprietary or patent rights in such inventions or designs as may be described in the present document.
SUPERMUX

A MULTI-HOST FRONT-END CONCENTRATOR SYSTEM

FOR ASYNCHRONOUS CONSOLES

T. Bruins, K.S. Olofsson, E.M. Palandri*, B. Segal,
H.J. Slettenhaar and H. Strack-Zimmermann

GENEVA
1974

*) Now at Division of Computing Research, CSIRO, Canberra, Australia.
ABSTRACT

This report describes a front-end concentrator/multiplexer system "SUPERMIX" for asynchronous time-sharing consoles operating on a minicomputer which has recently been put into operation at CERN. The concentrator/multiplexer will control up to 36 consoles at speeds between 110 and 9600 bits per second and has the capability of dynamically connecting these consoles to several large Host computers. The report describes both the hardware and the software developed at CERN, as well as the diagnostic and simulation techniques used to test the system.

The system runs on an HP 2100 minicomputer, and a special-purpose microcode is used to optimize console handling and facilitate the implementation of the system.

The Host computers are the CDC 6000 series computers using the INTERCOM time-sharing system.
CONTENTS

INTRODUCTION

CHAPTER I : HARDWARE OVERVIEW

1. THE HOST-CONCENTRATOR COUPLER
   1.1 Introduction
   1.2 Coupler description
      1.2.1 General
   1.3 Logic description coupler Host side
      1.3.1 Connect procedure
      1.3.2 Functions Host side
      1.3.3 The Interrupt system
      1.3.4 Read operation
      1.3.5 Write operation
      1.3.6 Functions: Concentrator side
      1.3.7 Parity checking/generation
      1.3.8 Coupler front panel
      1.3.9 Hardware specifications

2. THE HP ASYNCHRONOUS TERMINAL INTERFACE
   2.1 Control functions
   2.2 Status
   2.3 Data output procedure
      2.3.1 The 8-bit mode operation
      2.3.2 The 16-bit mode operation
   2.4 Data input procedure
   2.5 Clocks
   2.6 Interface circuits
   2.7 Switch settings

3. THE COMMUNICATION CONTROLLER
   3.1 Its function
   3.2 System hardware

4. DATA SETS

CHAPTER II : SOFTWARE OVERVIEW

1. HOST-CONCENTRATOR PROTOCOL
2. HP CONCENTRATOR SOFTWARE
   2.1 Load handling
   2.2 Console control

Page

1
3
3
3
3
5
5
5
7
7
8
10
11
11
14
14
15
16
16
17
18
18
19
19
21
22
24
24
25
25
2.3 The system control tables  27
2.4 System configuration  27
2.5 The microcode  27
2.6 The console driver  27
2.7 The Host driver  29
2.8 The interval timer handler  29
2.9 The buffer pool  29

3. THE PP DRIVER SOFTWARE  29
   3.1 HPZ driver call  29
   3.2 HPZ structure  30
   3.3 HPZ INTERCOM interface  31

4. SOFTWARE PREPARATION  32

5. USER PROGRAM INTERFACE  35
   5.1 Display code records  36
   5.2 ASCII code records  36
   5.3 Free code records  36
   5.4 Input code selection  36
   5.5 Console parameter table setting  37
   5.6 Assistance facility  39
   5.7 Screen Full and Type Ahead  40

CHAPTER III : TEST OVERVIEW  41

1. COUPLER DIAGNOSTICS (DIAGNOS)  41
   1.1 CM resident DIAGNOS  41
   1.2 PP resident DIAGNOS  42
   1.3 HP resident DIAGNOS  42

2. THE TEST PROGRAM FOR THE HPATIC LINE INTERFACE  42
   2.1 The control console  42
   2.2 The program Search  42
   2.3 The program Test  43
   2.4 Definition of the Test slot  43
   2.5 The different tests  43
      2.5.1 "Break" test  43
      2.5.2 "Character Lost" test  43
      2.5.3 "Flag" test  44
      2.5.4 "Data" tests  44
      2.5.5 "Random" data  44
      2.5.6 "Delayed" operation  44
      2.5.7 "No Print" facility  44
      2.5.8 "Output" only  44
      2.5.9 "Total" test  44
      2.5.10 "Parity" test  45
   2.6 Wrong parameter settings  45
   2.7 Examples  45

3. HOST-CONCENTRATOR SIMULATION  46

4. A MULTIPLE-USER SIMULATOR  47
INTRODUCTION

The CERN Central Computer facility consists of a Control Data Corporation (CDC) 7600 and three CDC 6000 series computers together with a number of smaller machines. It is used mainly by high-energy physicists working on the CERN site, which covers an area of about 3 km by 5 km. Time-sharing consoles located on the site are normally connected to the computer centre by CERN-owned cables, and instead of Modems, simple CERN-developed base-band transmission units operating over a wide range of speeds are used. There are no Post Office imposed speed limitations on the use of these lines, and a wide variety of asynchronous consoles are installed ranging from 110 bits per second (bps) Teletypes to 9600 bps alphanumeric and graphic displays. Synchronous consoles are not generally used because of their added cost and the software problems they cause.

With the delivery of the CDC 7600 a decision was made to use the standard CDC operating system. The CDC 7600 is coupled to a CDC 6400 computer which acts as an I/O processor and all multi-access and time sharing is done in the CDC 6400. This therefore involved the use of CDC 6000 series hardware for the communications, which proved most unsatisfactory in the CERN environment because it restricted the use of asynchronous consoles to 110 bps. Hence it was decided to develop a new front-end console concentrator for the CDC 6400 using a small computer (see figure below).

![Diagram of initial target configuration](image)

Initial target configuration

The over-all requirements were few:

a) The concentrator had to be capable of simultaneous connection to a number of Host processors. Plans for the full exploitation of the CDC 7600 call for the later use of both a CDC 6400 and a CDC 6500 as I/O processors. The concentrator subsystem had to be able to connect to both these Host machines, and to any additional large processors that might later be added to the central computer complex. Initially the concentrator had to support consoles using the INTERCOM interactive subsystem of the
CDC 6000 series SCOPE operating system, but this requirement was not meant to exclude later connection to other time-sharing systems in other hosts.

b) The concentrator had to be able to support a large range of asynchronous consoles, including hard-copy consoles and displays, and it had to be extremely flexible with regard to the addition of other types of consoles.

c) The concentrator had to handle up to 36 consoles at speeds up to 9600 bps.
CHAPTER I

HARDWARE OVERVIEW

Apart from the Hewlett-Packard 2100, all the hardware of SUPERMIX has been developed and constructed at CERN. This hardware can be divided into four sections. Each section will be described in detail in the next sections:

1. The Host-concentrator coupler.
2. The HP asynchronous terminal interface.
3. The Communication controller.
4. Data sets.

The experience gained with the FOCUS system\textsuperscript{1} has been of great help to the development team.

1. THE HOST-CONCENTRATOR COUPLER

1.1 Introduction

The Hewlett-Packard concentrator is directly connected to a 3000-compatible channel converter (6681)\textsuperscript{2} of the CDC 6000 series Host machines by means of a CERN-developed parallel channel coupler (Appendix 1). This coupler provides the circuitry to convert the CDC channel levels into TTL logic levels and vice versa. It contains a status register by which the two machines may signal each other to synchronize I/O transfers. Furthermore, two program-selectable modes of mapping between the 16-bit words of the HP 2100\textsuperscript{3}) and the 12-bit CDC 6000 series channel, are provided by means of a buffer register. One DMA channel in the HP 2100 is scheduled by software for all the Host connections, and transfers between Host and concentrator may proceed at speeds up to 1,000,000 12-bit words per second (see Fig. I.1).

1.2 Coupler description

1.2.1 General

The coupler is connected to the HP 2100 by means of two full duplex I/O channels. The interfaces used are two microcircuit interface cards, HP part number 12566-001, ground true version\textsuperscript{4}). One I/O channel is used for status and control and the other for data transfer. The connections to the CDC Data Channel Converter are made via two 9-pair CDC cables. Provisions are made for daisy-chaining as well as last equipment termination. CERN-developed transmitters and receivers are used.
Fig. 1.1  Host-concentrator coupler: general flow diagram
1.3 Logic description coupler Host side

1.3.1 Connect procedure

A CDC data channel may have eight possible pieces of equipment attached to its set of I/O cables. These are selected individually by means of the Connect signal. A 12-bit Connect code is transmitted via the data lines, the upper 3 bits of which select one of the eight pieces of equipment. The coupler examines this code upon detection of the Connect signal; if the code matches with the setting of the device selector switch, the coupler returns a Reply signal and from that moment the coupler is connected to the channel, provided that the parity of the code is correct. In the case that the parity is wrong, or the unit selected is absent (dual channel coupler), the coupler returns a Reject signal. Both Reply and Reject signals cause the channel to drop the Connect signal. If the coupler is connected and receives another Connect code which does not match with the device selector setting, it disconnects immediately.

1.3.2 Functions Host side

After connection the coupler is able to accept Function codes\(^5\). Various operating modes and conditions are specified using the Function instruction. The 12-bit Function code is placed on the data lines and a Function signal is sent by the channel. If the Function code is valid and the parity correct, the coupler responds with a Reply signal and executes the function. Should the coupler receive a non-valid Function code or detect a wrong parity, no Reply is sent and in that case the channel will generate an internal Reject.

The existing Function codes are as follows:

i) Select Mode I transfer (0001)
ii) Select Mode II transfer (0002)
iii) Select Interrupt on Hewlett-Packard in Read or Write (0004)
iv) Clear Interrupt on Hewlett-Packard in Read or Write (0010)
v) Select Interrupt on status register cleared by Hewlett-Packard (0020)
vi) Clear the status, clear Interrupt (0040)
vii) Set Interrupt mask register to bit pattern xxx (01xxxx)
viii) Selectively set status register to ones for ones in pattern xxx (10xxxx)
ix) Selectively clear status register to zeros for ones in pattern xxx (11xxxx).

The numbers within brackets represent the octal Function code. The x represents an optional bit setting and X represents an optional octal setting.

1.3.2.1 Functions 0001 and 0002

These are the two modes of mapping between the 16-bit words of the HP 2100 and the 12-bit CDC channel. Mode I maps one HP word into two CDC bytes, of which only the right least significant bits carry information. Conversely, two CDC bytes, each with eight bits of information, are mapped into one 16-bit HP word (see Fig. I.2). Mode II maps one CDC byte into one HP word, setting bits 6, 7, 14, and 15 to zero. Similarly, one HP word carrying 12 bits of information is assembled into one CDC byte (see Fig. I.2).
1.3.2.2 **Function 0004**

This function enables the Interrupt circuitry in the coupler to issue an Interrupt signal to the CDC channel whenever the HP is ready to commence a Read or Write operation. The HP computer indicates this condition in the status register by setting bit 10 for Write or bit 11 for Read.

1.3.2.3 **Function 0010**

This function clears the Interrupt signal and negates Function 0004, which means that no Interrupt signal will be generated when bits 10 or 11 are set in the status register.

1.3.2.4 **Functions 0020 and 0040**

Function 0020 enables the Interrupt circuitry in the coupler to issue an Interrupt signal to the CDC channel when the 12 least significant bits of the status register are being cleared. Function 0040 clears the Interrupt signal and negates Function 0020.

1.3.2.5 **Function 01**

The 10 least significant bits of the CDC byte containing this function are set into the 10-bit Interrupt mask register (IMR) in the coupler. Whenever a bit in the IMR corresponds to a corresponding bit in the status register, an Interrupt signal is generated.

1.3.2.6 **Functions 10 and 11**

The 10 least significant bits of the CDC byte containing Function 10 selectively set the corresponding bits in the status register, and similarly Function 11 selectively clears those bits in the status register which correspond to the ones in the 10 least significant bits of the CDC byte.

**Shaded areas are set to logical zero.**

Fig. I.2 Mapping modes I and II
1.3.3 The Interrupt system

As we have seen in the previous section, there are three ways in which the HP computer may initiate an Interrupt to the CDC machine. Whether an Interrupt will be generated depends upon conditions set by functions from the CDC machine itself. Those three possibilities are as follows:

i) Setting of status bits 0-9. Generation of Interrupt signal depends upon the setting of the DMR.


iii) Setting of status bits 10 or 11. Generation of Interrupt signal depends upon Function 0004.

Interrupts are always sent to the CDC channel even when the coupler is not connected by the channel.

1.3.4 Read operation

A Read operation must always be preceded by a Connect procedure and Function selection. By "Read" is meant reading into the CDC machine, which implies output from the HP. The two machines must also be completely synchronized prior to any data transfer. This is accomplished through status exchanges and Interrupts. There are two transfer modes which will be discussed separately.

1.3.4.1 Read Mode I

After the initialization procedure, the CDC channel issues a Read signal and a Data signal to the coupler. The data buffer in the coupler is now ready to accept a word from the HP. As soon as the buffer receives the first word, the eight most significant bits are gated through to the data transmitters, which are connected to the CDC channel (see Fig. 1.2). The coupler then generates a Reply signal to the CDC channel. The first byte is now read into the CDC channel. The channel responds by dropping its Data signal which in turn causes the coupler to drop its Reply. As soon as the channel issues another Data signal, the second byte may be gated onto the data transmitters. The second byte is composed of the eight least significant bits of the buffer register. Another Reply is sent to the channel and will drop when the Data signal is dropped. Another word from the HP may now be read into the buffer.

Twice as many bytes are generated as there are words sent from the HP. The Read signal remains true during the whole Read operation, which normally ends by dropping the Data signal and the Read signal after the last HP word is sent. It should be noted that the two computers inform each other during the initialization phase about the number of words to be sent. If there is a mismatch between the two machines, for example if the HP should fail to send further words while the CDC is expecting them, the coupler would generate an End of Record signal automatically after a time-out of 25 μsec. This signal is sent to the CDC channel and will terminate the transfer.

We will now refer to the circuit diagrams of the coupler (see Appendix 1).

In terms of signals in the coupler a Read operation in Mode I looks as follows.
The HP outputs OA 15 and ENA (Encode Ch. A), which set SB 11. The CDC starts Read operation by sending Read and Data signal (DS). Read and SB 11 make ROP (Read operation), which resets the byte counter A 1. It now waits for the HP to send a word to the buffer register (BR), which is accomplished by the HP sending OB 0-15 (Data) and ENB (Encode Ch. B). ENB and ROP make the signal OPB, and ENB together with AT and ROP make BUR 2 (Buffer Ready), which together with OPB causes the HP word to be strobed into the buffer by means of STR 1 and STR 2. BUR 2 together with DS and ROP generate RREP 2 (Read Reply), which is sent to the CDC with the Data (RB 0-7), gated with R 1. The CDC drops DS causing RREP 2 to drop and flopping byte counter to A 1. Upon arrival of a new DS, the coupler responds with a RREP 2 and Data, gated with A 1. The CDC receives the second byte and drops DS. The coupler switches the byte counter to AT, drops RREP 2, and sends OFLB (Flag) to the HP data channel, indicating that the word has been transferred.

The OFLB causes the HP to drop ENB thus dropping OPB in the coupler. OPB resets OFLB, which in turn resets BUR 2. The HP then sends a new ENB accompanied by a word, and the procedure for transferring it to the CDC is repeated.

1.3.4.2 Read Mode II

Similarly to the Mode I transfer, the CDC computer has to initialize the coupler before sending a Read signal together with a Data signal. The HP computer then outputs a word to the buffer register in the coupler. The buffer register contents are disassembled into one CDC byte (see Fig. 1.1) and sent to the CDC together with the Reply signal. The CDC then drops its Data signal and as a result the Reply drops. This completes the transfer of one word for the CDC side. For the HP side however, a Flag has to be sent in order to communicate to the HP that the transfer of the first word has taken place. This is accomplished by the coupler generating a Flag on the dropping of the Data signal. The HP can now output the next word, which is transferred to the CDC by means of a new Data signal. At the end of the transfer the CDC drops Read, which terminates the operation.

This transfer described above can be translated into signals in the coupler as follows.

The HP outputs ENA and OA 15 which make the SOB (Output Operation) setting SB 11. SB 11 and Read from CDC constitute ROP. Together with Read the CDC sends a DS. The HP now outputs the Data word by sending OB 0-15 and ENB. ENB together with AT and ROP make the BUR 2 signal, which is being used together with ENB and ROP to strobe the data into the buffer register. Since the Data word is in the buffer register, an RREP 2 can now be generated, sending the data to the CDC. RREP 2 is composed of DS, ROP, and BUR 2, and data gating is achieved by signals R 4 and MOD 2. DS now drops causing the RREP to drop and sets OFLB, which in turn resets BUR 2 and sends a Flag to the HP causing ENB to drop. A new DS can now appear from CDC and the HP can also output another word.

1.3.5 Write operation

Like the Read operation, a Write operation must always be preceded by a Connect, similarly to a Read operation. A Write operation is carried out blockwise, and synchronization and initialization are also achieved through status exchanges. Once these procedures are completed, a Write operation can take place. There are also two different data transfer modes (Section 1.3.2.1), which will be treated separately.
1.3.5.1 Write Mode I

After having completed the initialization and synchronization, the Write procedure can commence. The coupler must be logically connected to the CDC channel, which starts the procedure by sending one byte containing eight valid bits, while the four remaining bits are set to zero. The byte is strobed into the most significant part of the buffer register and a Reply is sent back to the CDC. Another byte is received in the coupler from the CDC and is strobed into the least significant eight bits of the buffer. A complete HP word now resides in the buffer and a Flag is sent to the HP signaling its presence and transferring it to the HP data channel. Concurrently a Reply is sent back to the CDC acknowledging the receipt of the second byte. This procedure can now be repeated for the transfer of another two CDC bytes. Synchronization between the two machines is maintained through coupler control over the signals acknowledging the two machines.

At the signal level in the coupler this procedure is executed as follows.

The HP has set SB 10 during the initialization of the transfer. The CDC starts by sending a Write signal followed by the first byte and a Data signal. Write and SB 10 together form WOP, which clears the byte counter to $\overline{A}T$. The first byte is strobed into the buffer by means of STR 1, which is composed of DS, WOP generating BUR 2, and W 1 composed by $\overline{A}T$ and DSW. The HP, which has already set the ENB to declare its readiness to receive data, is not yet involved since only one half of the buffer has been filled. A WREP 2 generates a Reply to the CDC, which drops the DS in turn and resets the WREP 2. The byte counter is switched to A 1 by means of WREP 2. A second byte is now received with a new DS, which together with A 1 forms W 2. DS together with WOP makes BUF 2, which, combined with W 2, strobos the second half of the buffer. The complete HP word now resides in the buffer and can be sent to the HP data channel. This is achieved by BUR 2, A 1, and IPB, which generate IFLB, sending a Flag to the HP data channel. ENB drops, the word is now in the HP and a Reply can be sent to the CDC. Thus IFLB causes a WREP 2 (Reply) which resets the byte counter to $\overline{A}T$. The Data signal (DS) now drops, which in turn makes WREP 2 drop. The resetting of the ENB causes the IPB to drop, which in turn resets IFLB. The HP can now issue another ENB and the CDC can send another byte which repeats the above sequence.

1.3.5.2 Write Mode II

As for Write Mode I, the two machines have to initialize and synchronize before a Write operation can start. The CDC then sends one byte to the coupler where it is disassembled and sent to the HP (see Fig. 1.1). Logically this procedure is as follows.

The HP has set SB 10 during the initialization phase. The CDC starts by sending a Write signal followed by the 12-bit data byte and a Data signal. Write and SB 10 together form WOP, which together with the Data signal form DSW. The data byte is gated into the right positions of the buffer by means of the W 1 and W 4 gate signals and strobed into the buffer by STR 1 and STR 2 which are both present. DSW generates BUR 2, which sets the Input flag (IFLB) if the HP is ready to receive data (IPB true). Now a WREP 2 generates a Reply to the CDC, which drops the DS which in turn resets the WREP 2. Now the loop is closed and as there is a one-to-one relation between bytes and words during Mode II, the byte counter remains in position $\overline{A}T$.

The above sequence is repeated for every byte in the transfer. Bits 15, 14, 7, and 6 are permanently set to zero during these transfers by means of a gate with Mode I.
1.3.6 Functions: Concentrator side

One I/O channel of the Hewlett-Packard is used to control the concentrator side of the coupler and to set and read the coupler status register. Status and control functions are sent from the HP by using an OTA or OTB instruction to this slot followed by an STC instruction. Unless specifically requested by setting the status input function bit (0040000) in the output status, no Flag is set to indicate completion of the operation and the status cannot be read back.

The HP is always able to send functions to the coupler even if the coupler is not connected to the CDC channel. The existing Function codes are as follows:

i) Initialize data I/O slot for output by setting bit 11 in status word (1000000).

ii) Initialize data I/O slot for input by setting bit 10 in status word (0400000).

iii) Selectively set status register to ones for the ones in pattern xxxx (0200xxxx).

iv) Selectively clear status register to zeros for the ones in pattern xxxx (0100xxxx).

v) Send the contents of the status register to the control channel of the concentrator when the channel control bit is set (0040000).

Function codes may be logically combined, e.g.

0600xxxx means select data input and selectively set status register.
0144xx means selectively set and read back status.

1.3.6.1 Functions 1000000 and 0400000

The Hewlett-Packard Microcircuit Interface Cards which are used contain independent input and output registers but use common control signals (Flag and Encode) during both input and output operations. The setting of the status bits allows the coupler to distinguish Input from Output and use the control signals in the right sequence. Furthermore, the status bits are available to the CDC side and cause an Interrupt if selected (see Section 1.3.2.2).

1.3.6.2 Functions 0200 and 0100

The ten least significant bits of the HP word containing these functions are used either to selectively set corresponding bits in the status register (0200) or selectively clear those bits in the status register (0100).

1.3.6.3 Function 004000

This function generates the Flag signal which is sent to the control channel together with the contents of the status register. If the interrupt system on the Hewlett-Packard is enabled, the Flag will cause an interrupt. If the CDC 3000 has executed a selective set function, a Flag signal will be generated as well.

1.3.6.4 Status input codes

There are 12 status bits available to the CDC and 16 available to the HP:

bits 0-9 common bits settable and clearable by HP and CDC
bit 10 Read selected on HP data slot
bit 11 Write selected on HP data slot
bits 12-14 not used
bit 15 parity error detected in the last data transfer.
Whenever a parity error is detected in the concentrator side of the coupler during a block transfer, bit 15 will be set to one.

1.3.7 Parity checking/generation
All Data transfers, Functions and Connect codes from the CDC to the coupler are verified for correct parity. Should a parity error occur during a Write operation, a transmission parity error is flagged to the channel. For a parity error occurring on a Function code, the error is flagged in the same way and the decoded function is not performed. A parity error on a Connect code will only set the indicator on the front panel but will not cause the transmission parity error line to be set.

There is no parity checking between the HP and the coupler. Should, however, a parity error be detected between the CDC and the coupler during a Write procedure, status bit 15 is set to indicate the error. SB 15 is cleared by a read-out of the status register from the HP. Parity is generated on Read data from the coupler to the CDC.

1.3.8 Coupler front panel
The front panel holds a number of light-emitting diodes (LEDs) indicating states of certain important signals or registers in the coupler. It also houses some switches for control and debugging purposes, whose function will be briefly described below.

The "ON LINE/OFF LINE" switch provides a possibility of logically connecting to or disconnecting from the CDC channel. There is a "Master Clear" push-button causing a general reset of all elements in the coupler when actuated.

For debugging purposes, the two switches "AUTO/MAN CLOCK CNTRL" and "MAN. CLOCK" were introduced. The "AUTO/MAN CLOCK CNTRL" stops the central clock, giving timing pulses to the whole coupler. The "MAN. CLOCK" push-button then generates a single clock pulse each time it is pushed.

The device selector switch permits the coupler to be easily reconfigured in a system.

The switch setting determines the device number, which will be compared with the CDC Connect code. The coupler will only respond to Connect attempts from the CDC channel that matches the device number.

1.3.8 Hardware specifications

1.3.9.1 Logic circuits
The logic family used in the coupler is the TTL Texas Instruments 7400 series. The integrated circuits are assembled on standard printed circuit cards, most of which are the Digital Equipment Corporation M series modules. There are also some CERN-developed cards.

1.3.9.2 Transmitters/Receivers for the CDC channel
These are CERN-developed. Both transmitters and receivers employ integrated circuits, each card of which contains four (see Figs. 1.3 and 1.4).
Fig. I.3 CDC 3000 transmitter
The CDC 3000 channel requires bidirectional balanced current drivers\(^5\)). The most appropriate integrated circuit available for this purpose is the DM 8831 tri-state driver of National Semiconductor. In order to create ground symmetric CDC compatible signal levels, the supply voltages are shifted to +2.7 V and -1 V. Consequently, shifting of the levels is necessary for both the Data signal and the Enable signal. The balanced line driver SN 75110 of Texas Instruments is quite appropriate for this purpose. It drives the transmitter in such a manner that it "enables" (low impedance) in the presence of logical ones and "disables" (high impedance) with logical zeros. The official CDC terminators provide the return current in the latter case. As a receiver, the circuit SN 75107 of Texas Instruments has been employed.

1.3.9.3 Packaging

The coupler is built into a 19-inch rack. The basic parts are

i) the two power supplies (+5 V/-5 V);
ii) the crate, wire-wrapped and equipped with logic cards and cables;
iii) front and back panels.
The two power supplies used are made by OLTRONIX, Type C7-10R. They provide power for the logic and for the transmitters and receivers interfacing the CDC channel.

The crate into which the logic cards are plugged is made by the Digital Equipment Corporation and carries type number H 9147). It is six levels high, and five levels are used for logic cards and cables while the sixth level is not used. All connections between cards are made by using wire-wrap techniques.

A front panel provides displays for certain important signals. It also holds the switches that are necessary to control the coupler. Sockets for interfacing the CDC channel are mounted on the back panel and are wired up to the crate.

The coupler has been constructed in the CERN DD Electronics Workshop.

2. THE HP ASYNCHRONOUS TERMINAL INTERFACE

The Hewlett-Packard asynchronous terminal interface card (HPATIC-DD 4270-d, Appendix 2) enables connection of a variety of terminals, using asynchronous transmission techniques, to Hewlett-Packard 2100 series computers8). It provides full duplex transmission of eight data-bit characters (enframed with one Start bit and one or two Stop bits) between the HPATIC and the terminal. Input and output transmission speeds are independently software-selectable, and both necessary clocks derive from a crystal-controlled oscillator, so no frequency adjustment is required. Parity generation/checking is software-selectable, and, when disabled, it allows 8-bit transparent code to be used. The HPATIC provides unpacking and buffering of a 16-bit output word and input buffering of an 8-bit character. It is also possible to output from the computer to the card on an 8-bit character base. A 4-bit status register provides information concerning Input/Output flags, Break detection, and Character lost. An Input flag is set either by a function or on receipt of a character from the terminal. Correspondingly, an Output flag is set by another function or when the output buffer becomes empty after an output operation; both these flags cause an HP flag and also an Interrupt in case of the interrupt system being enabled. The Flag and Interrupt logic is identical to the standard HP philosophy.

2.1 Control functions

When the HPATIC is set to the 8-bit mode (initial condition), any output instruction with bits 15 or 8 (or both) present is interpreted as a control function. Bit 15 indicates that the upper half of the word carries the control function while bit 8 indicates the lower half as follows:

0 BIT 15 - Upper byte function indicator
  14 - Disable input
  13 - Enable input
  12 - Select character unpacking, 16-bit Output mode

0 BIT 11 - Set Input flag
  10 - Set Output flag
  9 - Disconnect terminal
O BIT 8 - Lower byte function indicator
7 - Enable Output parity
6 - Enable Input parity
5 - Select even parity
4 - Select output speed
3-0 - 4-bit speed selection code (see Table I.1).

<table>
<thead>
<tr>
<th>Transmission speed in bps (decimal)</th>
<th>Input selection code (octal)</th>
<th>Output selection code (octal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>300</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>600</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>1200</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>1800</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>2400</td>
<td>7</td>
<td>27</td>
</tr>
<tr>
<td>4800</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>9600</td>
<td>11</td>
<td>31</td>
</tr>
</tbody>
</table>

Notes: Input selection code 0 or output selection code 20, does not change transmission speed.

For bits 5, 6, 7 when zero: reversed interpretation.

The Disconnect Terminal function is sent to the terminal via the 48-pin edge connector and is intended to be used in connection with an automatic exchange system.

2.2 Status

A 4-bit status register provides the following information:

IBIT 11 - Input flag
10 - Output flag
9 - Break
8 - Character Lost

It can be accessed at any time by executing an LIA/B instruction addressing the card.

IBIT 11 is set either by issuing a control function (OBIT 11) or on receipt of a character in the input buffer. Similarly IBIT 10 is set by a control function (OBIT 10) or when the output buffer has been emptied.

The absence of a Stop bit following the character or, when connected to a Modem, a drop of Carrier Detected causes a Break condition, which sets IBIT 9 and an Input flag with IBIT 11. IBIT 9 is reset and IBIT 11 set when Carrier Detected reappears, or when not
connected to a Modem on receipt of a new character followed by a Stop bit. When the terminal has sent two or more characters to the input buffer in succession without an intervening read-out of the buffer contents by use of a LIA/B instruction, a Character Lost indicator is set (IBIT 8).

IBITs 8, 10, and 11 are reset at the end of a read-out.

2.3 Data output procedure

The HPATIC accepts data from the computer in two different modes: namely, the "8-bit mode" or the "16-bit mode". When in the 8-bit mode, which is selected after a preset or by issuing a CLC instruction to the card, the lower byte of the HP word (DB 0-7) is transmitted to the terminal. The 16-bit mode is selected by a control function, which enables the card to receive 16-bit words from the computer. Each 16-bit word is automatically disassembled into two 8-bit characters, which are subsequently sent to the terminal. Since these two modes imply a different part of the control logic of the card, each of them will be discussed separately.

2.3.1 The 8-bit mode operation

When switching power on to the computer, the POP10 signal resets the mode FF thus setting the card to 8-bit mode operation. The card now accepts data from the computer (DB 0-7) on a character basis, thus ignoring the top byte of the HP word (DB 8-15). Control functions can also be sent to the card.

The main part of the output logic is the terminal transmitter, which is an integrated MOS circuit Motorola MC 2257L. The 8-bit character from the computer is sent to the card by an OTA/B instruction. It arrives in an external auxiliary buffer and, if the internal buffer of the terminal transmitter is empty, the character is copied into it, using the Load Strobe signal. A few microseconds later, the command System Ready is sent, starting the serializing of the character at a speed determined by the applied clock. At this time the internal buffer contents are copied into the shift register thus causing the O/P Buffer Empty to be set. When copying the auxiliary buffer contents into the internal buffer, status bit 10 is set and a Flag (and an Interrupt if enabled) is sent to the computer indicating that a new character can be sent. In the same way, this new character is first stored in the auxiliary buffer then copied into the internal buffer, setting a status bit 10 and a Flag/Interrupt, and is finally transmitted serially to the terminal.

The auxiliary buffer was introduced in order to cope with the relatively slow MOS terminal transmitter. Since it needs the data present at its inputs for a minimum time of 750 nsec it was not possible to load it directly from the computer and therefore the need for an auxiliary buffer arose. For the same reason the POP10 signal (used as MC for the chip) had to be extended to 2 μsec.

2.3.2 The 16-bit mode operation

The 16-bit mode operation is selected by the computer sending a control function to the HPATIC. Any subsequent outputs from the computer will now be interpreted as 16-bit Data words and are disassembled into two 8-bit characters by the HPATIC logic. The upper part of the 16-bit word (DB 8-15) is first transmitted to the terminal, followed by the lower part (DB 0-7). The detailed procedure is as follows.
A 16-bit word is sent to the HPATIC, which is already set to "16-bit mode". The word is stored in the 16-bit auxiliary buffer (of which eight bits are used in the 8-bit mode) and the upper eight bits of the word are loaded into the internal buffer of the terminal transmitter using the Load Strobe signal. A few microseconds later the System Ready signal is applied, thus copying the internal buffer contents into the shift register and starting serialization. When the internal buffer gets free (Buffer Empty signal), the lower part of the Data word (DB 0-7) is copied from the auxiliary buffer into the internal buffer of the terminal transmitter, serialized, and sent to the terminal. At this time the external auxiliary buffer is empty and therefore a Flag/Interrupt and status bit 10 are set indicating to the computer that another output could take place.

In order to reset the HPATIC to the 8-bit mode operation, a CLC instruction is required. Once reset, the card will accept 8-bit data characters and control functions.

The terminal transmitter can optionally generate even or odd parity by issuing appropriate control functions. When parity generation is disabled, the terminal transmitter will allow for transparent transmission of 8-bit characters, and when enabled, parity is generated on the seven significant bits (DB 0-6), adding a parity bit in DB 7 position.

The HPATIC offers a hardware-selectable number of Stop bits. By throwing a switch on the card, one or two Stop bits can be obtained.

The clock being used has 16 times higher frequency than the bit transmission speed, selected by software.

Note that no STS instruction is required to perform an output operation, and its use is only to enable the interrupt logic on the card.

2.4 Data input procedure

The HPATIC accepts from the terminal asynchronous 8-bit serial characters enframed by one Start bit and one or more Stop bits. These characters are staticized and then forwarded to the computer by using the lower part of the HP word (DB 0-7). Also related to the input procedure are three status bits indicating Input Flag, Break condition, and Character Lost.

The main part of the serial-to-parallel-conversion logic is comprised in the terminal receiver integrated MOS circuit M72259 L, as well as an 8-bit character buffer. Serial data enter the terminal receiver, and the internal clock logic synchronizes the internal clock to the data. Upon receipt of the whole character in the shift register, its contents are transferred to the internal buffer and the Buffer Full signal is set, which in turn sets Data Strobe RX and status bit 11. The Data Strobe RX signal enables read-out of the internal buffer, whose contents are transferred to the computer by an LIA/B instruction, resetting the Data Strobe RX. The Buffer Full is reset when applying Data Strobe RX. The terminal receiver provides optional odd or even parity checking, and a detected error is indicated by setting data bit 7. Should two or more characters arrive in succession without intervening read-out, a Character Lost signal is sent to the computer (IBIT 8). This signal is reset by an LIA/B instruction.

The absence of a Stop bit causes the Break FF to be set, which is indicated to the computer as an IBIT 9. A Break condition can also occur when the HPATIC is operating with a CERN-CODEC and the connection HPATIC-CODEC is interrupted. The Break remains in that
case until the connection has been re-established. The detection of either the presence or absence of a Break causes an Input flag (IBIT 11) to be set.

When parity checking is disabled, the terminal receiver allows for reception of 8-bit transparent codes.

A Master Clear pulse is applied to the receiver for a duration of approximately 2 μsec when powering on the computer. Disabling input to the receiver causes the Master Clear to be continuously applied, thus preventing any activity around the terminal receiver.

Owing to a design peculiarity of the MOS circuit, the clock is only applied to the circuit from the moment that a Start bit is detected until the staticized character has been read out by the computer.

2.5 Clocks

A 2.4576 MHz crystal-controlled oscillator provides the necessary timing information for both the transmitting and the receiving logic. Since transmitting and receiving speeds can be independently selected, it was necessary to duplicate the divider logic. Nine possible speeds are possible, and seven of those are derived by binary division of the main frequency (2.4576 MHz). The two remaining speeds (110 bps and 1800 bps) required extra logic for decoding and resetting of the counter since they were not obtainable by straight binary division. All nine frequencies are connected to a 16 input cata selector. By four control bits any of these nine frequencies can be selected according to Table I.1. The selected frequency is available at the data selector output and is 16 times higher than the bit speed. The frequency obtained for the output speed is applied to the terminal transmitter, while the input frequency is supplied to the terminal receiver. In both transmitter and receiver a division by 16 takes place, adopting the respective internal clocks to correspond to the chosen bit speed.

2.6 Interface circuits

The HPATIC provides three different interface circuits for the terminal connection. It allows connection of CODECs or local terminals via a CCITT receiver/transmitter (MC 1488/1489), and special provision has been made to facilitate connection of remote terminals via a balanced current receiver/transmitter (DM 8820/8830). A third option is the TTL receiver/transmitter, where the transmitter is a power gate (SN 7437) with an external pull-up resistor to \( +V_{CC} \), and the receiver an inverter (SN 7404) with line terminators at its inputs. In order to select the desired interface circuit, it is necessary to throw a switch on the card receiver whereby the three different interface drivers are connected to different pins on the 48-pin edge connector and therefore the selection is obtained through the wiring of the attached external cable.

2.7 Switch settings

As has been previously discussed, a certain number of options are selectable by throwing corresponding switches, located at the upper right corner of the HPATIC. A brief description of their significance is given below.
<table>
<thead>
<tr>
<th>Switch name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTL</td>
<td>Selects TTL driver/receiver as terminal interface circuits.</td>
</tr>
<tr>
<td>DM</td>
<td>Selects DM 8830/20 twisted pair driver/receiver as terminal interface circuits.</td>
</tr>
<tr>
<td>CCITT</td>
<td>Selects CCITT driver/receiver as terminal interface circuits.</td>
</tr>
<tr>
<td>MDEM and MDEM</td>
<td>When MDEM true and MDEM false, the connection HPATIC-terminal should be established via a Modem or a CODEC. For all other cases MDEM should be true and MDEM false.</td>
</tr>
<tr>
<td>1 STOP</td>
<td>Provides one Stop bit on transmitted data.</td>
</tr>
<tr>
<td>2 STOP</td>
<td>Provides two Stop bits on transmitted data.</td>
</tr>
</tbody>
</table>

The following rules must be observed in order to ensure proper functioning of the selected options:

a) Only one switch of the three TTL, DM, CCITT switches may be selected at one time.
b) Only one switch of the two 1 STOP, 2 STOP switches may be selected at one time.
c) Switches MDEM and MDEM must always be in complementary states.

3. THE COMMUNICATION CONTROLLER

3.1 Its function

With one HP 2100 computer responsible for the concentration of at least 36 terminals (if no channel sharing is applied), it was felt justified to install a second HP 2100 computer as a back-up facility. In connecting both machines to the system it is possible to share the load by distributing the traffic on both machines (Fig. 1.5). Initially this is done by a manually operated circuit switch (Fig. 1.6). In case one of the HPs or one of the couplers should be down, all traffic has to be routed through the other machine. The circuit switch has of course to communicate with all HPATIC line controllers in both HP machines. The appropriate way to do this is through balanced current, twisted pair connections. Where the HPATIC line controllers themselves have been equipped with such transmitters and receivers, compatible units have been mounted within the circuit switch. Switching is done on TTL levels, but "site" transmission (with distances of up to 5 km!) of course is not. This latter item requires long-distance, balanced current transmission techniques. The transmitters and receivers used are similar to the ones used in CERN CODECs 9). With bit speeds up to 9600 bps no special line-compensating filters will be required. The actual Communication controller (COMMCON) consists of all these elements; so it has basically two functions. The first one is to ensure a quick switch-over of terminals from one HP computer to the other, and the second is to provide long-line transmission capability.
Fig. I.5 Patching scheme for SUPERMIX terminals
3.2 System hardware

The basic elements of COMCON are:

i) the power supplies;
ii) the DEC crate with associated circuitry;
iii) the front panel.

The power supplies provide power for the circuitry and the light-emitting diodes (LEDs) on the front panel. One power supply delivers +5 V and its capacity is 10 A. The other delivers -5 V and its capacity is 5 A. The +5 V is used for the transmitters/receivers and the LEDs, while the long-line receivers require -5 V.
The basic circuitry required for one terminal-to-computer switch is as follows.

Data is received in the COMCON from a terminal in serial form and, depending on the front panel switch setting for that particular terminal, a transparent path to the selected computer is established. Conversely, data from the selected computer is transmitted through the COMCON to the terminal.

The front panel holds the switches and the LEDs. For each terminal connection there is one toggle switch determining the Host computer and four LEDs, two of which are used to indicate the selected Host computer and the other two for monitoring data to and from the terminal.

4. DATA SETS

As seen in the section on the HPATIC hardware, three transmission techniques are available, selectable by microswitch (CCITT V24, balanced current, and TTL); the CCITT V24 driver permits transmission distances of up to 15 metres. It has to be emphasized that Modem connections have to be of full duplex nature, because HPATICs do not generate a transmission clock. The two control signals needed for a full duplex connection have been provided. The V24 signal (Carrier Detected) has to be connected to pin E of the HPATIC edge connector. Its appearance as well as its disappearance will cause Interrupts in the HP computer. The V24 signal (Data Terminal Ready) has to be present in order to keep the data set connected (Data Set Ready). In using the signal at pin T of the HPATIC edge connector, disconnection occurs with the 700 nsec pulse generated with function 9. Private, non-PTT cables are used in most cases of "on-site" terminal connections. Using the balanced current drivers on the HPATIC for distances up to some hundred metres, more powerful balanced drivers are needed for distances up to 10 km. In principle, a pair of data sets are required. These so-called CODECs are base-band, Modem-compatible units equipped with integrated circuit drivers and receivers. The concept of the CODECs is fairly modular. Every CODEC has space for extensions, but in this case very simplified versions have been used.

The central CODECs are mounted within the COMCON, and translate twisted pairs into TTL levels and vice versa. The remote CODECs translate balanced current signals into CCITT V24 signals. Control signals such as Clear to Send and Carrier Detected are returned on the Request to Send and Data Terminal Ready signals, respectively. The Data Set Ready signal is generated as soon as the power is switched on. As integrated circuit transmitter, the DM 8830 of National Semiconductor has been chosen. For the long distances it is used in a rather unusual fashion. Instead of 0 V and +5 V, -5 V and +3 V have been taken as power voltages.

The two opposite data streams are decoded into TTL levels by means of integrated circuit comparators SN 75107 of Texas Instruments. For the short distances, receivers DM 8820 of National Semiconductor have been taken. Receivers have been provided with space for resistance networks or RC filters. Next to reducing the RC factor of the lines by a factor of about 10, they help to increase considerably the common mode noise immunity of the receivers (Fig. 1.7).
Fig. I.7 Long-line receivers
CHAPTER II

SOFTWARE OVERVIEW

Currently the SUPERMUX concentrator is only interfaced to CDC 6000 computers, although other Host computers are feasible.

Thus the software is divided into the HP 2100 concentrator software, the CDC 6000 PP driver and its associated modifications to the CDC 6000 series interactive system INTERCOM. Furthermore, there is a sizeable packet of software preparation and diagnostic routines.

The following sections will be treated:

1) Host-concentrator protocol
2) HP concentrator software
3) The PP driver software
4) Software preparation
5) User interface

1. HOST-CONCENTRATOR PROTOCOL

Communication between the HP concentrator and its multiple Hosts takes place via parallel couplers. However, the communication protocol is designed to be easily adapted to work via a serial communication link. The Host in all transfers acts as a master, while the HP concentrator can merely ask for attention by setting a bit in the coupler status register.

A Host-concentrator transfer consists of a control header usually followed by a data block. Data is coded in 8-bit or 6-bit mode, and the blocks are variable in size up to a fixed maximum of 150 characters.

Transfer format:

<table>
<thead>
<tr>
<th>HEADER</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td></td>
</tr>
<tr>
<td>SP</td>
<td></td>
</tr>
<tr>
<td>SA</td>
<td></td>
</tr>
<tr>
<td>H:SN</td>
<td></td>
</tr>
<tr>
<td>CF</td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td></td>
</tr>
</tbody>
</table>

DP - Destination Processor
DA - Destination Address
SP - Source Processor
SA - Source Address
HT - Header Type
SN - Stream No.
CF - Control Field
BL - Data Block Length
Note that the stream number field allows the protocol to be expanded for consoles with several Input/Output streams.

The different header types and Control Field codes allow for Host-concentrator table co-ordination and recovery.

Consoles attached to the concentrator are initially considered to be in a disconnected state. In this state a console user may type a special control sequence to indicate to which Host he wishes to be connected, or he may commence normal input directly, in which case the concentrator chooses a Host as default. In either case the concentrator sends a Connect request transfer to the Host indicating the physical console which wishes to connect; and the Host may then reply with a Connect transfer indicating the time-sharing port allocated to the console, or with a Disconnect transfer indicating that no ports are available. A console remains logically connected to a given Host until a Disconnect is generated by the Host (e.g. at log-out time), or by the user (e.g. after a Host failure), or by the concentrator (e.g. after a console line failure).

All stages of the Host-concentrator communication protocol are carefully checked and timed out. If a Host ceases to respond for a certain time, the concentrator places it in a suspended state and sends a warning message to all consoles connected to that Host. If a Host goes down, it may, after restarting, interrogate the concentrator for details of all consoles logically connected to the Host. Likewise if the concentrator goes down, it may be reloaded and it will then interrogate each Host and re-establish all console connect conditions known to the Hosts.

The concentrator and its Hosts are thus very loosely coupled together. Hosts may be dynamically put in and out of service without disturbing the concentrator, and all consoles are able to communicate with any active Host.

2. HP CONCENTRATOR SOFTWARE

2.1 Load handling

The concentrator handles asynchronous consoles only. This is because a load of 36 consoles, each operating at 9600 bps in character interrupt mode, would exceed the processor capacity of the HP 2100. However, for asynchronous keyboard consoles, only the time critical transmission direction is input, and since this is usually generated by humans the rate is generally low. Also, the use of a microcoded interrupt handler (see below) enables moderate amounts of high-speed input to be handled without loss of data. On the other hand, should all consoles be receiving output simultaneously (a most unusual situation), the HP 2100 would saturate; but because the transmission is asynchronous, each console would see only a slow-down of its output with no data lost. The current version of the system can output simultaneously to about 13 consoles at 9600 bps before any slowing down occurs.

2.2 Console control

Asynchronous consoles currently marketed generally operate using a form of the ASCII standard code. However, they vary widely in other respects. A flexible console control system must take account of such varying factors as I/O speed, line and screen size, new line control and timing, new page or screen control and timing, etc. Each console connected
to the concentrator is defined by a parameter table which identifies these varying console characteristics and also defines current console operating conditions. A partial list of these parameters is as follows:

- Console type (Teletype, T4002, T4010, T4012, T4023, VISTAR, etc.)
- Character echo Enable/Disable
- Number of characters in a line
- Number of lines in a page (if set = 0, page size is infinite)
- Erase Screen procedure index
- New Line procedure index
- End of Page procedure index
- Input character indicating End of Input record
- Input character to delete last character input
- Input character to delete current input record
- Input character to interrupt current console activity
- Horizontal tab character
- Horizontal tab settings
- Character echoed on receipt of Delete Last Character input
- Character echoed on receipt of Delete Current Input record
- Free code End of Input record delimiter
- Free code Maximum Input record length
- Select input speed
- Select output speed

Special procedures for End of Line, End of Page, etc., are identified by indices which point to algorithms in the code for handling the procedure.

All of the console control parameters may be individually modified either by a specially formatted data block from a connected Host or Host user program, or by a special input control sequence typed by the user. This permits the system to dynamically adapt itself when different types of consoles connect to the same concentrator port (e.g. by dialling in), and also permits the user to change the current operating characteristics of a console to suit his own tastes.

The concentrator provides three types of I/O character code handling dynamically selectable by the Hosts (see User Interface). There are two coded modes in which the concentrator recognizes and acts on the control characters (End of Record, Erase Line, etc.) defined for the console:

a) formatted mode, in which non-control characters are passed unchanged between console and Host;

b) ASCII-to-display-code conversion mode, in which alphanumeric characters are converted between the CDC 6000 series internal code (display code) and ASCII.

The third mode is Free code mode, which provides for transmission of 8-bit characters completely untouched between console and Host. End of Record conditions on input are recognized in this mode either by an optional user-specified End of Record character or by a Time Out or Record Full condition.
2.3 The system control tables

The system control tables contain the current status and equipment parameter information for each Host and each console. These tables are initially generated at system preparation time by a set of macro calls, and they are subsequently updated by the system drivers to record the current operating status of each piece of equipment for use each time the equipment is serviced.

2.4 System configuration

The HP 2100 is configured so that the console interfaces occupy the 36 highest peripheral priority levels. Interrupts on these levels are handled exclusively by microcode.

The interval clock handler, the Host interface driver, and the console driver are associated with lower interrupt levels. Except during Restart, interrupts are never disabled for more than a few instructions at a time, and access to the system control tables by driver modules operating on different interrupt levels is interlocked by forcing these modules to operate when necessary at a common software level lying between their own levels and the console interrupt levels. Console interrupts can therefore continue to be serviced even while the driver modules are active.

2.5 The microcode

Special microcode is used primarily to handle console interrupts, and all such interrupts cause direct entry into a microcode routine\(^19\). Input interrupts occur when a character has been received by the console I/O card, and the microcode combines the input character with a console identifier, stores it in a circular buffer, and generates a software interrupt to activate the console driver to process the buffer. Since the consoles operate on the highest interrupt levels, console interrupts may interrupt the console driver, and as the microcode takes about 11 \(\mu\)sec to process an interrupt, the system can handle bursts of high-speed input without loss of data so long as the circular buffer does not overflow (Fig. II.1).

Output interrupts occur after a one- or two-character output is complete, and they cause the microcode to set a flag for that console into a separate circular buffer and to activate the console driver to start a new output operation.

Special microcode is also used to extend the basic HP 2100 instruction set with instructions for byte and bit handling and for economically accessing the system control tables.

2.6 The console driver

The console driver has its own software interrupt level. It is activated by an interrupt set by the console interrupt microcode, and when active it scans the input and output console interrupt circular buffers, moving input characters to buffers associated with each console and starting a new output operation when an output interrupt indicates that a previous operation is complete. All console input line-editing, output formatting, and code translation are done in this driver. On receipt of an input End of Record or when an output buffer becomes empty, it sets flags in the appropriate system control table to initiate a new transfer with the Host.
Fig. II.1 Hardware-software priority levels
2.7 The Host driver

The Host driver carries out all control and data transfers between the concentrator and its Hosts, together with the associated error checking, recovery, and control procedures. It is activated by interrupts from the Hosts.

2.8 The interval timer handler

This handler is activated every 220 msec by an interrupt from the interval timer. It scans the Host and console control tables to determine if any activity has not been completed within a reasonable time, and to provide the timing necessary for special console procedures such as Screen Erase, End of Page, or End of Line.

2.9 The buffer pool

Memory not occupied by code or tables is kept as a pool of fixed-length buffers 80 words long. Buffers are chained by a simple forward link pointer located in the first word of each buffer. Free buffers are kept on one chain, and input and output buffers associated with each console are kept on separate chains for each console. A buffer is permanently allocated to each Host so that control messages may always be transferred. The number of input or output buffers that can be allocated at any one time to any console is limited (initially to two) to prevent flooding of the buffer pool. The buffer pool currently occupies about 9K of the 16K memory.

3. THE PP DRIVER SOFTWARE

The software interface into the CDC 6000 consists mainly of a non-standard INTERCOM driver (1HZ) executing in a CDC 6000 series PPU. It is conceptually somewhat similar to the Control Data LCC INTERCOM driver. The driver word in the LCC User Table and the Multiplexor Subtable was adapted to the needs of 1HZ, but otherwise the modifications to INTERCOM were kept to a minimum.

3.1 1HZ driver call

The PP routine 0HZ is called by 111 when the INTERCOM subsystem is brought up by the operator. 0HZ formats 12Z's input register as follows:

\[
\begin{array}{cccccc}
D.PPIR & & & & & \\
\hline
1 & H & Z & 0 & 0 & C & C & N & N & 0 & 0 & D & D & 0 & 0 & 0 \\
\hline
D.PPMES4 & & & & & \\
\hline
EST ORD. & EQ NO. & MIX SUBADDR & 0 & 0 & 0 & 0 & HP NO. \\
\hline
D.PPMES5 & & & & & \\
\hline
" & " & " & " & " & " & " & " \\
\hline
D.PPMES6 & & & & & \\
\hline
" & " & " & " & " & " & " & " \\
\end{array}
\]

where CC = Channel No. 
NN = Driver ordinal
ZZ = No. of HZ's described in message buffers
3.2 1Hz structure

The entire driver has been rigorously implemented to be table-driven. The state of
the driver at any moment in time is described by its HP states, line states, stream states,
and transfer states. Actions are performed during state transitions which are triggered by
external or internal responses. In this way the code becomes extremely modular.

The main loop of the driver scans external responses, and all HP, line, and stream
states for work, and initiates state transitions accordingly. During execution of this
main loop a cell called ACTIVITY is maintained which corresponds to the current driver load.
This cell determines at the end of the main loop if the driver enters Recall or not. Essential-
ly, in case of no active lines the driver will enter Long Recall (∼ 1 second), and in
case of no active data transfers Short Recall (∼ 1/4 second); otherwise the driver remains
resident in the PPU. The object of this is to minimize PPU occupancy while maintaining the
maximum SUPERMIX response throughout.

- Response actions

Responses can be grouped into two classes: headers coming from the SUPERMIX concen-
trator, and internally generated responses. Both types are treated in an identical way by
the software.

All possible responses are defined in the driver's Response Table. Each response con-
tains two pieces of information: first, the address of a Response action to be executed;
and second, an internal Response code which is used along with the state to describe a state
transition. (Note: Not all responses trigger a state transition.)

The macro XRES is executed for a response which makes a call to the procedure DORACT.
This picks up Response action and internal Response code and enters the action with a
direct jump. All actions return via DORACT.

- HP states

HP states are controlled by a full two-dimensional State Table. The actions referenced
by this table are mainly concerned with driver and SUPERMIX recovery and deadstart.

- Line states

Line states are controlled by a full two-dimensional State Table. This governs all
line-oriented actions such as Line Connect/Disconnect and User Table requests.

- Stream states

Streams are also controlled by a full two-dimensional State Table. Currently, only
the TTY (Terminal) State Tables are implemented, since SUPERMIX does not support remote
batch stations. There are separate State Tables for Input and Output at a TTY since these
are considered as two different streams with stream numbers 0 and 1. The stream actions
provide for the transfer of data between the INTERCOM buffers and the PP I/O buffer, and
perform the switching between the DISPLAY and ASCII code 1/0.

- Transfer states

The transfers between the SUPERMIX controller and the driver are controlled by a two-
dimensional State Table. While the transfer states do the polling of the coupler status
register of SUPERMIX, the transfer actions send commands to SUPERMIX and perform the actual
data transfer. All steps of a transfer are carefully timed out to achieve resynchronization between SUPERMUX and the driver in case of spurious hardware malfunctions.

3.3 HZ INTERCOM interface

HZ employs the usual interactions with INTERCOM, mainly: get and release interlocks, request and return buffers and User Tables. The User Table driver word has been modified and looks like the following:

<table>
<thead>
<tr>
<th>Word</th>
<th>Byte</th>
<th>Bits</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>W.UHDRV1</td>
<td>C.1ST</td>
<td>0-5</td>
<td>Interactive input stream state</td>
</tr>
<tr>
<td>(11)</td>
<td>(0)</td>
<td>6-11</td>
<td>Interactive output stream state</td>
</tr>
<tr>
<td>C.IDCNT</td>
<td>(1)</td>
<td>11</td>
<td>= 0 current input line had CR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 current input line had LF</td>
</tr>
<tr>
<td>C.IDFNIN</td>
<td>(2)</td>
<td>0-17</td>
<td>IN pointer for previous input line with LF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C.IBPOS</td>
<td>(2)</td>
<td>6-8</td>
<td>Byte position for previous input with LF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>= 0 current Output Mode DISPLAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 current Output Mode ASCII</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>= 0 NO EOR at end of last output transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 EOR at end of last output transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>= 1 Teletype output was interrupted</td>
</tr>
<tr>
<td>C.INERR</td>
<td>(4)</td>
<td>0-11</td>
<td>User Table Error Count</td>
</tr>
</tbody>
</table>

The HZ Multiplexor Subtable has been modified as well:
Table 11.2
HZ Multiplexor Subtable

<table>
<thead>
<tr>
<th>HEADER</th>
<th></th>
<th></th>
<th>No. of lines +1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP state</td>
<td>HP error count</td>
<td>HLP number</td>
<td></td>
</tr>
<tr>
<td>First term ID allocated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line state</td>
<td>User Table address</td>
<td>HP log. port number</td>
<td>Terminal type</td>
</tr>
<tr>
<td>LINE N</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The HLP number is communicated to the driver at SUPERJOB deadstart time, and from then on serves as a unique identifier between the driver and SUPERJOB during header transfers.

The HP logical port number is communicated to the driver at connect time of the terminal and serves as an identifier during header transfers as well.

4. SOFTWARE PREPARATION

The HP 2100 software is written in a dialect of HP assembly language, implemented on the CDC 6000 and 7000 series machines. Both normal code and microcode assemblers have been developed. The technique follows the methods of Jeavons\textsuperscript{11}). Each HP 2100 instruction (or micro-instruction) mnemonic is defined as a macro for the CDC COMPASS assembler, and produces the appropriate HP 2100 16-bit (or 24-bit) binary pattern, embedded in a CDC 60-bit word\textsuperscript{12}). Extraction of the HP bit fields from the CDC words is made later during a loading pass. This is elementary in the present application, as both HP assemblers described here produce only absolute code, but linking and production of relocatable code is possible (and has been performed using the CDC SCOPE loader in other applications). The resulting HP code is punched out as an absolute binary deck, formatted for direct input to the HP 2100 under control of a standard HP absolute binary loader. Once the HP code is loaded in this way, the microcode binary deck can be loaded into the Writeable Control Store (WCS) under control of a (non-microcoded) portion of the formerly loaded code. This latter step can be eliminated by incorporating the microcode into a Read-Only Memory (ROM) at some convenient time, using standard HP 2100 techniques.

Use of this "cross-assembler" technique confers several important advantages:

1) A large computer system gives high assembly speed, a convenient file base for storing and updating source code, and powerful peripherals for printing and punching results.

2) The "target" HP computer is released for its production function; furthermore, its configuration need not reflect the needs of software development, e.g. no disk need be fitted, no paper tape or card punch need be provided.
3) The very powerful facilities of COMPASS, such as macro and micro expansion, conditional assembly, and powerful cross-referencing, are not commonly available in minicomputer assemblers. The macro facility is particularly heavily used in SUPERMIX for reconfiguring and generating the system. For example, additional consoles, new console types, or a new Host computer may be added to the system at assembly time by inclusion of single macro calls; all associated tables, interrupt linkages, etc., are then generated automatically.

The only significant disadvantage incurred by the cross-assembler is that COMPASS's rules for formatting and syntax of input code must be observed, and this becomes reflected in the dialect of HP code which is accepted. In COMPASS, the fields in an assembly statement are all separated by blanks, and certain limitations exist on the use of these fields. Consequently:

1) The location symbol field, if present, must start in column 1 or 2 and be terminated by one or more blanks.

2) The maximum symbol size is eight characters.

3) The opcode field must always be present and be delimited by blanks with no embedded blanks. It cannot start in column 1 or 2.

4) The operand (address) field is always separated from the opcode by blanks, cannot contain embedded blanks, and must start before column 30 (or it will be treated as a comment).

5) The comments field is separated from the operand field by blanks. Entries starting after column 30, or following a * in column 1, are treated as comments.

6) If an instruction has multiple address symbols, they must not be separated by blanks, and must be bracketed ( ). The brackets aid assembler efficiency.

7) COMPASS requires IDENT and END cards at the beginning and end of routines.

8) COMPASS's rules for the evaluation of address expressions must be observed.

9) COMPASS's reserved words and characters (e.g. &,-,"') must be allowed for.

10) Only the first 72 characters of a card are assembled. Extension cards contain , in column 1.

The HP assembler dialect also incorporates the microcoded extensions made to the HP 2100 instruction repertoire. These improve byte and bit handling, access to the system control tables, and driving the CERN-designed HPATIC console I/O card. They have the following assembler mnemonics and characteristics:
General I/O

STFR → Normal HP STF
CLFR → " CLF
SFCR → " SFC
SFSR → " SFS
LIAR → " LIA
OTAR → " OTA
STCR → " STC
CLCR → " CLC
LIACR → Normal HP LIA, C
OTACR → " OTA, C
STCCR → " STC, C
CLOCR → " CLC, C

with channel select code from bits 0-5 of B register,
bits 6-15 of B should be 0

General loads and stores

LDAI → Load A from address following instruction, indexed by B
LDBI → Load B from " " " " " A
STAI → Store A to " " " " " B
STBI → Store B to " " " " " A
LDRL → Load A from address in bits 0-4 of instruction, indexed by B
STRL → Store A to " " " " " B

Byte addressing

LDRLB → Load bits 0-7 of A with byte whose address is sum of word address in B and
byte offset in bits 0-3 of the instruction.
STRLB → Store bits 0-7 of A in the byte whose address is sum of word address in B
and byte offset in bits 0-3 of the instruction.
LBYTE → Load bits 0-7 of A with byte whose byte address is in B.
SBYTE → Store bits 0-7 of A into " " " " "
LBYTEI → LBYTE, with B incremented after instruction.
SBYTEI → SBYTE, " " " " "

Bit testing

BCHK Tests A and skips to instruction at: P + 1 if A = 0
P + 2 if bit 15 of A = 1
P + 3 if bit 14 of A = 1, etc.
SBS Tests A and skips next instruction if the bit in A specified in bits 0-3
of the SBS instruction is set (= 1).
SBK Tests A as above, and skips if the bit in A is clear (= 0).

HPATIC I/O interrupt handling

This microcode handles an interrupt from an HP asynch I/O card. It does an input from
the card and determines if it is an input or an Output interrupt (or both).

For Output interrupts it does a table look-up to determine the logical device number
from the I/O address and then stores an indicator in a circular buffer reserved for Output
interrupts. The indicator is bits (10-15) = device logical number. After storing, it
updates and stores the In pointer of the circular buffer. Since only one Output interrupt
can be set on a device at any time, the output code does not check for buffer overflow.

For Input interrupts it does a table look-up to determine the logical device number
from the I/O address and then stores an indicator in a circular buffer reserved for Input
interrupts. The indicator is

bits (10-15) = device logical number
bit 9  = Break flag
bit 8  = Character lost flag
bit 7  = parity error (if parity checked)
bits (0-6) = input character.

The circular buffer In pointer is updated and a test (In = Out?) is made for buffer
overflow. If overflow is detected the current In pointer is stored in another buffer
(indexed by device logical number - address of buffer is in location 120) as a Non-Zero
flag to indicate overflow has occurred and where it has occurred. This flag is stored only
if the buffer location is zero so as to indicate clearly the first occurrence of overflow.
A global flag is also set to indicate that an entry has been made in the overflow table and
the In pointer is not updated.

If the word input from the I/O card contains no Input or Output interrupt indicator,
the interrupt is assumed to be an Output one.

The microcode assumes the following fixed locations:

110 = address of Start of Output interrupt circular buffer
111 = In pointer
112 = Out pointer
113 = address of limit
114 = address of Start of Input interrupt circular buffer
115 = In pointer
116 = Out pointer
117 = address of limit
120 = address of overflow buffer
121 = Flag - overflow buffer not empty
122 = STF instruction to set interrupt to console work level
130 onward = buffer indexed by I/O address containing device logical number in
bits (10-15) of each word. This buffer has a logical Start address
at 120 since I/O address 0-7 cannot be consoles.

For all interrupts a Flag is set to force an interrupt to the interrupt level where
circular buffers are processed.

5. USER PROGRAM INTERFACE

SUPERMIX allows for transmission in Display, ASCll, and Free code for both Input and
Output. While a terminal works in Display or ASCII code, SUPERMIX performs terminal
specific screen formatting and interprets the first data character of a record as Format
Control (FC). During Free code no screen formatting is done; this code is used for binary
transmission (e.g. graphics).
5.1 Display code records

This is standard INTERCOM I/O. Interactive programs use normally coded Read/Write statements on a connected file when it communicates with a terminal. Each record ends with a 12-bit zero byte at the end of a 60-bit central memory (CM) word with blank padding if necessary. For a FORTRAN program the padding on zero byte is added by the run-time I/O routines. The first 6-bit character of the record is taken as the Format Control character.

```
FC d d d 0 0 0 0
```

DISPLAY CODE RECORD 60-bit word boundary

5.2 ASCII code records

The full ASCII character set can be communicated via SUPERMIX to/from the terminal. Data in the CDC 6000 is packed as eight data bits in a 12-bit byte. The record starts off with a 12-bit zero byte which signals a Non-Display code record. The second 12-bit byte contains the Format Control character. The end of a record is signalled by a Free code Terminator Byte (FTB) in which bits 8 and 9 are set to one. The bits 0 and 7 select the code of the next input record. FTBs are used to pad to a 60-bit word boundary and, for compatibility with Display code records, a full 60-bit zero word finally ends the ASCII record. Note that the record formatting is entirely the user's responsibility.

```
0 0 0 0 0 Format Control d d d
  d d d d d d
  d d d FTB FTB
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

5.3 Free code records

Free code records are a variant of ASCII records. A Format Control of ASCII ESCAPE (33 octal) is employed to differentiate the two. In Free code, screen formatting is done by the user by including the proper control sequences (e.g. CR, LF, Cancel). This allows for entirely "transparent" output to a terminal (graphics, binary paper tape, binary cassette tape).

5.4 Input code selection

All three record types for Input are available. The selection is done by the preceding Output record, and the user must take care not to "type ahead" in case of mode changes.
While a Display code Output record is followed by Display code Input, for ASCII records the FTB (bits 0 and 1) signal the Input code

<table>
<thead>
<tr>
<th>FTB code (bits 0 and 1)</th>
<th>Input code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Display</td>
</tr>
<tr>
<td>1</td>
<td>ASCII</td>
</tr>
<tr>
<td>2</td>
<td>Free</td>
</tr>
<tr>
<td>3</td>
<td>Retain existing</td>
</tr>
</tbody>
</table>

5.5 Console parameter table setting

User programs can change the console parameters of the attached terminal by sending a special ASCII record with DLE (20 octal) as the Format Control character to the terminal. The SUPERMIX concentrator intercepts this record and changes its parameter table accordingly. The format of the message follows:

```
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 0 | 0 | 0 | 0 | 0 | 0
|   | V2|   | I3|   |   |
|   | V3|   |   | I4|   |
|   |   |   |   |   | V4|
|   |   |   |   |   |   |
```

I = parameter index
V = parameter value

The parameter indices and the corresponding parameters are as follows. (Each parameter index is a single ASCII alphabetic character.)
<table>
<thead>
<tr>
<th>Index character</th>
<th>Parameter</th>
<th>Action/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Console type -- single numeric character</td>
<td>Change the type index in the CPT -- copy to CPT standard settings for this type of console for</td>
</tr>
<tr>
<td></td>
<td>0 = TTY</td>
<td>a) No. of chars in line</td>
</tr>
<tr>
<td></td>
<td>1 = T4002</td>
<td>b) No. of lines in page</td>
</tr>
<tr>
<td></td>
<td>2 = T4010</td>
<td>c) Erase procedure index</td>
</tr>
<tr>
<td></td>
<td>3 = T4012</td>
<td>d) New Page procedure index</td>
</tr>
<tr>
<td></td>
<td>4 = VISTAR</td>
<td>e) New Line procedure</td>
</tr>
<tr>
<td></td>
<td>5 = GTE, 300 bps</td>
<td>f) Return procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>g) Line Advance procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>h) End of Page procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i) Console input speed select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>j) Console output speed select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Select new I/O speeds on the console</td>
</tr>
<tr>
<td>B</td>
<td>Echo enabled -- single numeric character</td>
<td>Copy character to CPT</td>
</tr>
<tr>
<td></td>
<td>0 = enable echo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = disable echo</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>No. of characters in a line -- single numeric character</td>
<td>Copy character to CPT</td>
</tr>
<tr>
<td>D</td>
<td>No. of lines in a page -- single numeric character</td>
<td>Copy character to CPT</td>
</tr>
<tr>
<td>E</td>
<td>Erase procedure index</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>New Page procedure index</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>New Line procedure index</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Return procedure index</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Line Advance procedure index</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>End of Page procedure index</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>Echo char. for Delete Char.</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Echo char. for Delete Line</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Enable check for EOR code. When in Free code</td>
<td>Copy to CPT</td>
</tr>
<tr>
<td></td>
<td>Input mode -- single numeric character</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Disable check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Enable check</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>EOR character for Free code Input -- single</td>
<td>Copy to CPT</td>
</tr>
<tr>
<td></td>
<td>numeric char. 0-255</td>
<td></td>
</tr>
<tr>
<td>Index character</td>
<td>Parameter</td>
<td>Action/Notes</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>O</td>
<td>Input End of Record char.</td>
<td>all single ASCII characters</td>
</tr>
<tr>
<td>P</td>
<td>Input End of Line char.</td>
<td>Copy to CPT</td>
</tr>
<tr>
<td>Q</td>
<td>Delete Input Character char.</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Delete Input Record char.</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Interrupt char.</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Tab. char.</td>
<td></td>
</tr>
</tbody>
</table>

| V               | Tab. setting string |
|                 | - multiple numeric char. |
|                 | - first char. after V gives character count |
|                 | n, subsequent n single numeric chars give |
|                 | the new tab. settings. |

| W               | Set new Input speed. One-word parameter -- |
|                 | the two chars following W are combined into |
|                 | one 16-bit word. Speed is specified in |
|                 | bits per second; 110, 150, 300, 600, 1200, |
|                 | 1800, 2400, 4800, 9600 are permitted. |

| X               | Set new Output speed specified as for |
|                 | Input speed. |

| Y               | Select maximum Free code record length. |
|                 | One-word parameter as for W and X. |

A special interactive program (called USER) was developed to alter the console parameter setting; it applies the above-mentioned technique of outputting special ASCII records.

5.6 Assistance facility

A first-level user assistance facility, containing basic information on SUPERMUX and INTERCOM, operates as long as the HP 2100 is up, regardless of the state of INTERCOM. About 1K of core is budgeted for assistance information; the contents of the assistance facility can be modified at SUPERMUX assembly time by replacement of the macro call(s) generating the message text(s) affected. By pressing CTRL-A (except in Free code Input mode), the user enters (or leaves) Assistance mode. Once in this mode, single keys pressed at the terminal produce corresponding assistance messages, e.g. the key 'Y' produces the Directory of Contents shown below:
5.7 Screen Full and Type Ahead

When the screen of a SUPERMUX display terminal is full, output from INTERCOM is suspended by SUPERMUX and the message 'PTO' (Please Turn Over) appears at the screen lower right. To clear the screen and resume output, the user must type a single Carriage Return. Any other input is treated as an INTERCOM command "Type Ahead" -- it is echoed on the screen, producing roll-up (e.g. on a Tektronix 4023) or overprinting in the second screen zone (e.g. on a Tektronix 4012) depending on the particular display hardware. Type Ahead commands are sent to INTERCOM and executed; their number is not limited by any SUPERMUX parameter. When a single Carriage Return is eventually entered, output resumes and the output from the Type Ahead commands then follows in sequence.
CHAPTER III

TEST OVERVIEW

A set of diagnostics for the CDC 3000 channel Hewlett-Packard coupler has been developed, called DIAGNOS. It consists of a CM program and a PPU package running on a CDC 6000 series computer, and a package running in the HP 2100 or HP 2116. Furthermore, a diagnostic program has been written to test HPATIC line controllers. It runs in a HP 2100 and is accessible through any console on the system. Hewlett-Packard based channel simulation has reduced the on-line tests to a minimum, and provides the maintenance service with a valuable tool for off-line error tracing. Finally, before putting the system into production, an external multiple-user simulator has been connected to the system. Each of these subjects will be discussed in detail in the following sections, which deal with:

1. Coupler diagnostics (DIAGNOS)
2. The Test program for the HPATIC line interface
3. Host-concentrator simulator
4. A multiple-user simulator.

1. COUPLER DIAGNOSTICS (DIAGNOS)

1.1 CM resident DIAGNOS

This program runs at a normal control point. It consists of a display communication interface, a command processor, and the interface to the PP resident part of DIAGNOS.

A data card tells DIAGNOS to accept commands from the CDC 6000 series operator display or a special-purpose display (TAPRAK console). Another command input medium can be a prestored command session on a permanent file.

The following tests are implemented:

TEST

0  Give command directory
1  Set block length, block pattern and EST mnemonic
2  CDC 3000 channel status set and read
3  CDC 3000 channel status and mask register
4  CDC 3000 channel-HP status communication
5  CDC 3000 channel interrupt on Read/Write + Status Clear
6  CDC 3000 channel data test (Output)
7  CDC 3000 channel data test (Input)
8  CDC 3000 channel data test (Input and Output)
Furthermore, the following options can be selected:

SJ1  Continue running after Error, accumulate summary
SJ2  Continue running after Error, no summary
SJ3  Stop immediately after Error and give diagnostic
LON  Keep log to be printed at end of session
LOF  No log kept
END  End of session

Each test is run for a major cycle (100 times) if no error occurs before it accepts
the next command. For tests 6 through 8 additional questions about the hardware transmis-
sion modes (1 or 2 or mixed) are asked.

1.2 PP resident DIAGNOS

The PP package is called by CM DIAGNOS to perform the actual communication with the
coupler via a CDC 6681 channel converter. It performs the transmission of functions and
data, and verifies the results. In case of errors an error number is returned to CM resident
DIAGNOS.

1.3 HP resident DIAGNOS

The package is loaded separately into the Hewlett-Packard from cards or paper tape.
It acts as a store to the PP resident DIAGNOS by getting the test number communicated via
the status register.

2. THE TEST PROGRAM FOR THE HPATIC LINE INTERFACE

Because the asynchronous line controller is a highly programmable one, a flexible
over-all test of the various functions is required.

The program performs data and function tests in a dynamic manner. It does not require
the card to be tested in any specific slot position and permits any console connected to
the system to take command.

2.1 The control console

The test program loaded into an HP 2100 computer starts at address 100 B and requires
a console to control the various test facilities. This console must be switched to full
duplex, 9600 bits/sec screen speed and 110 bits/sec keyboard speed. In case these bit
speeds are not available, the control words could be altered in addresses 101 B and 102 B,
in accordance with HPATIC specifications (bit speeds in the least significant four bits!).
The Screen Erase characters ESC and Control L are appropriate to the Tektronix T4012 ter-

inal. They could, however, be removed or altered by changing position 104 B in core.

2.2 The program Search

Initially, the program is in Search mode. This permits every console on the system to
send a character "W" and receive a full listing of all the test facilities. This leaves
the program in Search mode and allows other consoles to take over. Only after typing a "Z"
is the search abandoned, and the console where the last "W" has been typed will be defined
as the "control console". Return to Search will be accomplished by typing a "V" and rede-
finition of the control console is possible as before.
2.3 The program Test

The program to test the HPATIC is called with a "2" from Search. Once in Test an "X" gives the full facility listing. As soon as the words "TEST PARAMETERS", appear on the screen, the various tests can be selected.

The full facility list is as follows.

HPATIC TEST
CONTROL CONSOLE (9600 B SCREEN AND 110 B KEYB.)
MAY BE CONNECTED TO ANY SLOT NUMBER
CONTR. WORDS FOR CONTROL CONSOLE IN MEM. LOC 101 B, 102 B
A TEST BREAK FACILITY
B TEST CHAR. LOST FACILITY
0110, 0300, 0600, 1200, 1800, 2400, 4800, 9600 (FOUR DIGITS)
D + BIT SPEED PARITY TEST (4, 5)
E + BIT SPEED PARITY TEST (7)
F 16 BIT MODE
G FLAG TEST
H RANDOM DATA
I TEST PERIOD DELAYED, DELAY TIME IN S REG.
K DISPLAY SUPPRESSED
L TOTAL TEST (WITH OR WITHOUT H)
N + BIT SPEED OUTPUT ONLY
P + OCTAL SLOT NUMBER DEFINE TEST POSITION
V GO TO REDEFINITION OF CONTROL SLOT
W REDEFINE SLOT POSITION
X EXIT AND RESTART
Z GO TO TEST PROGRAM

I.E. TEST PARAMETERS, FH 9600 CR OR L CR
TYPE X FOR FULL FACILITY LIST
TEST PARAMETERS,

2.4 Definition of the Test slot

As soon as the control console has been defined, the position of the card to be diagnosed has to be defined. This happens by typing a "P", the slot number, and a Carriage Return. A successful definition is replied to by: YOU HAVE NOW DEFINED THE POSITION OF THE CARD TO BE TESTED. Every message is followed by "TEST PARAMETERS", hence a new test can be selected at once.

2.5 The different tests

2.5.1 "Break" test

Upon typing an "A", a Break test is performed. It makes the program set the input speed of the card to 9600 bps and the output speed to 110 bps. The first opening bit received behaves as a Break signal for the high bit-rate input logic. If a Break is successfully detected, the resulting status bits 11 and 9 will be displayed or printed on the control terminal.

2.5.2 "Character Lost" test

This test is selected by typing a "B". It brings the card into 16-bit mode but delays acceptance of the first input character, until the second part of the 16-bit word arrives. The message terminal will print the appropriate status bit 8 on completion. One notices the presence of both the input and output status bits 10 and 11 in this test.
2.6.3 "Flag" test

Apart from the ordinary flag generation in data Output and Input tests, flags are selectable separately. Keyboard character "G" activates the Flag test by generating the functions 9, 10, and 11. Function 9 is only to be tested with an oscilloscope as it causes a 700 nsec pulse to be seen at the 48-pin connector (pin "T" or "U")

2.6.4 "Data" tests

Data tests are selected by typing in the required bit rate in the following way: "0110 CR" makes the card transmit a fixed U-character at 110 bps. After reception, the received and transmitted characters are compared and a message is sent to the message terminal in case of No Comparison.

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0150</td>
<td>150 bps</td>
</tr>
<tr>
<td>0300</td>
<td>300 bps</td>
</tr>
<tr>
<td>0600</td>
<td>600 bps</td>
</tr>
<tr>
<td>1200</td>
<td>1200 bps</td>
</tr>
</tbody>
</table>

The 16-bit format is tested by selecting the appropriate functions, using the keyboard character "P". The transmitted 16-bit word corresponds to the two ASCII characters "U" and "Space", both without parity.

2.6.5 "Random" data

If one types an "H" as well, pseudo-random data are transmitted. No Comparison messages with the specific bit values are printed in case of erroneous transfers.

2.6.6 "Delayed" operation

By typing an "I" one creates gaps between successive outputs. The length of the gap is defined by the contents of the switch register and is zero when all switches are set to one.

2.6.7 "No Print" facility

Particularly in cases of failure it is useful to provide a continuous loop in spite of Data Comparison errors. Error checking is disabled if a "K" is typed in addition to the four digits defining the bit rate.

2.6.8 "Output" only

With the keyboard "N", Input will be disabled. This permits data tests to be done in case Input is not functioning properly. It requires a definition of bit speed and mode, as for all other data tests.

2.6.9 "Total" test

A general Input/Output test is selected typing "L" followed by a CR. It makes the program update the bit rate of the card after every successful data transfer. In this way it selects successively the nine possible bit speeds. The "L" test is done in 16-bit mode and may be combined with "H" for random data.
2.6.10 "Parity" test

Two Parity tests are selectable. By typing a "D" and a four-digit bit speed, the combination of functions 5 and 6 will be tested. One is reminded that:

Bit 6 - selects Input parity check and
Bit 5 - selects even parity in and out (otherwise odd).

The test is normally performed with the standard fixed Data word ASCII U - SPACE. This word is particularly chosen because the No Error Detected status bit will be a zero and consequently No Comparison will be the result in bit 7 of the Data word.

With an "E", function 7 is activated.

One is reminded that bit 7 selects Output parity generation. Here the appropriate odd parity bit is generated, and the resulting data word will not be modified with the parity check result as the input check has not been enabled. The result is again an Error message due to the appropriate choice of the fixed Data word. This time the error is in bit 15.

2.6 Wrong parameter settings

In case some parameters are wrongly combined or in case a card under test is not equipped with the special Input/Output strap, an Error message will appear on the screen of the control console. This message will also appear if one tries to define a card to be diagnosed having the slot position of the control console itself.

2.7 Examples

Type "W" on any standard terminal.

Result: Complete listing of all the facilities. The terminal where the last 'W' has been typed will be the control console.

Type "Z" on any standard terminal.

Result: Complete listing and entrance into the test program.

Type "P16" followed by a Carriage Return (CR) on the control terminal.

Result: Channel 16 has been defined as the channel to be tested. An acknowledging message appears on the screen.

Type "A" on the control terminal.

Result: CHARACTER LOST OR BREAK DETECTED

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>09</th>
<th>08</th>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Type "DF 4800" followed by a Carriage Return.

Result: A mismatching in the parity bit (07) and the result is a No Comparison message.

NO COMPARISON IN 16 BIT MODE

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>09</th>
<th>08</th>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>REC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SEND</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Type "NHF 9600" followed by a Carriage Return.

Result: Every 1000 successful transfers, a message appears. This test does not require any test strap and is consequently to be done on all interfaces. In case terminals are connected instead, characters will be displayed continuously.

3. **HOST-CONCENTRATOR SIMULATION**

On-line hardware testing on computers which are used 24 hours a day, seven days a week, has proved to be a difficult problem. Therefore it was decided to develop a Host-concentrator simulator to enable off-line coupler testing and reduce the on-line test time to a minimum.

The heart of the simulator is an HP 2100 series minicomputer with 4K of memory. The minicomputer acts both as Host and concentrator during off-line coupler testing (see Fig. III.1). The simulator can be divided into three parts:

a) a Host interface simulator unit which contains the CDC 3000 type drivers, receivers, and connector, and associated logic;

b) controlware in the HP consisting of three I/O interfaces, and software to generate and check all the CDC coupler functions;

c) controlware in the HP consisting of two I/O interfaces, and software to generate and check all the concentrator-coupler functions.

The hardware of the Host interface is entirely controlled by software in the minicomputer. All the channel control, status, and Data signals are represented by bits on different I/O channels. Parity checking and generation is done by hardware.

The concentrator connections are compatible with the simulator. All the software is developed in assembly language. Drivers exist for all the channel functions. A diagnostic package is available for total off-line testing.

![Diagram of Host-concentrator simulator configuration](image)

*Fig. III.1 Host-concentrator simulator configuration*
4. A MULTIPLE-USER SIMULATOR

In order to test the system before it was put into production, a multiple-user simulator was developed. It consists of an HP 2100 computer equipped with line controllers similar to those used in SUPERMUX itself. It is connected to the system in such a manner that every line interface in the simulator acts as a user, hence it is programmed typically at 9600 bps on input and 110 bps on output. In case the 9600 bps input feature on the system has to be tested, a mere change in the simulator bit-speed table settings will be sufficient. A special control console in the simulator can take command over any of the simulator ports and is to be used for monitoring or for special commands (i.e. LOG OUT). After an initial automatic LOG IN procedure of the various simulator ports (up to 36), the system will be brought into Edit mode. A special command on the control console will activate the simulator sessions. Messages will be generated in the simulator and sent across the various ports to the corresponding input ports of SUPERMUX.

Both the length of the messages as well as the typing interval are defined by random numbers generated by the power method. The average typing speed is variable and selectable at the control console. In the ultimate case, intervals between characters will go to zero. In this way the resulting continuous data streams simulate buffered terminals instead of non-buffered ones. Selective monitoring of messages returned by SUPERMUX is possible by means of the control console. Incoming data are passed through a circular buffer and sent to the console screen.

So far this system has been used with 18 simulator ports on all speeds possible without a significant degradation of the system.

Acknowledgements

Of the many people who contributed to the realization of this project, we want to mention in particular:

Colin D. Gilbert, who was involved with the initial specifications of the project and Peter van de Kerk who developed the IC transmitters and receivers on the CDC channel. The staff of the DD Electronics Workshop constructed the various pieces of hardware (B. Maulini excelled in making the HPATIC layout faultlessly). The CE maintenance engineers did most of the hardware testing.

The detailed logic diagrams have been drawn by Mrs. O. Marais, and Miss Ingrid Kurth made the flow charts and block diagrams.

We would also like to thank the CERN Scientific Reports Typing Service and the Documents Reproduction Section for their contribution to the printing of SUPERMUX.

Finally we would like to acknowledge J.B. Sharp for his encouragement to publish this paper.
REFERENCES


2) Control Data Corporation, Control Data 3000 series computer systems input/output specifications, Publication No. 600488800; and CDC Reference manual, Publication No. 60108400.

3) Hewlett-Packard, A pocket guide to the 2100 computer (Hewlett-Packard Company, Cupertino, California, USA) (5951-4423).

4) Hewlett-Packard, Operating and service manual, 12566B, Microcircuit Interface Kits (Hewlett-Packard Company, Cupertino, California, USA).

5) E.M. Palandri, The HP 2115B/HP 2116B - CDC 3000 Interface, FOCUS Note FN-54, DD Division, CERN.

6) T. Bruins and P. Van de Kerk, Driving the CDC 3000 channel with integrated circuits, Data Communication Note 25, DD Division, CERN.


8) C.D. Gilbert, Hewlett-Packard 2100 asynchronous terminal interface, Data Communication Note 2, DD Division, CERN.


10) Hewlett-Packard, Microprogramming guide for the Hewlett-Packard model 2100 computer (Hewlett-Packard Company, Cupertino, California, USA).


APPENDIX I

THE LOGIC DIAGRAMS OF THE
CDC-HP COUPLER
Z61 = D30
All resistors on D30
22 or 23 C11-E1 are COUPLER SELECTED

MODIFICATIONS

CERN-DD

GENEVE

CONNECT LOGIC

6000 PART

S 077 - 22 - 2
Z43 : B17 from 6000 via Z51, Z52 (DATA)
Set inputs of FF's connected to +5V
BUFFER CONTROL 2

COUPLER CDC-HP M6000-B
APPENDIX 2

THE LOGIC DIAGRAMS OF THE
HPATIC LINE INTERFACE
NOTE: ALL RESISTORS UNLESS OTHERWISE INDICATED: ±5% 1/8W
ALL 1K RESISTORS USE INTERNAL RESISTORS IN IC PACKAGE CT3L 955
GROUND ON PINS: 1, 2, 8, 38, 47, 48, 59, 60, 69, 70.
ALL PIN REFERENCES CONCERN THE 85 PINS EDGE CONNECTOR UNLESS OTHERWISE NOTED