

---

**RCSL No:** 991 10630

**Edition:** September 1986

**Author:** Flemming Biggas  
Ole Thomasen

---

**Title:**

RC8000/ADP3270 Input/Output Process  
Reference Manual

---

---

**Keywords:**

RC8000, ADP3270, Local Area Network

---

**Abstract:**

Reference Manual for RC8000/ADP3270 Input/Output Process. The manual describes the user interface and the associate functions. It covers the specific aspects of using the RC8000/ADP3270 Input/Output Process for Local Area Network communication, and is meant to be used by RC8000 users.

---

(40 printed pages)

---

| <b>TABLE OF CONTENTS</b>                           | <b>PAGE</b> |
|----------------------------------------------------|-------------|
| 1. INTRODUCTION.....                               | 1           |
| 2. DEVICE HANDLING.....                            | 2           |
| 2.1 Device Addressing.....                         | 2           |
| 2.2 Input/Output.....                              | 2           |
| 2.2.1 I/O processes.....                           | 2           |
| 2.2.2 Extended use.....                            | 3           |
| 2.3 Application Select Menu.....                   | 3           |
| 2.4 Printers.....                                  | 4           |
| 2.4.1 Print formats.....                           | 4           |
| 2.4.2 Printer control characters.....              | 4           |
| 2.4.2.1 Transparent print format.....              | 5           |
| 2.4.2.2 Fixed print format.....                    | 5           |
| 2.4.3 Printing operations.....                     | 6           |
| 2.5 Compatibility.....                             | 6           |
| 2.5.1 Device addressing.....                       | 6           |
| 2.5.2 Connect.....                                 | 6           |
| 2.5.3 Printer handling in the ADP3270 process..... | 6           |
| 3. MESSAGE FORMAT.....                             | 8           |
| 3.1 First and Last Storage Address.....            | 8           |
| 4. ANSWER FORMAT.....                              | 9           |
| 4.1 Result.....                                    | 9           |
| 4.2 Answer Buffer.....                             | 9           |
| 4.2.1 Status word.....                             | 9           |
| 5. ADP3270 PROCESS.....                            | 10          |
| 5.1 Control Messages.....                          | 10          |
| 5.1.1 Sense.....                                   | 10          |

**TABLE OF CONTENTS****PAGE**

|          |                                       |    |
|----------|---------------------------------------|----|
| 5.1.2    | Sense ready.....                      | 11 |
| 5.1.3    | Connect.....                          | 11 |
| 5.1.4    | Disconnect.....                       | 11 |
| 5.1.5    | Lookup device.....                    | 12 |
| 5.1.6    | Reserve device.....                   | 12 |
| 5.1.7    | Release device.....                   | 13 |
| 5.1.8    | Wait ready.....                       | 13 |
| 5.1.9    | Lookup address.....                   | 14 |
| 5.1.9.1  | Lookup physical.....                  | 14 |
| 5.1.9.2  | Lookup logical.....                   | 15 |
| 5.2      | Output Message.....                   | 16 |
| 5.3      | Output Format.....                    | 16 |
| 5.3.1    | Write command.....                    | 17 |
| 5.3.2    | Erase write command.....              | 17 |
| 5.3.3    | Copy command.....                     | 17 |
| 5.3.4    | Erase all unprotected command.....    | 18 |
| 5.3.5    | Read buffer command.....              | 18 |
| 5.3.6    | Read modified command.....            | 18 |
| 5.3.7    | Read modified all command.....        | 19 |
| 5.3.8    | Write control character.....          | 19 |
| 5.3.9    | Copy control character.....           | 20 |
| 5.3.10   | Device buffer orders.....             | 20 |
| 5.3.10.1 | Start field.....                      | 20 |
| 5.3.10.2 | Set buffer address.....               | 21 |
| 5.3.10.3 | Insert cursor.....                    | 21 |
| 5.3.10.4 | Repeat to address.....                | 21 |
| 5.3.10.5 | Program tabulation.....               | 21 |
| 5.3.10.6 | Erase all unprotected to address..... | 22 |
| 5.3.10.7 | Unsolicited message.....              | 22 |
| 5.3.11   | Output elements.....                  | 22 |

TABLE OF CONTENTSPAGE

|        |                                                 |    |
|--------|-------------------------------------------------|----|
| 5.3.12 | Transfer display device.....                    | 23 |
| 5.4    | Input Message.....                              | 24 |
| 5.4.1  | Status.....                                     | 24 |
| 5.5    | Input Format.....                               | 24 |
| 5.5.1  | AID codes, read commands and input formats..... | 26 |
| 5.5.2  | Status byte.....                                | 27 |
| 5.5.3  | Input elements.....                             | 27 |

APPENDIX:

|    |                                   |    |
|----|-----------------------------------|----|
| A. | AID CODES.....                    | 29 |
| B. | MESSAGE/ANSWER FORMAT SURVEY..... | 31 |



## 1. INTRODUCTION

This manual describes the external process interface to IBM 3270 compatible display and printer devices within the RC local area network, through the RC8000/ADP controller. Both display and printer devices are supported by the ADP3270 process.

## 2. DEVICE HANDLING

The external process ADP3270 implements the support of IBM 3270 type devices on RC8000, in a local area network environment. The devices are RC855 display stations and attached printers, connected to the local area network through a RC890 control unit.

### 2.1 Device Addressing

Within the RC local area network a device is unambiguously identified by the control unit index 'cu' and an associated device index 'dev'. The control unit index 'cu' is in the range from 0 to 127, and the device index 'dev' is in the range from 0 to 127.

### 2.2 Input/Output

Input (answers to input messages) is delivered as a response to an operator attention key (SEND, PF, PA etc.) activation, and contains the 'cu,dev' of the sending device (device address), the key which was activated (AID code) and data (if any). The 'cu,dev' address delivered are or-ed with 128.

Output must contain the device address, 'cu,dev', ESC and a 'write' operation including data to be entered in the device buffer of the device pointed out by the device address. The 'cu,dev' address delivered are and-ed with 127 before use.

The above indicates the normal use of I/O in connection with IBM 3270 type devices in a RC8000 environment.

#### 2.2.1 I/O processes

The ADP3270 process is represented by respective an input and an output process, and in order to communicate with these two processes, a RC8000 link must be made to both processes. General all output messages must be

transferred to the ADP3270 process via the output link, and all input messages to the ADP3270 process via the input link.

Control messages must be transferred to the ADP3270 process via the link, the messages concern.

### 2.2.2 Extended use

ADP3270 removes some previous limitations, concerning the handling of IBM 3270 type devices, in RC8000 context, and offers the following 'extensions' to the command set which may be specified as operations in output messages:

Device specific read operations, which are used to initiate read operations at devices on the request of the application (see 5.3 Output Format). The response (input) will arrive as an answer to an input message, as well as operator initiated input (see 5.5 Input Format).

Control operations, which may be used for erase and copy purposes between devices residing at the same control unit (see 5.3 Output Format).

### 2.3 Application Select Menu

The control units maintain an 'menu' which is displayed in order to enable terminal users to select application . In order to be 'selected' by a terminal, the application must send a 'connect' operation (see 5.1.3 Connect) to ADP3270 specifying the application select identification which is to be entered in the 'menu' of the control units. The 'connect' operation establish a session between the application and the available control units. The application need only send one 'connect', as ADP3270 automatically 'distributes' it to all available control units.

If a session already exist at the time of 'connect', it is disconnected since only one session is handled per ADP3270 process at a time.

Additionally an application may 'remove' itself from the application select menu by sending a 'disconnect' operation (see 5.1.4 Disconnect).

When the application has been 'selected' an ATTENTION CONNECT (see 5.5 Input Format) is delivered to the application. No indication is given, when a terminal user disconnects himself and selects another application.

## **2.4 Printers**

Generally printers should be handled through the dedicated printer process, but it is possibly to use the ADP3270 process, which is tailored also to handle IBM 3270 type printers from a RC8000 application program.

### **2.4.1 Print formats**

Two types of print formats can be used:

Transparent format with an instalation dependant maximum line length. New line and other printer control characters can be inserted in the printer buffer to control formatting.

Fixed format with 40, 64 or 80 characters per line. Control characters to cause the printer to move to a new line are generated automaticly. A limited support is provided for printer control characters.

Regardless of the print format, attribute characters and all characters, including printer control characters, which occur in fields with the non displayable attribute, are treated as nulls during a print operation, i.e. they are printed as blank positions or completely supressed.

### **2.4.2 Printer control characters**

In conjunction with a write, erase-write, or copy command (see 5.3 Output Format), printer control characters can be written as data characters in the printer buffer.

#### 2.4.2.1 Transparent print format

NL (value=10): Causes the the carriage return and line feed charcters to be generated. If NL (or CR) is missing, carriage return and line feed will be generated automaticly when the maximum line print length is obtained.

CR (value=13): Pass through.

VT (value=11): Pass through.

FF (value=12): Pass through.

EM (value=25): Causes the carriage return and line feed characters to be generated, unless the EM character immediately follows an NL character. Subsequently terminates the printing operation. If no EM character is present, the printing operation terminates when the end of buffer is reached. At this time carriage return and line feed characters are generated as if an EM character had been encountered.

#### 2.4.2.2 Fixed print format

NL (value=10): Treated as null.

CR (value=13): Treated as null.

VT (value=11): Treated as null unless the VT character is the first in a line.

FF (value=12): Treated as null unless the FF character is the first in a line.

EM (value=25): Treated as null.

### 2.4.3 Printing operations

Printing operations can be initiated by an application, when the printer bit in the Write/Copy Control Character in a write or copy-type command addressed to a printer is set (see 5.3.8 Write Control Character and 5.3.9 Copy Control Character). A transparent or fixed print format is used, as specified in the print format bits of the write or copy control character.

## 2.5 Compatibility

Relevant differences compared to former implementations of IBM 3270 device support in RC8000 are described in the following sections.

### 2.5.1 Device addressing

In former implementations the device address 'cu,dev' was merely an index, indicating the sequence in which devices had logged on (select) to the application. In ADP3270 the device address 'cu,dev', unambiguously identifies the location of the device, within the local area network to which it belongs.

### 2.5.2 Connect

In the ADP3270 implementation of IBM 3270 device support, the application must send a connect operation, in order to access the available devices.

### 2.5.3 Printer handling in the ADP3270 process

The printer status block, which in former implementations had the format:

---

! cu ! dev ! s0 ! s1 ! ETX !

Where s0 and s1 are status bytes, now has the format:

! cu ! dev ! AID(=156) ! SB ! ETX !

SB Status Byte now contains the status information (see 5.5.2 Status byte).

### 3. MESSAGE FORMAT

The general format of message to ADP3270 is:

message:

- +0: operation < 12 + mode
- +2: first RC8000 storage address
- +4: last RC8000 storage address

#### 3.1 First and Last Storage Address

If first storage address is odd it is decreased by one. First and last storage address must point inside the sending process.

#### 4. ANSWER FORMAT

##### 4.1 Result

The following results may be delivered by the wait answer monitor call:

result 1: message accepted, the answer buffer may contain further information.

result 2: message rejected because of violation user/reserver requirements.

result 3: message unintelligible; illegal operation or mode.

result 4: malfunction, ADP controller troubles.

result 5: receiver unknown, the specified ADP3270 process does not exist.

The answer contains the logical RC8000 status, which may be supplemented by a 'result byte'.

##### 4.2 Answer Buffer

The contents of the answer buffer is only relevant if result is 1.

+0: status word (or zero)

+2: operation dependant

+4: operation dependant

+6: operation dependant

+8: operation dependant

##### 4.2.1 Status word

The status word is bit encoded:

timer (bit 2):

This bit is set if the status byte indicates: not ready, timeout or unavailable.

## 5. ADP3270 PROCESS

Operations can be initiated by an internal process which has initialized or reserved the process, i.e. respective the input and output process of the ADP3270 process.

The ADP3270 output process accepts output and control type messages, and the ADP3270 input process accepts input messages, according to the specifications listed below.

In order to gain access to the available control units of the system, the application must send a connect operation (see 5.1.3 Connect) to the process, thereby entering its identification name in the 'application select menu' of the control units.

### 5.1 Control Messages

Generally all control messages, except sense ready, must be send to the ADP3270 output process. The control message sense ready must be send to the ADP3270 input process.

#### 5.1.1 Sense

message:

answer:

+0: 0 < 12 + 0

+0: 0 (status)

+2: irrelevant

+2: 0

+4: irrelevant

+4-10: <:<appl. select name>:>

This message is implemented to sense the application select name. The message is answered immediately with status equal to zero, and the application select name defined in a previous connect message. If no connect message has been sent yet, the first three characters of the name will be binary zeroes.

5.1.2 Sense ready

message:

answer:

+0: 0 &lt; 12 + 2

+0: status

+2: irrelevant

+2: 0

+4: irrelevant

+4: 0

This message must be send to the ADP3270 input process.

status: 0; ok

bit2; timeout

5.1.3 Connect

message:

answer:

+0: 4 &lt; 12 + 4

+0: 0 (status)

+2: irrelevant

+2: irrelevant

4-10: &lt;:application select name&gt;: 4-10: unchanged

A session is assigned to the application and the 'application select name' is entered at all cu's accessible through this process. No devices are accessible before the CONNECT has ben called; successible call's of CONNECT will disconnect any previous connected devices.

5.1.4 Disconnect

message:

answer:

+0: 4 &lt; 12 + 8

+0: 0 (status)

+2: irrelevant

+2: irrelevant

An existing session (if any) is disconnected, and the application identification is withdrawn from the select menu of the control units.

### 5.1.5 Lookup device

message:

+0:  $4 < 12 + 12$   
 +2: irrelevant  
 +4:  $0 < 16 + cu < 8 + dev$

answer:

+0: 0 (status)  
 +2: result  
 +4: unchanged  
 +6: AID byte  
 +8: status byte

This message enquires whether the device is available and whether it is a printer or a display device. The AID and the status byte are only relevant if result equals zero, (see 5.5.1 AID codes, read commands and input formats, and 5.5.2 Status byte).

result: 0; ok

6; unavailable  
 7; device number out of range

### 5.1.6 Reserve device

message:

+0:  $4 < 12 + 16$   
 +2: irrelevant  
 +4:  $0 < 16 + cu < 8 + dev$

answer:

+0: 0 (status)  
 +2: result  
 +4: unchanged

This request tries to reserve the printer pointed out by 'cu,dev'.

result: 0; ok

1; not processed, regretted  
 3; no resources  
 6; unavailable  
 7; device number out of range  
 8; device not a printer  
 10; printer reserved

12; printer busy, hardcopy

#### 5.1.7 Release device

message:

answer:

+0: 4 < 12 + 20

+0: 0 (status)

+2: irrelevant

+2: result

+4: 0 < 16 + cu < 8 + dev

+4: unchanged

This request cancels a previous (printer) reservation.

result: 0; ok

7; device number out of range

8; device not a printer

11; device not reserved

#### 5.1.8 Wait ready

message:

answer:

+0: 4 < 12 + 32

+0: 0 (status)

+2: irrelevant

+2: result

+4: 0 < 16 + cu < 8 + dev

+4: unchanged

This request may be used when a reserve printer request has been answered with: busy, reserved or unavailable. When the request is answered, positively, the printer is reserved by the application.

result: 0; ok

1; not processed

3; no resources

7; device number out of range

8; device not printer

### 5.1.9 Lookup device address

The lookup operations have been included in the ADP3270 process purely for compatibility reasons, as earlier systems exploit a dynamic assignment of device addresses ('link,cu,dev'), which had no affinity whatsoever with the physical locations of devices. In ADP3270 the addresses known to the application, unambiguously identifies the location.

The message format of the LOOKUP OPERATION contain the following fields:

line, cu, dev (physical device address):

line ; always zero in ADP3270 context (byte).

cu ; cu index associated with the control unit (byte).

dev ; device number within the control unit (byte).

link, cu, dev (logical device address):

link ; always zero in ADP3270 context (byte).

cu ; cu index associated with the control unit (byte).

dev ; device number within the control unit (byte).

The answer results are:

result (lookup result) (byte):

result = 0 ; ok. )\*

result = 1 ; limits exceeded, may be caused by line <> 0 or link <> 0 or  
dev > 127.

result = 2 ; cu does not exist (i.e. is not connected).

)\* ok result means that the cu is accessible and connected to ADP3270, but gives no indication to the type of device (dev) or whether it is connected to ADP3270.

#### 5.1.9.1 Lookup physical

message:

answer:

|                   |                   |
|-------------------|-------------------|
| +0: 2 < 12 + 14   | +0: 0 (status)    |
| +2: not used      | +2: result        |
| +4: line, cu, dev | +4: line, cu, dev |
| +6: irrelevant    | +6: line, cu, dev |

Returns the physical address of a device specified by its logical address. In ADP3270 this operation is rudimentary, as logical and physical addresses are identical; however a simple address check is performed.

#### 5.1.9.2 Lookup logical

message:

answer:

|                   |                   |
|-------------------|-------------------|
| +0: 2 < 12 + 12   | +0: 0 (status)    |
| +2: not used      | +2: result        |
| +4: irrelevant    | +4: line, cu, dev |
| +6: link, cu, dev | +6: line, cu, dev |

Returns the logical address of a device specified by its physical address. In ADP3270 this operation is rudimentary, as logical and physical addresses are identical; however a simple address check is performed.

## 5.2 Output Message

|                   |                           |
|-------------------|---------------------------|
| message:          | answer:                   |
| +0: 5 < 12 + 0    | +0: status                |
| +2: first address | +2: halfwords transferred |
| +4: last address  | +4: bytes transferred     |
| +6: irrelevant    |                           |

Output is used to send operations from the application to ADP3270. The operations may contain write, copy and transparent operations. Some operations may cause a response to be delivered as input. The actual data format is specified in 5.3.

Write operations may be sent in more block's (messages).

## 5.3 Output format

All operations sent by ADP3270 are headed by a device address field, 'cu,dev', and terminated by 'ETX':

```

! cu ! dev ! data ! ETX !

```

cu ( cu = (0..127) ; Byte ):

Constitutes the index of the control unit (RC890) within the ADP3270 processes.

dev ( dev = (0..127) ; Byte ):

Constitutes the device index at the cu.

data ( string of Bytes ):

Text and orders, format is specified in following descriptions.

ETX ( ETX = 3 ; Byte ):

End of text, signals the end of the transaction.

It is possible to send an operation in more than one block (output-buffer), as long as a few simple rules are observed:

- a) only the first block contains the 'cu,dev' header field.
- b) only the last block contains the 'ETX' field.
- c) an operation to a device (output) must be terminated (by 'ETX') before another device is selected for output.

In the following the available commands are shown together with the required format specification.

### 5.3.1 Write command, WRT (value=49)

The write command is used to modify the existing contents of the buffer of device 'dev' at control unit 'cu'.

The command body obeys the format:

! cu ! dev ! ESC ! WRT ! WCC ! ! DEVICE BUFFER ORDER ! ! TEXT ! ! ETX !

### 5.3.2 Erase write command, EWRT (value=53)

The erase write command is used to erase the entire contents of the buffer of device 'dev' at control unit 'cu'. The cursor is moved to position 0 and the buffer address is reset to 0. If the command includes 'write parameters' ('WCC' etc.), the write operation is performed as for the write command.

The command body obeys the format:

! cu ! dev ! ESC ! EWRT ! ! WCC ! ! DEVICE BUFFER ORDER ! ! TEXT ! ! ETX !

### 5.3.3 Copy command, CPY (value=55)

The copy command is used to transfer device buffer data from device 'DEV' to device 'dev', within the same control unit 'cu'.

The command body obeys the format:

! cu ! dev ! ESC ! CPY ! CCC ! DEV ! ETX !

#### 5.3.4 Erase all unprotected command, EAU (value=63)

The erase all unprotected command performs the following actions at device 'dev' at control unit 'cu':

- a) All input fields are cleared to nulls.
- b) MDT(modified data tag) is reset in all input fields.
- c) Keyboard restore.
- d) Reset AID code.
- e) Cursor is moved to the first position in the first input field.
- f) The buffer address is set equal to the cursor address.

The command body obeys the format:

! cu ! dev ! ESC ! EAU ! ETX !

#### 5.3.5 Read buffer command, RB (value=50)

This request will initiate a read operation at control unit 'cu' at device 'dev'. Input will arrive as an answer to an input message. (see 5.5.1 AID codes, read commands and input formats).

! cu ! dev ! ESC ! RB ! ETX !

#### 5.3.6 Read modified command, RM (value=54)

This request will initiate a read operation at control unit 'cu' at device 'dev'. Input will arrive as an answer to an input message. (see 5.5.1 AID

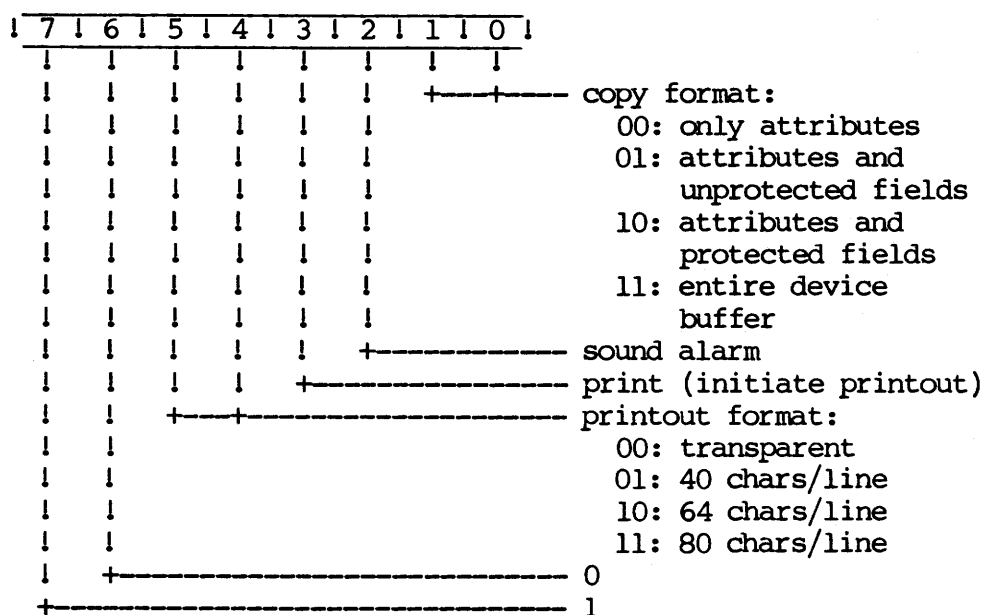
codes, read commands and input formats).

```
! cu ! dev ! ESC ! RM ! ETX !
```

### 5.3.7 Read modified all command, RMA (value=62)

### 5.3.9 Copy Control Character (WCC)

CCC byte coding:



A status block will be generated when the print bit is set and the operation is addressed to a printer device (see 5.5 Input Format).

### 5.3.10 Device buffer orders

In the body of write type command, DEVICE BUFFER ORDER's may occur as elements of the write string. The orders and their parameters are specified in the following sections.

#### 5.3.10.1 Start Field, SF (value=29)

The command body obeys the format:

! SF ! ATR !

Indicates the start of field, with attributes as indicated in ATR. The SF order may be preceded by an SBA order.

#### 5.3.10.2 Set Buffer Address, SBA (value=17)

The command body obeys the format:

! SBA ! ADDR !

The SBA order indicates the buffer address (ADDR) at which the following data should be stored. The SBA order may precede an SF, IC, PT, RA or EAU order.

#### 5.3.10.3 Insert Cursor, IC (value=19)

The command body obeys the format:

! IC !

The IC order causes the cursor to be moved to the position pointed out by the current buffer address. The IC order may be preceded by an SBA order.

#### 5.3.10.4 Repeat to Address, RA (value=60)

The command body obeys the format:

! RA ! ADDR ! CHAR !

The RA order indicates that the character CHAR is to be stored in each position starting at current buffer address and extending to but not including the position specified in ADDR, which will be the current buffer address when the operation is completed. The RA order may be preceded by an SBA order.

#### 5.3.10.5 Program Tabulation, PT (value=5)

The command body obeys the format:

! PT !

The PT order indicates that that a tabulation operation is to be performed in the buffer. The current buffer address is incremented accordingly to the address of the first position in the next input field. The PT order may be preceded by an SBA order.

#### 5.3.10.6 Erase all unprotected to address, EAU (value=18)

The command body obeys the format:

! EAU ! ADDR !

The EAU order indicates that 'nulls' are to be inserted in all input positions starting at the current buffer address and extending to but not including the position specified by ADDR, which will be the current buffer address when the operation is completed. The EAU order may be preceded by an SBA order.

#### 5.3.10.7 Unsolicited message, USM (value=31)

The command body obeys the format:

! USM !

The USM order causes the USM-lamp to be lit.

#### 5.3.11 Output elements

ADDR (field ADDRESS) ( two Bytes):

ADDR is constructed, extracting the six least significant bits from the two bytes and concatenating them to a twelve bit entity.



## 5.4 Input Message

message:

answer:

+0: 3 < 12 + 0

+0: 0; (status)

+2: first address

+2: halfwords transferred

+4: last address

+4: bytes transferred

Input message is answered (by ADP3270) either as a response to an operator request (the activation of an attention button) or as a response to an operation send by the host. The actual format is specified in 5.3 Output Format.

### 5.4.1 Status

The RC8000 status word does not reflect any status delivered by the device.

## 5.5 Input Format

All operations received by ADP3270 are headed by a device address field 'cu,dev', and terminated by 'ETX':

! cu ! dev ! data ! ETX !

cu ( cu = (0..127) ; Byte ):

Constitutes the index of the control unit (RC890) within the ADP3270 processes.

dev ( dev = (0..127) ; Byte ):

Constitutes the device index at the cu.

data ( string of Bytes ):

Text and orders, format is specified in the following descriptions.

ETX ( ETX = 3 ; Byte ):

End of text, signals the end of the transaction.

The input format depends on the attention key which is activated as well as the type of read operation which is employed to read the device data. The normal situation is that the user let ADP3270 send read operations to the devices as a response to the activation of attention keys, and only sends input messages in order to receive the generated input. When this is the case the 'read' operation will always be a 'read modified' operation and the format of the input structured accordingly.

Input may also be sent as a response to a copy or write type operation, if the 'print' bit was set in the Copy or Write Control Character (see 5.3.8 Write control character and 5.3.9 Copy control character). The user may however send specific read operations to devices, in order to obtain other responses.

In the following, the different input formats are specified as well as their relationship with attention keys (AID codes) and read operations.

format 1:     ! cu ! dev ! AID ! ETX !

format 2:     ! cu ! dev ! AID ! CUR !   ! SBA ! ADDR !   ! TEXT !   ! ETX !

format 3:     ! cu ! dev ! AID ! CUR !   ! SBA ! ADDR !   ! ETX !

format 4:     ! cu ! dev ! AID ! CUR !   ! SF ! ATR !   ! TEXT !   ! ETX !

format 5:     ! cu ! dev ! AID(=156) ! SB ! ETX !

format 6:     ! cu ! dev ! AID(=157) !   ! TEXT !   ! ETX !

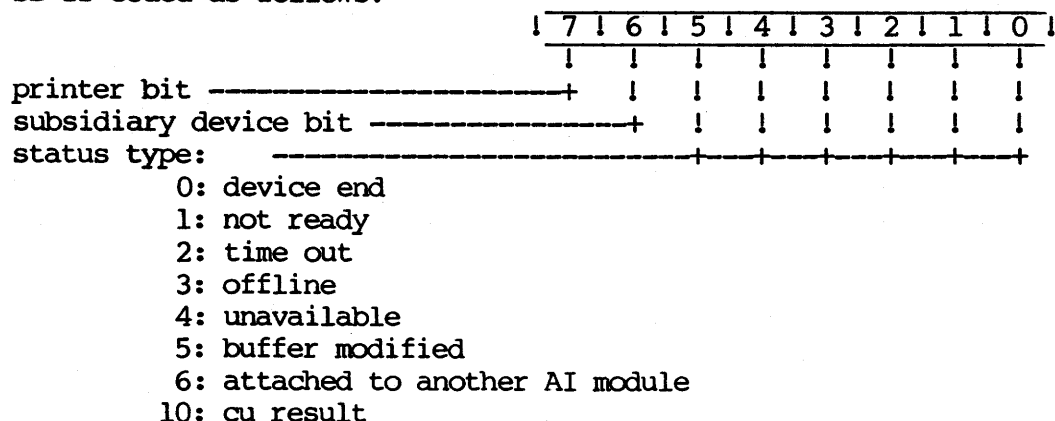
### 5.5.1 AID codes, read commands and response formats

| AID CODE (value) | ! NO COMMAND | ! Read mod. | ! Read Mod. All | ! Read Buffer |
|------------------|--------------|-------------|-----------------|---------------|
| NO AT ( 160 )    | ! none       | ! none      | ! none          | ! format 4    |
| NO AP ( 161 )    | ! none       | ! none      | ! none          | ! format 4    |
| SEND ( 129 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF1 ( 130 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF2 ( 131 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF3 ( 132 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF4 ( 133 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF5 ( 134 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF6 ( 135 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF7 ( 136 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF8 ( 137 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF9 ( 138 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF10 ( 139 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF11 ( 140 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF12 ( 141 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF13 ( 147 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF14 ( 148 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF15 ( 173 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF16 ( 174 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF17 ( 175 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF18 ( 176 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF19 ( 177 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF20 ( 178 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF21 ( 179 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF22 ( 180 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF23 ( 181 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| PF24 ( 182 )     | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| MSR ( 170 )      | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| 3278MSR ( 171 )  | ! format 2   | ! format 2  | ! format 2      | ! format 4    |
| CLEAR ( 145 )    | ! format 1   | ! format 1  | ! format 1      | ! format 4    |
| USM ( 152 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA1 ( 142 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA2 ( 143 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA3 ( 144 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA4 ( 150 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA5 ( 151 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA6 ( 162 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA7 ( 163 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA8 ( 164 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA9 ( 165 )      | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| PA10 ( 166 )     | ! format 1   | ! format 1  | ! format 2      | ! format 4    |
| CURSEL ( 149 )   | ! format 3   | ! format 3  | ! format 2      | ! format 4    |
| RESET ( 155 )    | ! format 1   | ! format 1  | ! format 1      | ! format 4    |
| STATUS ( 156 )   | ! format 5   | ! none      | ! none          | ! format 4    |
| CONNECT ( 157 )  | ! format 6   | ! none      | ! none          | ! none        |

NO COMMAND means that input is not due to a read type command sent by the application i.e. the application has 'only' sent input message, which is the normal mode of operation.

### 5.5.2 Status byte, SB

SB is coded as follows:



#### COMBINATIONS:

| printer bit: | subsidiary bit: | device:  | response to:          | status: |
|--------------|-----------------|----------|-----------------------|---------|
| on           | off             | printer  | print bit in WCC      | 0 - 4   |
| off          | on              | display  | print bit in CCC      | 0 - 3   |
| off          | off             | not used |                       |         |
| on           | on              | display  | suppressed by ADP3270 |         |

### 5.5.3 Input elements

#### ADDR (field ADDRESS) ( two Bytes):

ADDR is constructed, extracting the six least significant bits from the two bytes and concatenating them to a twelve bit entity.

#### CUR ( CURsor address; two Bytes):

CUR is constructed, extracting the six least significant bits from the two bytes and concatenating them to a twelve bit entity.

#### SBA (Set Buffer Address) ( SBA = 17 ; Byte ):



**A. AID CODES****PROGRAM FUNCTION CODES**

SEND = 129  
MSR = 170  
3278MSR = 171  
PF1 = 130  
PF2 = 131  
PF3 = 132  
PF4 = 133  
PF5 = 134  
PF6 = 135  
PF7 = 136  
PF8 = 137  
PF9 = 138  
PF10 = 139  
PF11 = 140  
PF12 = 141  
PF13 = 147  
PF14 = 148  
PF15 = 173  
PF16 = 174  
PF17 = 175  
PF18 = 176  
PF19 = 177  
PF20 = 178  
PF21 = 179  
PF22 = 180  
PF23 = 181  
PF24 = 182

**PROGRAM ATTENTION CODES**

PA1 = 142  
PA2 = 143  
PA3 = 144  
PA4 = 150  
PA5 = 151  
PA6 = 162  
PA7 = 163  
PA8 = 164  
PA9 = 165  
PA10 = 166

**RESET AND UNSOLICITED MESSAGE -CODES**

CLEAR= 145  
USM = 152  
RESET= 155

**APPLICATION SELECT ATTENTION CODE**

CONNECT = 157

**NO ATTENTION CODES**

NO AT = 160 display  
NO AP = 161 printer

STATUS:

STATUS = 156

**B. MESSAGE/ANSWER FORMAT SURVEY**

| <u>Operation</u> | <u>Message</u>                                                                                             | <u>Answer</u>                                                                                   |
|------------------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Sense            | 0<12                                                                                                       | status<br>0<br><app. sel. name ><br><app. sel. name ><br><app. sel. name ><br><app. sel. name > |
| Sense ready      | 0<12 + 2                                                                                                   | status<br>result<br>0                                                                           |
| Connect          | 4<12 + 4<br>irrelevant<br><app. sel. name ><br><app. sel. name ><br><app. sel. name ><br><app. sel. name > | status<br>result<br>unchanged<br>unchanged<br>unchanged<br>unchanged                            |
| Disconnect       | 4<12 + 8                                                                                                   | status<br>result                                                                                |
| Lookup device    | 4<12 + 12<br>irrelevant<br>0<16 + cu<8 + dev                                                               | status<br>result<br>unchanged<br>AID byte<br>status byte                                        |
| Reserve device   | 4<12 + 16<br>irrelevant<br>0<16 + cu<8 + dev                                                               | status<br>result<br>unchanged                                                                   |
| Release device   | 4<12 + 20<br>irrelevant<br>0<16 + cu<8 + dev                                                               | status<br>result<br>unchanged                                                                   |
| Wait ready       | 4<12 + 32<br>irrelevant<br>0<16 + cu<8 + dev                                                               | status<br>result<br>unchanged                                                                   |
| Lookup physical  | 2<12 + 14<br>irrelevant<br>line<16 + cu<8 + dev<br>irrelevant                                              | status<br>result<br>line<16 + cu<8 + dev<br>line<16 + cu<8 + dev                                |
| Lookup logical   | 2<12 + 12<br>irrelevant<br>irrelevant<br>line<16 + cu<8 + dev                                              | status<br>result<br>line<16 + cu<8 + dev<br>line<16 + cu<8 + dev                                |

| <u>Operation</u> | <u>Message</u>                            | <u>Answer</u>                                        |
|------------------|-------------------------------------------|------------------------------------------------------|
| Output message   | 5<12 + 0<br>first address<br>last address | status<br>halfwords transferred<br>bytes transferred |
| Input message    | 3<12 + 0<br>first address<br>last address | status<br>halfwords transferred<br>bytes transferred |

## RETURN LETTER

Title: RC8000/ADP3270 Input/Output  
process

RCSL No.: 991 10630

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

---

---

---

---

Do you find errors in this manual? If so, specify by page.

---

---

---

---

How can this manual be improved?

---

---

---

---

Other comments?

---

---

---

---

---

Name: \_\_\_\_\_ Title: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Date: \_\_\_\_\_

Thank you

RCSL Nr. 48-F 0093

..... **Fold here** .....

..... **Do not tear - Fold here and staple** .....

**Affix  
postage  
here**

**REGNECENTRALEN**  
af 1979

**Information Department  
Lautrupbjerg 1  
DK-2750 Ballerup  
Denmark**