
RCSL No: 31-D523

Edition: November 1978

Author: Jens Ramsbøl

Title:

RC8000 FPA MAIN PROCESS

Keywords:

RC8000, FPA, Main Process, External Processes.

Abstract:

This manual describes the operations of the FPA main process in RC8000.

(13 printed pages)

**Copyright © 1980, A/S Regnecentralen af 1979
RC Computer A/S
Printed by A/S Regnecentralen af 1979, Copenhagen**

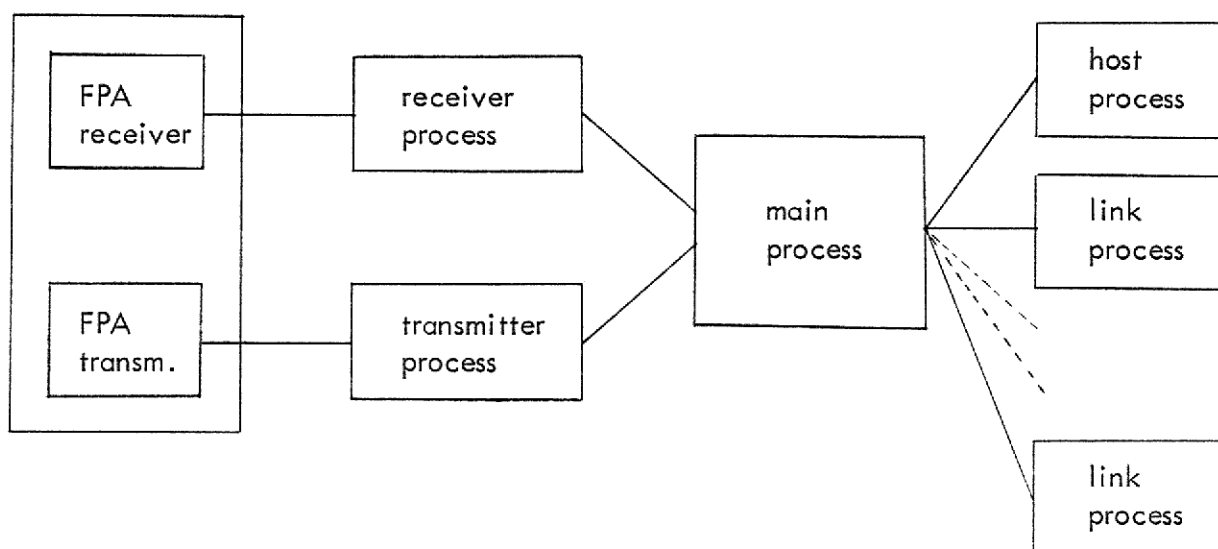
Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

References:

1. FPA801, Reference Manual
Villy Hansen 52-AA661
2. RCNET, Device Control Protocol
Erik Lilholt 43-RI0174
3. FPA Line Control Protocol
Jens Ramsbøl 31-D496

FPA Process Complex

The front end computers on RC8000 are normally connected to the main computer through a Front Process Adaptor (FPA). The communication between a front end and the RC8000 in the RC8000 monitor is supervised by a number of external processes - the fpa process complex - as shown below.



The FPA process complex.

These processes may be divided into three groups related to the different levels of FPA protocols. The link processes communicate by using the device control protocol, the main process by using the FPA line protocol, and the receiver and the transmitter processes work at the level of direct input/output operations (and sometimes at the line control level).

The internal processes in RC8000, which are communicating on the FPA line, are normally using the link processes (and therefore the device control protocol), but in some cases it is necessary to communicate at a lower level, i.e. when auto-loading a front end or when initiating the communication on the FPA line. For this purpose it is possible to perform operations directly at the main process level.

Main Process

The operations that can be executed by the main process are reset, master clear, autoloading, set mask, receive block, transmit block, and autoloading.

The reset and master clear operations are used to control the FPA line in connection with the use of the FPA line control protocol, the set mask is used to change the internal test mask of the main process, and the receive block, transmit block, and autoloading operations used to execute tasks at input/output level at the FPA receiver and transmitter.

Reset, master clear, and autoloading messages are accepted from internal processes that have reserved the main process, set mask messages from all in internal processes. Receive block and transmit block operations can be executed from internals that are reservers of the main process or - in case the main process is not reserved - of internals that have reserved the FPA receive process and the FPA transmit process relatively.

The main process can only be reserved when neither the FPA receiver nor the FPA transmitter processes are reserved, - and these processes can only be reserved if the main process is not reserved. Initialisation of the main, receiver or transmitter processes has no effect.

Reset

This operation resets the FPA process complex by returning all messages queued up at the link processes of this main process with result 'malfunction' (= 4). At the same time temporary link processes are removed.

Master Clear

The master clear operation is used to initiate the communication on the line according to the FPA line control protocol. The answer of the message is returned just before the start up is initiated.

The initialisation of the line is carried out in this way: the FPA process complex is reset as described above, a master clear header is transmitted and a statusbyte received, an accept master clear header is waited for and a statusbyte delivered (a detailed description is found in ref. 3). If any errors occur during this cycle the initialisation is repeated after a delay. The attempt to start up the line is repeated infinitely, - the delay is for the first 50 tries 0.1 sec., else 10 sec.

Set Mask

The FPA driver complex includes a testbuffer facility. At certain points in the drivers, testrecords will automatically be generated in case the testpoint is active. The states of the different testpoints are defined in the testmask, which is 96 bits defining the states of testpoint 0 to 95.

The operation set mask inserts a new mask in the FPA driver complex.

Receive Block

The receive block message operates on the FPA receiver device at io level. The function depends on the value of <mode> and <size>, in this way -

```
if <mode> and 2 = 2 then
    reset receiver
if <mode> and 1 = 1 then
    transmit statusbyte (message)
```

```

if <size> > 0 then
    receive startbyte (answer)
    receive data (first, size)

```

The statusbyte in the message and the startbyte in the answer are the rightmost 8 bits in the words. The <size> - in bytes (8 bits) - defines a buffer inside the area limited by <first> and <last>. The <size> in the message and the <number of bytes transferred> in the answer do not include the startbyte.

The value of <number of bytes transferred> in the answer specifies the number of bytes that have been received in memory, eventually less than <size> in the message. Unused positions in the last word (the rightmost bytes) are filled in with zeroes. The <number of halfwords transferred> holds the number of halfwords that have been changed in the memory. If the value of the word in the answer containing startbyte differs from -1 a startbyte has been received.

The operation of the message must be transferred within a certain time - the execution timeout (60 seconds) - else the operation is terminated and the device reset.

Transmit Block

The transmit block message operates on the FPA transmitter device at io level.

The function depends on the value of <mode> and <size> in this way -

```

if <mode> and 2 = 2 then
    reset transmitter and wait for 0.1 seconds
if <size> > 0 then
    transmit startbyte (message)
    transmit data (first, size)
if <mode> and 1 = 1 then
    receive statusbyte (answer)

```

The startbyte in the message and the statusbyte in the answer are the rightmost 8 bits in the words. The <size> - in bytes (8 bits) specifies a buffer inside the area defined by <first> and <last>.

The values of <size> in the message and <number of bytes transferred> in the answer do not include the startbyte.

In case no statusbyte has been received the word normally containing the statusbyte in the answer equals -1.

The execution of the operation must be finished within a certain time - the execution time (1 second) - else the operation is terminated and the FPA transmitter device reset.

Autoload

The FPA transmitter device includes a possibility of transferring an autoload signal to the FPA controller at the other end of the line, which normally will force the front end into its autoload state. This function is executed with an autoload operation that functions in this way -

```
if mode and 2 = 2 then
    reset transmitter and wait for 0.1 seconds
activate autoload
if mode and 1 = 1 then
    receive statusbyte (answer)
```

The statusbyte received is placed as the rightmost byte in the answer. If no statusbyte has been transferred this word equals -1. The timeout of this operation is 1 second.

Results of Operations

The answer to the different operations hold a result and eventually a status. The possible values and their interpretation is given below. In brackets are mentioned the reason for the error in question.

- result 1 ok, the status field contains further information, see below.
- result 2 rejected, sender has violated the rules of reservation.
- result 3 unintelligible, the value of operation or mode is illegal, or if the operation is receive or transmit block the buffer is outside sender's process, <first> is bigger than <last> or <size> specifies a buffer bigger than the one given by <first> and <last>.
- result 4 malfunctioning, can only occur in connection with receive block, transmit block, and autoloading. The channel program is terminated before end caused by some hard errors as front end disconnected (current status, bit 0), controller does not start (start io, result 1, 2), bus parity error (event status, bit 20), status transfer error (event status, bit 21), bus-time out (event status, bit 22) and bus communication error (event status, bit 23).
- result 5 unknown, no process of this name.

In case of result 1 the status holds further information on the success of the operation.

- bit 0 reset received (event status, bit 0), only after receive block. A reset operation has been executed from the other end of the FPA line. <bytes transferred> equals 0.
- bit 1 parity error (event status, bit 1), only from input operations. Receive block - one or more parity errors have been detected during reception of start byte or data. Transmit block, autoloading - parity error under reception of status byte.
- bit 2 timeout (event status, bit 2), only from output operations. Receive block - the status byte could not be transmitted, the execution continued with the reception of startbyte and data. Transmit block, autoloading - the transmission of startbyte and data was terminated because no receiver was ready, the execution continues with reception of statusbyte.
- bit 4 blocklength error (event status, bit 4), only under input. Receive block - the data block received was bigger than the buffer defined in the message. Only <size> bytes are transferred to memory, the remaining part are skipped.

- bit 14 execution timeout (io result 3). The operation is terminated and the device reset.
- bit 15 stopped, occurs only when transferring data. At the time of transmitting or receiving data the state of sender was stopped. Only an eventually reset operation (defined in mode) has been executed. <bytes transferred> equal to 0.

Formats of Messages and Answers

	Messages:	Answers:
<u>Reset</u>	4<12 + 0	0
<u>Master Clear</u>	8<12 + 0	0
<u>Set Mask</u>	12<12 + 0 mask (0 - 23) mask (24 - 47) mask (48 - 71) mask (72 - 95)	0
<u>Receive Block</u>	3<12 + mode first (core buffer) last (core buffer) size statusbyte	status halfwords transferred bytes transferred startbyte
<u>Transmit Block</u>	5<12 + mode first (core buffer) last (core buffer) size startbyte	status halfwords transferred bytes transferred statusbyte
<u>Autoload</u>	6<12 + mode	status 0 0 statusbyte
<u>Mode</u>		
mode and 2 (bit 10)	reset	
mode and 1 (bit 11)	{ transmit statusbyte (receive) receive statusbyte (transmit, autoload)	
<u>Status</u>		
bit no 0	reset received	
1	parity error	
2	time-out	
4	block length error	
14	execution time-out	
15	stopped	

RETURN LETTER

Title: RC8000 FPA MAIN PROCESS

RCSL No.: 31-D523

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

..... Fold here

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

Affix
postage
here

 **REGNECENTRALEN**
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark