Title:

RC8400 FPA testpackage
Test of frontendprocessor adapter, FPA801

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 30-M57

Edition: May 1977

Author: Flemming Hansen

Contents Page

Applicable documents:

RC8401 Front End Adaptor Manual Revision 2, RCSL: 52-AA610

RC8000 testprogram system, users manual, RCSL : 30-M98

RC8000 testprograms, Slanglisting, RCSL : 80-M56

RC8400, FPA801 Testprogram, source, RCSL : 30-M74

RC8400, FPA801 Testprogram, binary, RCSL : 30-M50

1. Testprograms, parameter values.

Standard conf. reliability test (test a).

paramno	text	default	min	max
000	select testprogram :	a	a	b
001	number of runs =	19	1	2xx23-1
015	datacheck ?	yes	-	-
016	first bufferword =	10262	10262	memtop
017	min blocksize (chars) =	1	1	1048576
018	max blocksize (chars) =	6900	1	1048576
019	first dataword (octal) =	0	2**23	2**23÷1
021	max messages pr. block =	10	1	1000

configuration requirements : minimal

Special configuration reliability test (test b).

paramno	text	default	min	max
000	select testprogram :	a	a	b
001	number of runs =	19	1	2xx23-1
013	deviceno. 1.fpa =	10	6	64
014	deviceno. 2.fpa =	20	6	64
015	datocheck ?	yes	-	-
016	first bufferword =	10262	10262	memtop
017	min blocksize (char) =	1	1	1048576
018	max blocksize (char) =	6900	1	1048576
019	first dataword (octal) =	0	2**23	2**23-1
021	max messages pr : block =	10	1	1000

Special configuration diagnostic test (testc).

paramno	text	default	min	max
000	select testprogram	a	a	c
001	number of runs	19	1	2xx23-1
002	testdevice 1. fpa	10	6	64
003	ok device 2. fpa	20	6	64
004	testloop	-1	-1	93
005	errormessages ?	yes	-	-
006	check result ?	yes	-	-
007	initiate testloops ?	yes	-	-
008	tcp-on ?	no	-	-
009	print device table ?	no	-	-

010	halt if error ?	no	-	-
011	bufferdescr in halfwords * 1024 (31,95,159,223) =	31	31	223
012	runs per loop =	1	1	2xx23-1

2. Configuration requirements : minimal +

2 FPA801 (coupled back - to - back).
accessories : 2 cables, CBL 316

3. Standard conf. reliability test. (test a).

Purpose

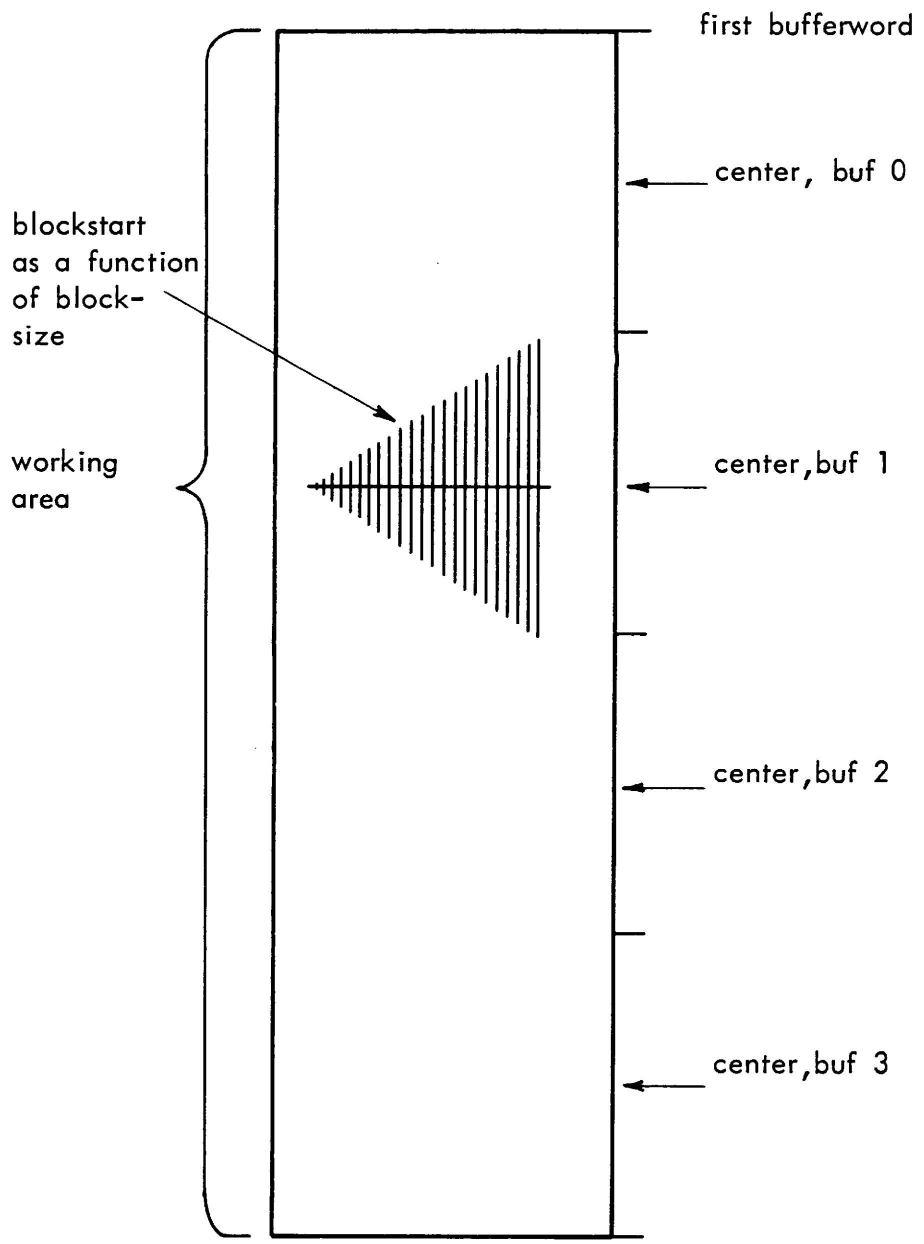
For verification of proper operation of the frontend link via FPA801 - FPA702 this test transfers testdata to a bufferarea in the RC3600, reads them back, and checks the contents.

Test strategy

The working area in the RC8000 memory is defined by the startaddress (first bufferword param) and the blocksize (bytes) param.
The size of the working area is computed as

$$w = (\text{blocksize} // 3 + 62) \times 4 \quad [\text{words}]$$

If the working area exceeds the free memory area the size is decreased accordingly.
The working area is split into 4 equal areas.
Four buffers are created around the centers of these areas, (see figure 1).
In the start of each buffer a buffer descriptor containing all variables concerning the buffer is found. The size of the bufferdescriptor is 30 words.



Working area layout for FPA801 reliabilitytests

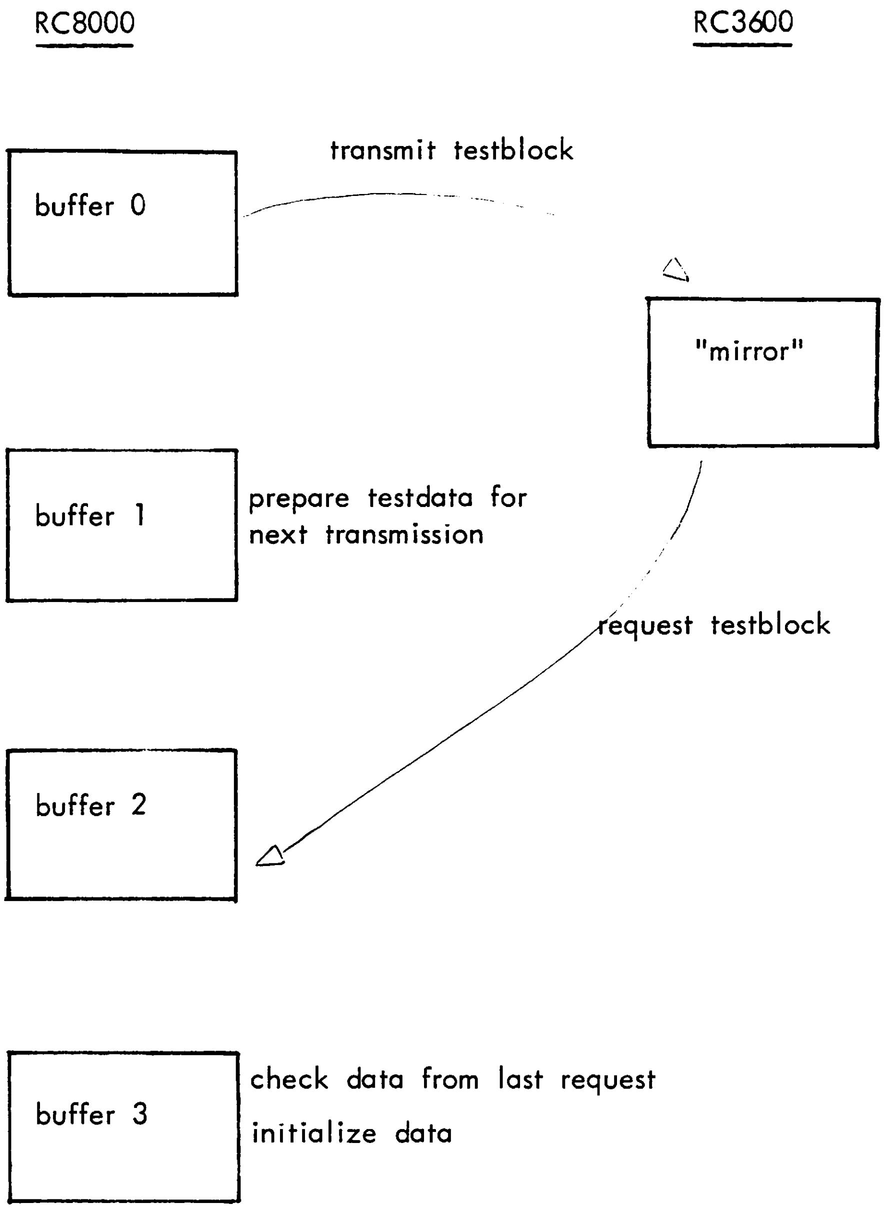
fig. 1

During each run a number of blocks is transferred to and from the RC3600 "mirror". The first block has a blocksize of one byte, the next of two bytes, and so on until the number indicated by the "blocksize" param is reached.
The buffer in the RC3600 "mirror" has a maxsize of 16400 bytes, (for the disc load version only 8200 bytes). If the blocksize specified is larger than this value it will be cut to the proper size.

For even blocksizes the data is transferred from buffer 0 to RC3600 mirror and back to buffer 2 or vice versa. Odd blocksizes are transferred to and from buffer 1 and buffer 3.

While data are transferred between mirror and one set of buffers, the other set of buffers are checked and cleared, respectively filled with new test data (see figure 2).

If the parameter datacheck ? is answered "no" the datachecking is skipped.



Example of datatransfers in test a

fig. 2.

The "mirror" is a buffer area in the top of the RC3600 memory. The data block is placed around the center of the area in a way similar to that in RC8000 (see fig. 1.)

Testdata

The data contents of the buffers is counting, (on word level) rounded to integer multiples of 32 by buffer change.
This means, that the first word in two consecutive blocks will be increased with at least 32 or an integer multiple of 32.

The FPA702 requires an extra byte before each block, the startbyte. The contents of this startbyte is the blocksize mod 256.

Before reading a block from the "mirror" the inputbuffer is initialized to a word contents of all zeroes alternating with all ones. This must be remembered in case of addressing errors where the contents printed may be this "background pattern".

Errormessages

If a dataerror is detected, a message containing coreaddress, blocksize, received and expected data is printed on selected output device.
In case of startbyte error a special message indicating "startbyte error" is output. The word used for receiving the startbyte is initialized to

.... IIII IIII IIII IIII

If this pattern is displayed as "received" it means, that nothing at all has been stored in this cell during the input transfer.

During upstart of the program it is checked that the blocksize wanted can be in the free core area. If the space is exceeded, the blocksize is cut to a proper size, a message informing of the altered values is output, and the program proceeds.

Remarks

As the test is executed on the controlling link of the testprogram, there will be a great risk, that an error may cause the test system break down - even before an error-message can be output.

As a consequence, the diagnostic value of this testprogram may be considered rather little, it should preferably be used to verify the proper operation of the system.

On the other hand, this testprogram may be run on any standard RC8000 and it may be considered an additional test of the FPA702 and the RC3600 memory.

No explicit statuscheck is performed by the test, as the blocktransfers are carried out by the standard testadm routines. The recovery procedures found here will take action on any status error condition.
Time consumption pr run is app. 31 minutes with a maximum blocksize of 16400 bytes.

If a blocktransfer goes completely wrong, there will be dataerror in nearly all words. To limit these error output messages, the parameter "max messages pr.block" defines how many bad words to be displayed.

4. Special conf. reliability test (test b)

Purpose

For workshop configurations, when two FPA801 can be coupled back-to-back this test performs a number of blocktransfers to various addresses with various blocklengths and datacontents.

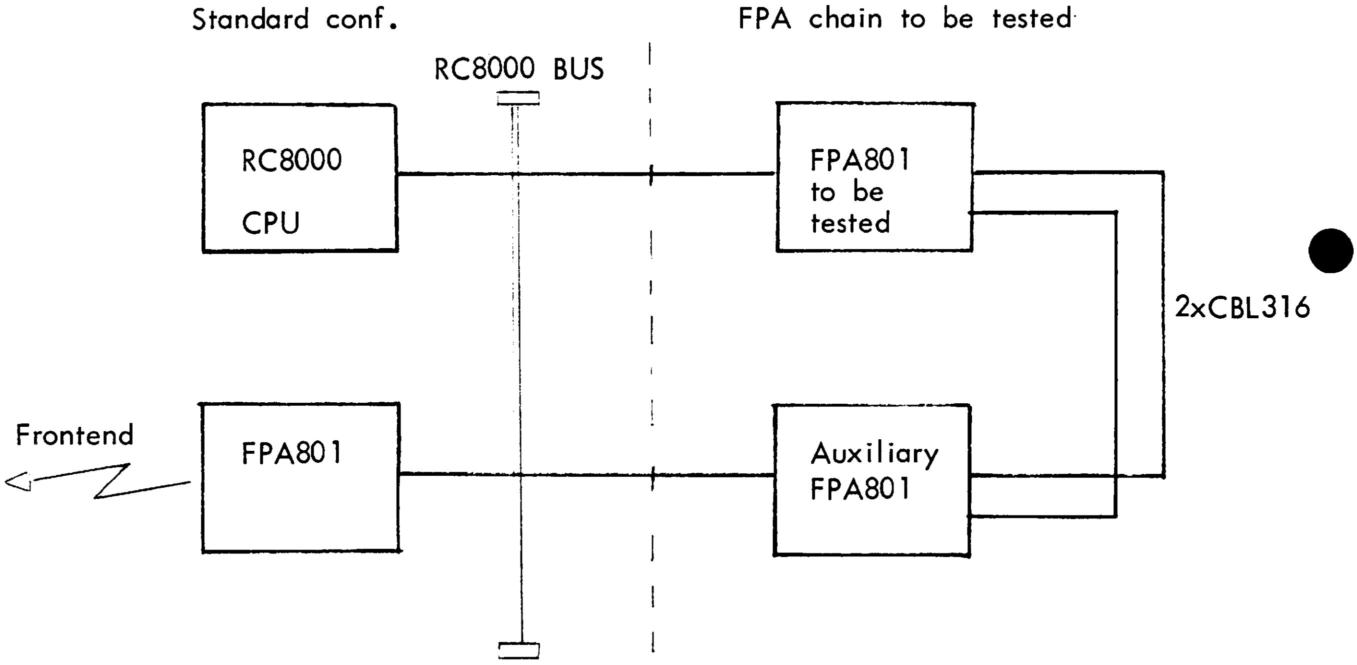


fig.3.

Test strategy

The working area in the memory is defined by the startaddress (first bufferword param) and the blocksize (bytes) param. The size of the working area is computed as

$$w = (\text{blocksize} //3 + 62)\times 4 \;\text{words}$$

If the working area exceeds the free memory area the size is decreased accordingly.

The working area is split into 4 equal areas. From buffers are created around the centers of these areas (see figure 1). In the start of each buffer a bufferdescriptor containing all variables, and channelprogram, concerning this buffer, is found. The size of the bufferdescriptor is 30 words.

During each run a number of blocks is transferred from one core area to another via the FPA chain. The first block has a blocksize of one byte, the next of two bytes and so on until the number defined by the "blocksize" param is reached.

For even blocksizes the datablocks are transferred between buffer 0 and buffer 2, for odd blocksizes between buffer 1 and buffer 3.

While data are transferred between one set of buffers, the other set of buffers are checked and cleared, respectively filled with new testdata. Figure 2 illustrates a typical situation, if the "mirror" box is replaced by the FPA chain.

For every 5 transfersthe direction is changed, that is for 5 transfers the direction is from buffer 0 to buffer 2 or from buffer 1 to buffer 3. The next 5 times the direction is the opposite. The transmitter or the receiver may be started at first. The sequence of this is inverted for every 7th transfer.

Parameters

The two devicenumbers to be specified must be in the range 4 - 64. They define the receiver address of the FPA's, that is, the numbers must be even (the transmitter device address is always one higher than the receiveraddress). The devicenumbers may be chosen arbitraryly, but a representative test should consist of two different runs carried out on the device-under-test having complementary deviceaddresses, e.g. 20 and 40. The devicenumber of the auxiliary FPA is unimportant (but must of course differ from that of the test-FPA).

A significant part of the CPU time is used for checking data contents and clearing buffer areas. These activities may be skipped if the "data-check?" parameter is set to "no". The initialization of the output buffers (filling with testdata) can not be suspended.

If a blocktransfer goes completely wrong, there will be dataerror in nearly every word. To limit the number of error output messages, the parameter "max messages pr. block" defines how many bad words pr. block that will be displayed.

Errorhandling

The FPA's are operated by the "do" instruction with an argument address of either start and reset type. If a bus-exception is detected upon the "do" instruction an errormessage is output and the operation is repeated. This will go on until the exceptions disappear.

When the FPA's have been started, an interrupt is expected signalling the end of the channelprogram, and status information available. The interrupt device specified in the channelprogram description does not describe the CPU but at core address (in the buffer descriptor). Here will be delivered a word containing the interrupt number.
If no such interrupt has been received for 2 seconds, the corresponding device is reset (by a do command with reset-address) and appropriate error-messages are output. The operation causing the timeout will not be repeated.

Before the device start is issued, the status area is initialized to -1 (all ones).
This must be kept in mind when inspecting statuserrormessages after no interrupt or similar situations.

Channelprogram

The channelprograms used are stored in the actual bufferdescriptors.. This descriptor is placed in front of the corresponding bufferarea.
This means, that every times the blocksize increases with two words the descriptor is moved 1 word downwards (ref. fig. 1).

The channelprogram is the simplest possible, i.e. one command followed by a stop command.
Status is transferred automatically before interrupt.

Test data

The data contents of the buffers are counting (on word level) in modules of 32 words.
This means, that the first word in two consecutive blocks will be increased with 32 or an integer multiple of 32.

Before a block is read to an input buffer the buffer area is initialized to words of all zeroes alternating with words of all ones.
This must be remembered in case of addressing errors, where the contents printed as "received" may be this "background pattern".

Operator Messages

bufferstart = < new bufferstart > halfwords

blocksize = < new blocksize > chars

the working area start address and size is printed before the test is started.

Errormessages

xxx devicenumber odd or equal

an odd devicenumber or two equal devicenumbers have been specified

xxx min blocksize > max allowed blocksize

the specified minimum blocksize is greater than the allowed.
the test terminates.

start exceptions, device < deviceno > < exceptions >

reset exceptions, device < deviceno > < exceptions >

after start or reset by means of a "do" instruction
bus exception has been detected < exception > may be:
bus parity, bus timeout or bus nack.

statuserror, device < deviceno >, < operation >, blocksize < blocksize >
< status kind >
received : < status received >
expected : < status expected >

An unexpected status has been detected after interrupt or timeout.
< operation > is either transmit or receive.
< status kind > will be one of the following :

channelprogramcounter
remaining chair
current status
event status

< status received > and < status expected > will appear as integers for the two first kinds of status. The last two kinds will be output as texts, or in special cases, in binary form.

Notice, that if < status received > contains all ones this usually means, that no status at all has been received.
In all four kinds of < status received >, are all ones the messages above are suppressed and instead the message : no status received is printed.

The possible status texts are

current status word :

disconnctd	(bit 0)
transmittr	(bit 21)
blckorient	(bit 23)

event status word :

resetrecvd	(bit 0)
lineparity	(bit 1)
timeout	(bit 2)
blcklngtrr	(bit 4)
busparity	(bit 20)
staxfererr	(bit 21)
bustimeout	(bit 22)
buscomeror	(bit 23)

possible other bits will be printed as a bitnumber.

device busy for 2 secs

the device has been started two seconds ago, and no interrupt has yet been received.

dataerror, coreaddr = <coreaddr> , blocksize = <blocksize> bytes
received <binary received> <decimal received>
expected <binary expected> <decimal expected>

Dataerror has been detected in the block. The datawords are printed both on binary and decimal form. Notice, that the buffer is initialized to a "background pattern" of words of all zeroes alternating with words of all ones. If this is printed as received data, no data at all has been received. If the number of bad words in that block exceeds the value of max. messages pr block, the rest of it is skipped. This situation is indicated by the symbol :

```
x  x
x  x
x  x
x  x
x  x
x  x
```

5. Special conf. diagnostic (test c)

Purpose

The test is intended to be used in workshop configurations, when two FPA's can be coupled back to back. The test examines all channel commands with various parameters in a convenient way for diagnostic testing.

copy of fig. 3

Test Strategy (operators guidance).

The test uses the start small philisophy. It consists of 93 independant testloops. In Appendix 1 is given a short description of each loop.

The program list contains information about:

1. the channelcommands used , and the expected status
2. the microprogram addresses used by the channelcommand
3. the chips tested

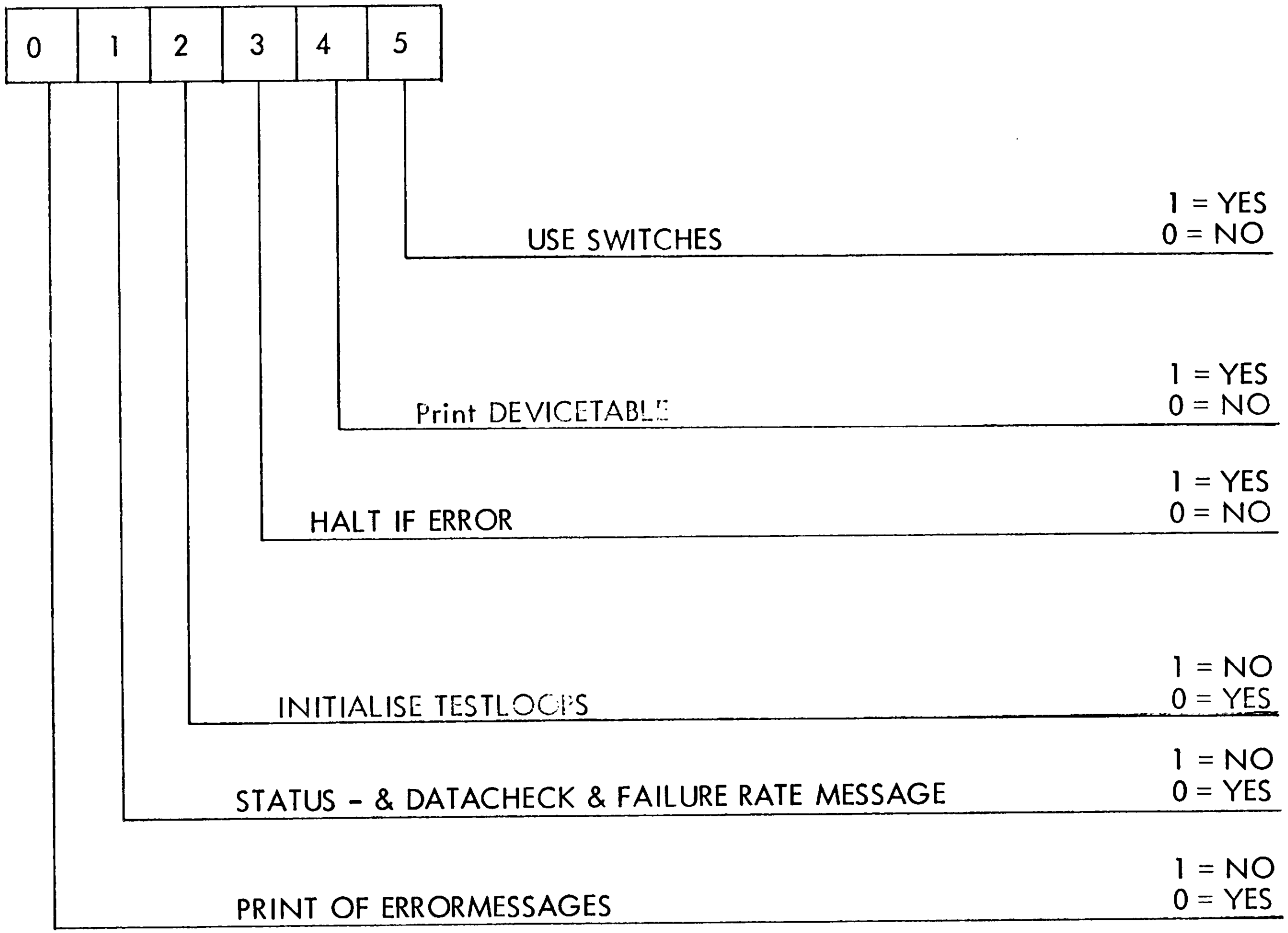
The test may fetch its parameters from either the RC3600 or from the TCP801. If the TCP-ON parameter is ''yes'' and RESEL(5) = 1 the testparameters are taken from the TCP801. The definition of the switchparameters on TCP is given on page 17. If an error is detected the testloopno is printed on DISP. The DSWR register corresponds to the looptest parameter. At the end of a testloop the TCP switches are examined and the testprogramparameters changed according to the switch-setting.

On the following pages the meaning of the parameters are defined.

When entering loopmode the number of the testloop is printed.
Hereafter for each 1000 loops the number of erroneous loops are printed (provided than an error is detected). If the parameters are taken from the RC3600, the parameter "runsprloop" determines the failure rate printing. see page 19.

If the TCP is mounted, the testloopno is displayed on the DSWR. If an error in that testloopno is detected the left hand 12 bits of DSWR are displayed as all ones.

REGSEL:



DSWR (corresponds to Testloop Parameter).

DSWR < 0 i.e. DSWR(0)=1 : GOTO TEST1 + CONTINUE MODE
DSWR = 0 : GOTO TEST(LOWEST ERRORNO) + LOOPMODE
DSWR = N : GOTO TESTN + LOOPMODE
(1 =< N < = 93)

Parameters

Devicenumbers

The two devicenumbers must be specified in the range 4-64. They define the receiver address of the FPA's, that is, the numbers must be even, (the transmitter device address is one higher than the receiveraddress). The fpa-under-test is named'' testdevice'' The opposite fpa is named ''ok device''.

Testloop

The test consists of 93 independant testloops. The value of the testloop parameter determines the mode of operations:

Testloop > 0 : goto test1 + continue mode

The testloops are executed in succession.
Having executed test93, test 1 is selected as next test.
The lowest testloop number where an error is detected is remembered as lowest errorno.

Testloop = 0 : goto test (lowest errorno) + loopmode

The test with the lowest errorno is executed in loopmode.
If lowest errorno = 0 the test is operated as if testloop < 0.

Testloop = N : go to testN + loopno

The testhaving number N is executed in loopmode.
N must be in the range 1 to 93.

ERRORMESSAGES:

If the errormessageparameter is '' no '' the testoutput is not printed. In loopmode the failurerate message will still be printed.

Checkresult:

If the checkresult parameter is '' no '' then status,data, and interrupt check will be suspended and no failureratemessage will occur.

Initiate testloops

The testloops are initialized by reseting both FPA's with ''do'' instructions.
The eventstatus ''Attention'' is thereby set. It is cleared with the STOP channel command. This initialisation may be bypassed by setting initiate tesloop to ''no'' thereby speeding the test and assuring a better picture on the oscilloscope.

TCP-ON

If this parameter is ''yes'' and REGSEL(5) = 1 the parameters to the test are operated from the DSWR and REGSEL registers as specified in chapter.

Print Device Table

If this parameter is ''yes'' and an error is detected, the following information is printed before the channel commands. see page 23.

```
DEVICE TABLE :
CHANPRGADDRESS = 8. < address in octal > HALFWORDS
STATUSAREAADDRESS = 8. < address in octal > HALFWORDS
ITR DESTINATION = 8. < address in octal > HALFWORDS
ITR LEVEL = < binary pattern >
```

Halt if error

If this parameter is ''yes'' and the testprogram is operated from the RC 3600 and an error is detected the program terminates.
If the program is operated from the TCP 801 and an error is detected the testprogram is set in a waiting condition until the halt if error switch is set to ''NO''.

Runs pr. Loop

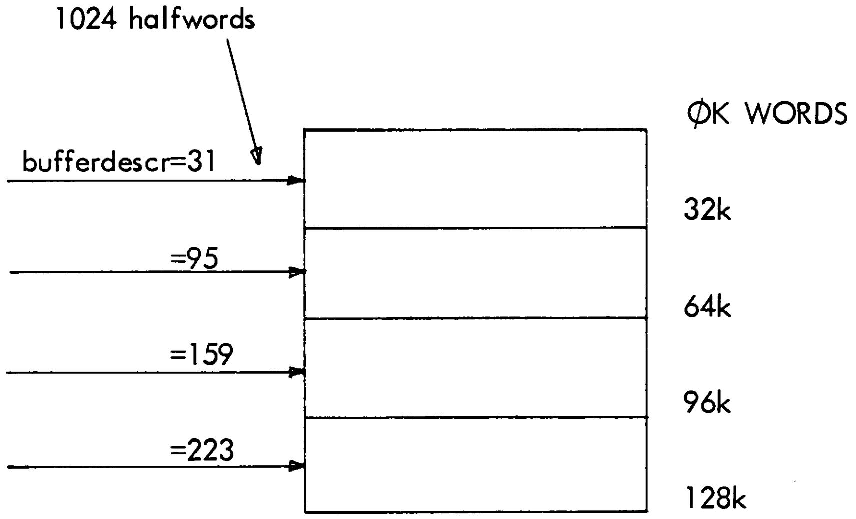
The parameter specifies the number of runs between the failureratemessages:

FAILURATE = < No of ERRONEOUS RUNS > per < MAXRUNS >

In continue mode this message is printed if an error is detected and runsprloop > 1, maxruns is set to runsprloop.
In loopmode maxruns is set to 1000 if runsloop < 100 else it is set equal to runsprloop, the failureratemessage is only printed if an error is detected.

Bufferdescr in halfwords * 1024 (31, 95, 159, 223).

The channel commands, the status- and dataarea's for each FPA are located in 2 buffers containing 36 words. The position of the buffers may be chosen within different memory moduls as follows:



Device table.

The device table for each FPA device is printed out by device tabledecsr:
= 8 * deviceno - 4.
It contains four words:

1. Channelprogram address
2. First address of status area
3. Interrupt destination
4. Interrupt level.

If the parameter print device table is ''yes'' this table is printed as a comment in an errormessage. see.p. 23

Test data

The data words used in the read and write-commands are alternating ones and zeroes. The datawords are initialized to zero.

Errorhandling

The FPA'S are operated by the ''do'' slang instructions with an argument address of either start or reset type.
If a bus-exception is detected upon the ''do'' instruction an error message is output and the operation is repeated. This will gon on until the exceptions disappear.

When the FPA's have been started, an interrupt is expected signalling the end of the channel program, and that status information is transfered to memory.
The interrupt device specified in the channelprogram description does not describe the CPU (address = 1<23) but a core address to where the interrupt number in the device table is delivered.
If no such interrupt has been received within 2 seconds an appropriate errormessage is output.
Before the device start is issued, the status area is initialized to -1 (all ones).
This must be kept in mind, when inspecting statuserrormessage after no interrupt or similar situations.

Errormessages common for all testloops

If an error is detected in a testloop the following is printed

```
* * * * * * * * * * * * *
```

TESTNO = < testloopno >
<testloopNo> ::= 1,2,3..., 93
If the error is detected in the okdevice
OKDEVICE - < type > = < ok device no > is printed
If the error(s) are detected in the testdevice
TESTDEVICE- < type> = < test device no > is printed.
< type > :: = trm/rec
The following messages until the device is redefined belongs to this device.

If the print device table parameter is "yes" the following message is issued:

```
DEVICE TABLE:
CHANGPRGADDRESS = 8. < octal address >
STATUSAREAADDRESS = 8. < octal address>
ITRDESTINATION    = 8. < octal address >
ITRLEVEL          =    <binary number >
```

Now the channel commands executed by the device are printed as follows:

CHANNELCOMMANDS:

< command1 >, BUFFERADDRESS = 8. < address in octal > HALFWORDS, BYTESIZE = < NO >
< command2 >, x,x:
< command 1 >::= read/write/read+chain/write+chain/sense
< command 2 >::= stop/wait/dummy
< NO > is number of characters transfered.

Channel command	Number code
read	1 < 8
write	3 < 8
read+chain	1 < 8 + 1 < 7
write+chain	3 < 8 + 1 < 7
stop	7 < 8
wait	4 < 8
sense	0 < 8
control	6 < 8
dummy	2 < 8
dummy	5 < 8

Here after the error is specified as either startexceptions, resetexceptions, statuserror, interrupterror or dataerror. see following pages.

An errormessage preceded by *** specific for each testloop is printed, when the testloop is finished. see appendix 1.

START EXCEPTIONS: < exceptions >
RESET EXCEPTIONS: < exceptions >
    <exceptions > :: = busparity/bus timeout/bus nack

After a start or reset by means of a "do" instruction bus exceptions has been detected.

STATUSERROR:
<STATUSKIND>
RECEIVED:<status received>
EXPECTED:<status expected>
An unexpected status has been detected after interrupt or timeout.
< status kind > :: = channelprogramcounter/remainingchars/
                    /current status/event status
< status received > and < status expected > will appear as integers for the two first kinds of status. The last two kinds will be output as texts, or in special cases, in a binary form.
Notice that if < status received > contains all ones this usually means that no status at all has been received.
If all four kinds of < status received > are all ones the messages above are suppressed and instead the message "no status received" is printed.

The possible status texts are:

current status word:

disconnctd	(bit 0)
transmittr	(bit 21)
blckorient	(bit 23)

event status word:

resetrecvd	(bit 0)
lineparity	(bit 1)
timeout	(bit 2)
blcklngtrr	(bit 4)
bus parity	(bit 20)
staxfererr	(bit 21)
bustimeout	(bit 22)
buscomeror	(bit 23)

possible other bits will be printed as a bitnumber.

device busy for 2 secs
the device has been started two seconds ago and no interrupt has yet been received.

INTERRUPTERROR:

```
ITRDESTINATION = 8. < addr in octal > HALFWORDS
   RECEIVED: < binary received interruptno >
   EXPECTED: < binary expected interruptno >
```

An erroneous interrupt number has been received.
The " background " interrupt number is all zeroes.

DATAERROR:

```
COREADDR = 8. < addr in octal > HALFWORDS
   RECEIVED: < binary received >
   EXPECTED: < binary expected >
```

Dataerror has been detected. The datawords are printed on binary form. Notice, that the buffer is initialized to a " background pattern " of words of all zeroes. If this is printed as data, no data at all has been received.

Error messages to the operator.

*** devicenumber odd or equal

an odd devicenumber or two equal devicenumbers has been specified.
the test terminates

*** illegal bufferdescr

a bufferdescr <> 31,95,159, or 223 has been specified.
the test terminates

*** bufferdescr > memtop

a bufferdescr > memory top has been specified.
the test terminates

*** testno too big

a testno > 93 has been specified. The test terminates if tcp-no else it is awaiting a new number.

Operator messages:

FIRST FREE WORD = 8. < octal address > Halfwords.
MEMTOP = 8. < octal address > Halfwords.

before the test is initiated, the first free word not occupied by the testprogram and the memory top is printed.

Appendix 1 : Testloop description and specific errormessages

testno.	device.	description.	errormessage.
1	rec/trm	The power-up sequence is examined.	*** power-up
2 3	rec trm	It is examined if the FPA is addressed in a reset-command (data out, slang instruction).	*** reset
4	rec	It is examined if the FPA is started with a reset-command.	*** reset
5 6	rec trm	The stop-command is executed	*** stop
7 8	rec trm	The sense-command is executed	*** sense
9 10	rec trm	The write command is executed. 1 char is transfered. The opposite FPA reads 1 char.	*** write 1 read 1
11 12	rec trm	The read command is executed. 1 char is transfered. The opposite FPA reads 1 char	*** read 1 write 1
13 14	.rec trm	The write command is executed. 2 chars are transfered. The opposite FPA reads 2 chars.	*** write 2 read 2
15 16	rec trm	The read command is executed. 2 chars are transfered. The opposite FPA writes 2 chars.	*** read 2 write 2
17 18	rec trm	The write command is executed. 3 chars are transfered. The opposite FPA reads 3 chars.	*** write 3 read 3
19 20	rec trm	The read command is executed. 3 chars are transfered. The opposite FPA writes 3 chars.	*** read 3 write 3

testno.	device.	description.	errormessage.
21 22	rec trm	The write command is executed. 4 chars are transfered. The opposite FPA reads 4 chars.	*** write 4 read 4
23 24	rec trm	The read command is executed. 4 chars are transfered. The opposite FPA writes 4 chars.	*** read 4 write 4
25	rec/trm	Examination of the ''change device'' signal.	*** change device.
26 27	rec trm	It is examined if the FPA can fetch its channel commands everywhere in the memory	*** chanprgaddress = 8.<octal address> halfwords
28 29	rec trm	It is examined if the FPA can deliver ist status everywhere in the memory	*** statusareaaddress = 8.<octal address> halfwords.
30 31	rec trm	It is examined if the FPA can deliver its interupt-number everywhere in the memory	*** itrdestination = 8.<octal address> halfwords
32 33	rec trm	It is examined if the FPA can write its data everywhere in the memory	*** bufferaddress = 8.<octal address> halfwords
34 35	rec trm	It is examined if the FPA can read its data from everywhere in the memory	*** bufferaddress = 8. <octal address> halfwords
36 37	rec trm	It is examined if the FPA can deliver its sensestatus everywhere in the memory	*** senseaddress = 8.<octal address> halfwords
38	rec	The control command is executed	*** control
39 40	rec trm	The read + datachain command followed by another read command is examined.	*** read + chain 3 read 6

testno.	device.	description	errormessage
		The write + datachain command is executed,<wd> chars are transfered. It is followed by a write command, 6 chars are transfered.	*** write + chain <wd> write 6
41	rec	wd=1	
42	rec	wd=2	
43	rec	wd=3	
44	trm	wd=1	
45	trm	wd=2	
46	trm	wd=3	
47 48	rec trm	The wait command is executed. The device is started with a stopcommand.	*** wait + start
49	rec	The wait command is executed. The device is started by the reception of an ''ATTENTION'' signal	*** wait + reset
		It is examined if a bustimeout is generated, when ''addresskind'' is > = memtop.	*** timeout ''addresskind'' = 8. <octal address> halfwords
50	rec	addresskind = channelprogramaddress	
51	rec	addresskind = statusareaaddress	
52	rec	addresskind = senseaddress	
53	rec	addresskind = itr destination	
54	rec	addresskind = bufferaddress (in a write command)	
55	rec	addresskind = bufferaddress (in a read command)	

testno	device	description	errormessage
		It is examined that the event status "blocklength error" is generated.	
56	rec		*** read 1 write 2
57	trm		*** read 1 write 2
58	rec		*** read 1 write 3
59	rec		*** read 2 write 1
60	rec		*** read 2 write 3
61	rec		*** read 3 write 2
62	rec		*** read 3 write 4
63	rec		*** read 4 write 3
64 65	rec trm	It is examined that the event status "live timeout" is generated after 2,36 msch. The opposite FPA is idle and unassigned, while the test FPA writes 9 characters	duration of linetimer in 100 microsecs : < number > *** line time-out
66	rec	as above, but 1 char is written	*** timing
67	rec	as above, but 2 char are written	*** timing
		In this test the opposite FPA is idle and unassigned, while the test FPA executes the write+datachain command (WD chars are written) followed by a write command (W chars are written). The remaining characters count is expected to be = (WD÷2)+W.	
68	rec	WD = -1447446 W= 460552	*** line time out + addition
69	rec	-6381921 7368818	*** line time-out + addition
70	rec	6710887 -6710886	*** line time-out + addition
71	rec	-1184274 -2039582	*** line time-out + addition
72	rec	-2171169 921193	*** line time-out + addition
73	rec	as above but WD = 2 W=D	*** timing

testno	device	description	errormessage
		In this the test FPA executes the read + datachain command (rd chars are read) followed by a read command (r chars are read). The opposite controller executes the write+datachain command (1 char is written) followed by the write command (1 char is written) The remaining character count is expected to be =(rd-1)+r÷1.	
74	rec	rd = -1447446 r = 460552	*** line time-out + addition
75	rec	-6381921 7368818	*** line time-out + addition
76	rec	6710887 -6710886	*** line time-out + addition
77	rec	1184274 -2039582	*** line time-out + addition
78	rec	-2171169 921103	*** line time-out + addition
79	rec	It is examined, if the event status signal ''attention'' is set when the opposite FPA is reset.	*** attention
80	trm	It is examined, if the ''reset-out'' signal is generated	*** reset-out
81	rec	It is examined, if the ''attention'' signal can be cleared.	*** clear attention
82	trm	It is examined, if the ''reset-in'' signal from the opposite receiver is low.	*** reset in
83	rec	It is examined, if the ''reset-out'' signal is low for the receiver	*** reset-out

testno	device	description	errormessage
		The microprogram adds 2 to the < addresskind > in order to address a new location. This is tested by using the following addresses (< memtop) 0000 0000 1000 0000 0000 0010 0000 0000 1000 0000 0000 0110 0000 0000 1000 0000 0000 1110 ⋮ 0000 0011 1111 1111 1111 1110	*** carry < addresskind > 8. < octal address > halfwords
84	rec	addresskind = channelprogram address	
85	trm	= " " " "	
86	rec	= statusareaaddress	
87	trm	= " " "	
88	rec	= senseaddress	
89	trm	= " "	
90	rec	= bufferaddress (write - command)	
91	trm	= " " " " " "	
92	rec	= bufferaddress (read - command)	
93	trm	= " " " " " "	

Appendix 2 :

Testprogram Structure .

The program is written in the PATAC (Pseudo Algol To Assembler Code) language. (see ref.: RC SL 30-MXX)
Important: In arithmetic and logical expressions brackets are not allowed - there is no operator hierarchy -; all expressions are evaluated from left to right.

Examples:

A:= B + 5 // 6 + 2 corresponds to
A:= ((B+5) // 6) + 2.

if -, A or B and C corresponds to
if -, ((A or B) and C).

The program structure is as follows:

	page
1. Parameters	1 - 17
2. Variable declarations	17 - 18
3. Textstrings	19 - 22
4. Procedure declarations	22 - 79
5. Program initialization	80 - 82
6. Program A, B	83 - 84
7. Program C (93 testloops)	85 - 137