RC5040/RC5050

Central Processing Unit

Operating Guide

PN: 99001343

Keywords:
RC5000, RC5040. RC5050, Operation.

Abstract:
This document contains information necessary for daily operation of the RC5040 and RC5050 Central Processing Units including a description of front panel switches and indicators, console commands, tests, error messages and autoload procedures. In addition key information is found in the 17 appendices.

Date:
January 1991

Author:
Harald Villemoes

Disclaimer
RC International makes no representations or warranties with respect to the contents of this publication and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, RC International reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of RC International to notify any person of such revision or changes.

TABLE OF CONTENTS

Page

Appendices

1. FOREWORD

This edition of the RC5040 / RC5050 Central Processing Unit Operating Guide is based on components of the following versions:

SW50020,	RC5000 Operating System release 5.0 90.04.01
RC5040,	DEBUG Firmware ROB584 87.09.23
RC5050,	Microprogram, version 5.09 Service Processor, version 5.17 Service Processor Initiator, version 5.16
F505A,	Autoload PROM version 11.3 .

RC5000 MegaSwitch autoload is included.

The chapter describing the functionalities of the autoload switches has been rearranged.

January, 1991

2. SWITCHES AND INDICATORS

This chapter describes the switches and indicators found on:

- the operators control panel
- the front panel of the RC5040 and RC5050 processor boards
- the power supply

2.1 Operator's Control Panel

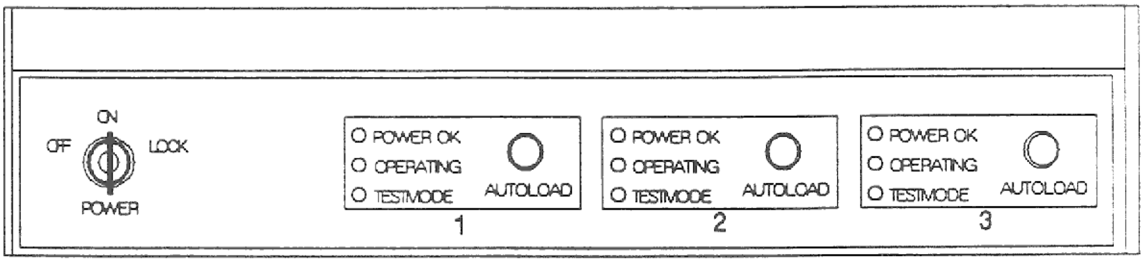


Fig. 2.1. OCP for Rack with up to Three RC5000

Power off of the RC5000(s) is done by turning the power key to the OFF position.

Power on of the RC5000(s) is done by turning the key to the ON position (or further on to the LOCK position).

The AUTOLOAD button(s) is (are) enabled when the key is in the ON position, and disabled, when in the LOCK position.

The AUTOLOAD button initiates autoloading of the RC5000 in question.

The POWER OK indicator is illuminated during power OK condition on the RC5000.

The OPERATING lamp indicates that the RC5000 is running normally.

The TEST MODE lamp indicates that the RC5000 is executing the built-in test programs.

2.2 RC5040 Processor Front Panel

The front panel of the RC5040 processor boards contain five switches, five indicators, and a jack.

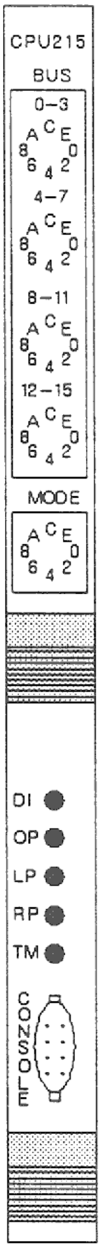


Fig. 2.2. RC5040 Processor Front Panel

2.2.1 RC5040 Switches

All of the switches are rotary switches with 16 positions, indicated by the hexadecimal numbers 0 to F. The switches are set by means of a screwdriver.

2.2.1.1 RC5040 Bus Switches

The four switches marked BUS are used to supply the processor with data. There is a switch for bits 0 to 3, 4 to 7, 8 to 11, and 12 to 15.

2.2.1.2 RC5040 Mode Switch

The switch marked MODE is used to control the baud rate for the console and the execution of the builtin test programs (sect. 3.2.2).

If the mode switch is equal to, or greater than 8, the console is locked to Terminal-mode (T-mode), i.e. the console will not switch to Debug-mode (D-mode) by activating the BELL key (CTRL G). The mode switch is read continuously, so the baud rate may be changed on a running machine. The current value may be examined by the display command Y3F (see 3.2.2).'

Settings		Baud Rate	Execution Mode
0	(8)	300 bps	run test, loop
1	(9)	1200 bps	run test, loop
2	(A)	300 bps	skip test
3	(B)	1200 bps	skip test
4	(C)	300 bps	run test, no loop
5	(D)	1200 bps	run test, no loop
6	(E)	300 bps	skip test
7	(F)	1200 bps	skip test

Test Program Execution Modes

run test The test programs are executed whenever the autoload button is pressed.

skip test The test programs are not executed.

loop The test programs are executed in an endless loop.

no loop The test programs are executed once.

2.2.2 RC5040 Indicators

DI Disabled Interrupt

This lamp, when lit, indicates that the processor is running in the disabled interrupt mode.

OP Operating

This lamp, when lit, indicates that the processor is running normally; when it is extinguished, the processor has stopped.

LP Left Parity Error

This lamp, when lit, indicates that a parity error has been detected during a memory read in the left byte. The lamp can be extinguished only by autoloading.

RP Right Parity Error

This lamp, when lit, indicates that a parity error has been detected during a memory read in the right byte. The lamp can be extinguished only by autoloading.

TM Test Mode

This lamp, when lit, indicates that the processor is executing the built-in test programs. The current program is indicated by the DI, OP, LP, RP, and TM lamps, TM representing the least significant bit of the program number.

If an error is detected by a test program, one of the following messages is displayed on the console. The number in front of the following text is the number shown by the indicators DI through TM:

1 8085 Communication Test

Message: ERR00 <dummy><dummy><dummy>
Y68 gives 6 bytes of transmitted data.
Y70 gives 6 bytes of received data.
Y8B is zero after debug/microprogram communications timeout.

3 Interrupt Test

Message: No message.

Test microprogrammed interrupt of control microprocessor. RP and TM are lit.

5 Working Register Address Test

Message: ERR02 <address><errdata><04>
OK data = address.

7 Working Register Data Test

Message: ERR03 <address><errdata><sub>

sub = 01: if lsb.addr = 0 then OK data = AAAA
else OK data = 5555

sub = 02: if lsb.addr = 0 then OK data = 5555
else OK data = AAAA

7 8085 EPROM Sum Test

Message: ERR0B <expected sum><computed sum><dummy>

9 Memory Address Test

Message: ERR04 <address><errdata><sub>
Y40 gives the module number under test.

The test will read both by means of word and by byte read. In the latter case OK data is the byte contents of the address read by byte read.

sub = 02: right parity error = R
04: left parity error = L
06: left and right parity error = LR
41: dataerror
43: dataerror + R
45: dataerror + L
47: dataerror + LR

B Memory Data Test

Message: ERR05 <address><errdata><sub>
Y40 gives the module number under test.
sub = 02: right parity error = R
04: left parity error = L
06: left and right parity error = LR
40: dataerror, okdata = AAAA in addr 0000
41: dataerror, okdata = 5555
42: dataerror + R, okdata = AAAA
43: dataerror + R, okdata = 5555
44: dataerror + L, okdata = AAAA
45: dataerror + L, okdata = 5555
46: dataerror + LR, okdata = AAAA
47: dataerror + LR, okdata = 5555

The test will write alternating AAAA,5555.

D Internal Interrupt Test

Message: ERR06 <low,high><errdata><04>
low = byte with lowest interrupt
high = byte with highest interrupt

OK data = high.

F Schedule Test

Message: ERR07 <param1><errdata><sub>

sub = 01: no external interrupt
02: maperror, okdata = 07FF
03: external interrupt or missing "interrupt chain end plug"
04: maperror, okdata = 0007
05: coroutine error, okdata = 0007
06: coroutine error, okdata = param1
07: medium priority error, okdata = param1
08: low priority error, okdata = param1
09: high FF, okdata = 000F
0A: mediumFF, okdata = 0017
0B: low FF, okdata = 001F

11 Interrupt Map Test

Message: ERR08 <address><errdata><04>
okdata = address.

13 Prefetch Test

Message: ERR09 <address><errdata><sub>

sub =	01:	load of ICD,	okdata = 5555
	02:	load of ICD,	okdata = AAAA
	03:	nxtbyte read ICD,	okdata = AAAB
	04:	nxtword read ICD,	okdata = AAAD
	05:	nxtbyte read ICD,	okdata = address
	06:	read of nxtbyte,	okdata = address and OFF
	07:	ICD,	okdata = address
	08:	nxtword even ICD,	okdata = address
	09:	nxtword,	okdata = address
	0A:	odd addr,	okdata = address

15 Register Stack Test

Message: ERR0A <param1><errdata><sub>

sub =	01:	not stack limit	
	02:	stack limit	
	03:	size error,	okdata = 0016
	04:	stack limit	
	05:	stack limit	
	06:	data error,	okdata = param1+7
	08:	stack limit	
	09:	stack limit	
	0A:	data error,	okdata = param1+7

2.3 RC5050 Processor Front Panel

The front panel of the RC5050 processor board contains 5 hexadecimal LED digits, 3 push buttons and a console jack.

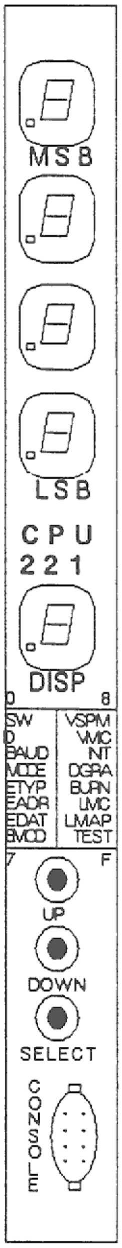


Fig. 2.3. RC5050 Processor Front Panel

2.3.1 Hexadecimal LED Display

The hex display will, at any time during normal operation, indicate both the display mode and the display data type.

2.3.1.1 RC5050 Display Mode

The display mode (i.e. the result caused by pressing the SELECT, UP or DOWN buttons, see section 2.3.2) is indicated by the decimal dots in the hex display.

Idle is indicated by

> one dot walking slowly from data digit to data digit in the hex display in direction from LSB to MSB (processor running)

or

> all dots blinking together (processor stopped).

Walking / blinking frequency is 1 step per second.

This display mode indicates that the UP and DOWN buttons are disabled.

One walking dot indicates that the processor is running, while all dots binking indicates that the processor is stopped.

Type Select is indicated by one fast blinking dot in the DISP digit. The blink frequency is 5 blinks per second.

This display mode indicates that the UP and DOWN buttons may be used to increase and decrease the value of DISP, thereby selecting the type of data being displayed.

Data Update is indicated by one fast blinking dot in one of the hex data digits. The blink frequency is 5 blinks per second.

This display mode indicates that the UP and DOWN buttons may be used to increase and decrease the value of the presently selected hex data digit, thereby updating the data being displayed.

2.3.1.2 RC5050 Display Data Type

The type of data displayed at any time is indicated by the lowest of the 5 hexadecimal LED digits. This digit is marked DISP.

The four upper display digits are used to display data. These are marked MSB to LSB, where MSB holds the most significant digit and LSB the least significant digit.

Below table describes the 16 display data types.

A plus sign in the column marked U means that the data displayed can be updated using the push buttons as described in section 2.3.2 below.

DISP	MSB			LSB	U	Meaning
0	x1	x2	x3	x4	+	Autoload Switches equal (x1 x2 x3 x4) The autoload switch value may be changed either by using the Push Buttons as described in section 2.3.2 below, or by setting the NVM value by the I N command, see appendix N.
1	x1	x2	x3	x4	+	ID-register equal (x1 x2 x3 x4) This ID-register value will, however, be overruled by a physical ID-register such as F506 (IDR202). This value may be changed as described for DISP 0.
2	b1	b2	b3	b4	+	Console baudrate equal (b1 b2 b3 b4) This value may be changed as described for DISP 0.

DISP	MSB			LSB	U	Meaning
3	d	t	l	m	+	Mode m: Console and Test mode, see section 2.2.1.2 . Console baudrate, however, is not affected by the value of m. d: d => console enabled, - => console disabled t: T => test enabled, n => test disabled l: L => testloop enabled, - => testloop disabled This value may be changed as described for DISP 0.
4	t1	t2	e1	e2	-	Hard Error Code The detection of a hard error will cause the display to display mode 4 with the mode digit blinking. Error type (e1 e2) indicates the type of error, numbers between 00 and 1F indicates testprogram number e1 e2 while other numbers indicates runtime errors. Error param (y1 y2) further describes the error, see below section 2.3.1.3 for a detailled description of all error displays.
5	a1	a2	a3	a4	-	Latest error addr (a1 a2 a3 a4)
6	d1	d2	d3	d4	-	Latest error data (d1 d2 d3 d4)
7	m1	m2	a1	a2	-	Latest error module address (m1 m2)

DISP	MSB			LSB	U	Meaning
8	r1	r2	r3	r4	-	Service Processor revision
9	r1	r2	r3	r4	-	MicroProgram revision
A	t	x1	x2	x3	-	SP Initiator test error t= 1 => RAM error at addr (x1 x2 x3 0) t= 2 => NVM test t= 3 => Flash test
B	br	nv	f0	f1	-	SP initiator degraded mode br: console baudrate 1: 110, 2: 300, 3: 600, 4: 1200, 5: 2400, 6: 4800, 7: 9600, 8: 19200 nv: nvm status (0: checksum error, 1: checksum ok) f0: flash0 (0: checksum error, 1: checksum ok) f1: flash1 (0: checksum error, 1: checksum ok)
C	ph	-	a1	a2	-	SP initiator EEPROM programming ph: programming phase 0: init, 1: data load, 2: programming) (a1 a2) current load address
D	-	-	a1	a2	-	MicroProgram load, addr (a1 a2)
E	-	-	a1	a2	-	MicroMap load, addr (a1 a2)
F	t1	t2	e1	e2	-	Test Mode, test (t1 t2), error (e1 e2) see below.

2.3.1.3 Test Program error display codes

Errors may be detected by either the builtin test programs or by hardware or microprogram during normal operation. In both events the processor stops and displays an error code at the hexadecimal display and on the debugger console. In addition the error code with parameters is stored in the Non Volatile Memory (nvm).

Below table describes the different error codes with their meaning.

test/ error	type	addr	data	description
00	aa	-	-	Configure memory, module aa. Done by SP.
01	-	d1	d2	SP / MP communication test. If d1 equals 0 then d2 contains data found, which should have been $ffff. If d1 differs from $ffff then d1 and d2 contains data found, which should have been $55aa and $aa55, respectively.
02	-	a	d	Register address test. At address a was found data equal d, and data equal a was expected.
03	-	a	d	Register data test. At address a was found data equal d, and data equal either $5555 or $aaaa was expected.
04	-	a	d	Working Register address test. At address a was found data equal d, and data equal a was expected.

test/ error	type	addr	data	description
05	-	a	d	Working Register data test. At address a was found data equal d, and data equal either $5555 or $aaaa was expected.
06	-	-	d	Gate Array Register test. Gmode/Gstat register data error, data d found, data $5555 expected.
07	-	a	d	Scheduler RAM test. At scheduler RAM address a data d was found, ($1ff8 -a) div 8 and 7 was expected.
08	-	exp	found	Scheduler NextSet test. SchRAM(1):= 1; SchRAM(2):= 2; SchRAM(3):= 4; Scheduler delivered found, not exp.
09	typ	intno	Rl	Interrupt Chain test. Type 1, Rl = 2000 : Interrupt 7 cannot be set. Type 1, Rl < 100 : Interrupt Rl cannot be cleared. Type 2 : Interrupt chain disconnected.
0A	Type	addr	data	Memory address test. Error found at <errormodule>:<addr>, found <data>. Type = GateArray ErTyp register or'd with $10, see GA error code section.

test/ error	type	addr	data	description
0B	Type	addr	data	Memory data test. Error found at <errormodule>:<addr>, found <data>. ErrType = GateArray ErTyp register or'd with $10, see GA error code section.

test/ error	type	addr	data	description
0C	Sub	d1	d2	Hardware Stack test. Sub Error 1 Cs does not remove StkLim, d1 and d2 undefined. 2 Cs, Pop does not set StkLim, d1 and d2 undefined. 3 Cs, PopB does not set StkLim, d1 and d2 undefined. 4 Less than 9 Push's set StkLim, d1 and d2 undefined. 5 9 Pushs does not set StkLim, d1 and d2 undefined. 6 Less than 9 Pop's set StkLim, d1 and d2 undefined. 7 Data error during Pop after Push d1 = expected, d2 = found. 8 Less than 9 PopB's set StkLim, n/a 9 Data error during PopB after Push, n/a a Less than 9 PushB's set StkLim, n/a b Less than 9 Pop's set StkLim, n/a c Data error during Pop after PushB, n/a d Less than 9 PopB's set StkLim, n/a e Data error during PopB after PushB, n/a f Ssize error, d1 = expected size, d2 = Ssize found 10 Stack(Radr) integrity Test data error, d1 = expected data, d2 = data found 11 Stack(Radr) integrity Test size error, d1 = expected size, d2 = Ssize found

test/ error	type	addr	data	description
0D	0	-	v	Microsecond Timer test The microsecond timer value v is outside the range 199 - 201 after 2000 microprogram steps.
0E	Sub	d1	d2	FIFO test. Sub Error 1 Fe (FIFO empty) status after one Push, d1 = d2 = undefined 2 Data error at first 7 Pop's d1 = data expected, d2 = data found 3 Premature Fe (FIFO empty) status, d1 = d2 = undefined 4 Missing fe after 8 Pop's, d1 = d2 = undefined 5 Data error at 8th Pop, d1 = data expected, d2 = data found

test/ error	type	addr	data	description
0F	Sub	d1	d2	PreFetch test Sub Error 1 Even addr Nxb (d2) differs from expected (hi(d1)) 2 Icd (d2) differs from expected (d1) 3 Smem NxbI byte counter (d1) differs from expected (d2) 4 Odd byte Nxw (d2) differs from expected (d1) 5 Odd byte Nxb (d2) differs from expected (d1) 6 Even byte Nxw (d2) differs from expected (d1) xx Where xx has bit 4 ($10) set indicates a GateArray error xx, see GA error code section.
1F	-	-	-	Microprogram communication error. The Service Processor received a request from the microprogram with request code equal zero.
20	type	addr	data	Gate Array detected error. The Gate Array error code is found in <type>, see below description. The module is <addr>, if relevant, and the displacement is <data>, if relevant.
21	-	inst	reg	Stack Limit error The hardware stack for registerset <reg> has over- or underflowed during execution of an instruction with code <inst>.

test/ error	type	addr	data	description
22	-	-	-	Unchain Register error An unchain registerset operation terminated either because the registerset were not part of any chain (CH = $FFFF) or because the registerset were not found as member of the chain.

2.3.1.4 Gate Array Error Codes

An Gate Array error code is a bit array, where each bit indicates an error.

Bin	Hex	Meaning
1xxxxxxx	$80	Addressing non-existing memory.
x1xxxxxx	$40	I/O timeout during IORS or IORW.
xx1xxxxx	$20	Byte operation.
xxx1xxxx	$10	This bit does not indicate an error and may be ignored.
xxxx1xxx	$08	n/a.
xxxxx1xx	$04	Left Parity error.
xxxxxx1x	$02	Right Parity error.
xxxxxxx1	$01	RC3502 Bus Timeout. /read or /write exceeds 16 uSec.

E.g.: $A0 means Addressing non-memory with byte access. $B0 has same meaning.

2.3.2 Push Buttons

The RC5050 Front Panel has three push buttons located below the hex display marked: UP, DOWN, and SELECT.

The hex display will at any time during normal operation indicate both the display mode and the display data type.

Pressing the SELECT button will change the display mode in a cyclic way:

```
Idle Mode                     to
Type Select                   to
Data update digit 1 (MSB)     to
Data update digit 2           to
Data update digit 3           to
Data update digit 4 (LSB)     to
Idle Mode
```

Result of pressing the UP or DOWN button depends upon the current display mode as indicated at any time during normal operation by the hex display decimal points.

The display data type (i.e. the type of data presently displayed) is indicated by the DISP digit as described in section 2.3.1.2 above.

The display mode (i.e. the result caused by pressing the UP or DOWN buttons) is indicated by the decimal dots in the hex display as described above.

Pressing one push button during power-up will cause the Service Processor Initiator to refrain from entering normal execution and at the same time select a specific console baud rate:

Push button	Result
UP	Select 19.200 baud
DOWN	Select 9.600 baud
SELECT	Select 1.200 baud

2.4 POW207 Power Supply

The power supply POW207 is supplied with the following controls:

POWER: Circuit breaker, lit when power on.

POWER OK: Indicator which is illuminated during power ok condition.

POWER FAILURE:
OVER-TEMPERATURE:
OVER-VOLTAGE: Error indicators which are illuminated after an error condition. These indicators are reset after activating the circuit-breaker later than 10 sec. after power off, or after activating the RESET pushbutton.

RESET: Push-button for manual generation of an autoload signal and a reset of the error indicators. When this control is activated, the voltage to the crate will be turned off. Use the Autoload button on the OCP to generate an autoload signal without turning off the power.

2.5 POW240/241 Power Supply

The power supplies POW240 / POW241 are supplied with the following controls:

PHASE NO
SLAVE NO Hex switches used for setting the load levelling phasing of the mains supply. The recommended setting is:

Total no of supplies	Master setting	Slave 1 setting	Slave 2 setting
1	PHASE 0 SLAVE 1		
2	PHASE 0 SLAVE 1	PHASE 1 SLAVE 1	
3	PHASE 0 SLAVE 2	PHASE 1 SLAVE 2	PHASE 2 SLAVE 2

UNIT OK This gren LED indicates that the power module is working and contributes sufficient to the total load.

RESET Pressing this countersinked pushbutton will reset both the power supply modules and all processing modules connected to the back plane bus.

3. CONSOLE OPERATION

The console may be in one of two possible modes: Debug-mode (D-mode) or Terminal-mode (T-mode). A switch between the two modes takes place when the BELL key (<CTRL>G) is activated.

3.1 Terminal Mode

The console may work as terminal for the RC5000 software system, while in T-mode.
An operator process coordinates the communication between the software system and the operator.

3.1.1 Operator Process

Input and output messages to the operator are identified by a name. The operator process holds a variable 'current name', which identifies the current process for input or output.

'Current name' is updated in the following situations:

1. <ESC> <name> <NL>

   The input line contains a name which is assigned to 'current name'. The operator searches its queue of pending input messages for a message with <name> as identification. If at least one message is found, it is activated. If it is a reactivation, the old input is repeated.

   If no message is found, BELL is echoed.

   Note: <ESC> <NL> is attention to 'current name'.

2. By output.

   If the output message has name = 'current name', the output message is printed on the console.

   If the output message has name <> 'current name', 'current name' is updated and

   > "current name"

   is printed followed by the text from the output message.

3. <ESC> ? <NL>

   Prints the identifications of all pending input messages. 'current name' is updated to the last name in the printout.

The operator has a number of facilities for controlling output and for editing purposes:

<CTRL> E	- deletes the whole line
<BS>	- deletes the last character
<ESC> <NL>	- repeats the whole line
<RUBOUT>	- deletes the last character ('_' is echoed).
<CTRL> S	- stops output. May be reset by <ESC>.
<CTRL> Q	- starts output.
<CTRL> O	- skips output. The function is alternating between skip and noskip, and is reset to 'no skip' by <ESC>.

Note: <NL> may be Carriage Return or Line Feed.

3.1.2 OPSYS Commands

OPSYS interprets the following commands. The underscored characters are sufficient for the interpretation.

More than one command may be typed on one line, unless the syntax is terminated by <nl>.

Whenever <process> or <program> is listed - unless otherwise stated - we refer to incarnations of two program families declared in ADAM with one of the following headings:

```
PROGRAM pip (VAR sv: system_vector);
PROGRAM pop;
```

In the following all numbers are decimal, unless otherwise stated.

BREAK <process>

The child <process> is broken with the current value of excode as exception code.

The child may be any process in the process tree.

E.g.: BREAK S

CHECK <module>

performs a CRC16 check of the memory module <module>. <module> is the hexadecimal base address. If an error is detected, the word address interval, the expected, and computed checksum are printed.

E.g.: CHECK F2

CREATE <process> (AS <program>)$^{1}_{0}$

creates an incarnation of the program <program>. If AS <program> is omitted the program is supposed to have the same name as the process. The size of the stack is the current value of SIZE (see the SIZE command).

E.g.: CREATE T1 AS TEST CREATE T2 AS TEST

DATE (<year>.<month>.<day> <hour>.<minute>.<seconds>) $^{1}_{0}$

If the parameters are included the date is initialized. The command always responds with the current date.

E.g.: DATE 88.06.14 15.30.20

EXCODE <integer>

initializes the current value of excode to <integer>.
Default: excode = -1.

E.g.: EXCODE 47

FREE (<module>)$^{1}_{0}$ <nl>

lists the free interruption levels and memory bytes. If <module> is specified, the free memory areas in the RAM module <module> are listed. The start displacement and size in bytes of the holes are listed, besides the number of holes, the minimum hole, the maximum hole and the sum of the holes.

FROM ($\frac{\text{FPA}}{\text{LAN}}$)

initializes the current value for load kind to FPA or LAN. Load kind controls the kind of driver used for dynamic load (see LOAD). Default load kind is taken from the hexadecimal switches.

E.g.: FROM LAN

GETSWITCHES

lists the current value of the four switches marked BUS on the Processor Front Panel.

HELP

lists the available OPSYS commands.

IN <inchannel>

initializes the current value of the I/O channel used for load. Default <inchannel> is taken from the hexadecimal switches.

E.g.: IN 16

KILL <process>

works as the REMOVE command.

LINK <program>

links a program with name <program> to a program declaration in ADAM.

E.g.: LINK CPUSE

LIST (<process>) $^{n}_{0}$ <nl>

lists the process tree with root <process>. If <process> is missing 'adam' is taken as root. The wildcard character '*' may be used in <process> to specify zero or more occurrences of "I don't care" characters. If wildcard characters occur, only processes fulfilling the wild compare operation are listed. The process state 'fault' means that the process has entered the 'exit' state via an exception.

E.g.:

```
list
process  depth  prio  state  stack    regset  father
adam     0      -1    wait   c0.d648  084     operator
opsys    1      -1    run    c0.d9d0  085     adam
fs       1            exit   c0.e0c0  086     adam
bfs      2      -2    wait   c0.fa9c  08b     fs
lfs      2      -2    wait   c6.aaf0  08a     fs
fsevent  2      -2    wait   c0.f024  089     fs
dsm1     2      -2    wait   c0.ed04  088     fs
sail     2      -2    wait   c0.e9c8  087     fs
```

LISTMAILBOX <process> (<mailbox>)

lists the mailbox catalog of the process, besides the state of the named mailboxes. If no key mailbox name is specified all cataloged names are listed. If the mailbox key contains wildcard characters, only the mailbox names fulfilling a wild compare operation are listed.

LOAD (<program>)$_0^n$ <nl>
or
LOAD <filename> <nl> if LAN

loads the programs from an external device of kind 'current load kind' (see the FROM command) in the I/O channel 'current in-channel' (see the IN command). If no programs are specified, all appearing programs are loaded.

```
E.g.:     FROM LAN IN 16 LOAD MYFILE
E.g.:     FROM FPA IN 81 LOAD PRINTCHAR PRINTNL
```

A binary relocatable file is loaded from RC8000 via FPA by e.g.:

MAIN35001 = CRC16 (Bxxxx)$_1^n$

LOG <module>

prints the contents of the MEM205 or MEM206 errorlog, if not empty.

```
E.g.:     LOG C8
```

LOOKUP (<program>)$_0^n$ <nl>

makes a lookup in the LINKER catalog for the listed programs.

If <program> is empty, the whole LINKER catalog is listed. 'LOOKUP PROGRAM' lists all the programes, 'LOOKUP FUNCTION' lists all the functions, 'LOOKUP PROCEDURE' all the procedures, and 'LOOKUP DATA' all the data programes. Note: The wildcard character '*' may be used in <program> to specify zero or more occurrences of "I don't care" characters.

E.g.:

```
lookup program
>linker
PROGRAM linker      1990.03.25:13.53   1990.03.25:14.02    4312   3.4
PROGRAM monitor     1990.11.05:08.34   1990.11.05:08.37    6438   3.4
PROGRAM timer       1990.03.25:13.53   1990.03.25:14.02    3538   3.3
PROGRAM allocator   1990.03.25:13.53   1990.03.25:14.02    4556   3.6
PROGRAM printexcept 1990.10.30:09.57   1990.10.30:10.01    5676   3.10
PROGRAM adam        1990.03.25:13.53   1990.03.25:14.02    9060   4.1
PROGRAM opsys       1990.11.05:08.27   1990.11.05:08.31   11178   4.1
PROGRAM loader      1990.10.30:08.57   1990.10.30:09.00    7878   3.10
PROGRAM loaddriver  1990.10.30:08.57   1990.10.30:09.00     558   3.10
PROGRAM fp          1990.10.15:22.55   1990.10.15:22.59   12064   1.18
PROGRAM getfpa      1987.06.09:14.12   1987.06.09:14.14    2314   1.7
PROGRAM fts         1990.07.17:11.59   1990.09.10:10.30    5876   3.0
PROGRAM lld         1990.10.24:08.07   1990.10.24:08.17    4884   2.11
PROGRAM mfc2        1990.06.11:13.06   1990.06.11:13.13    6462   2.11
PROGRAM fsformat    1990.09.27:11.35   1990.10.09:09.54    6966   2.13
PROGRAM fstest      1987.05.11:10.40   1990.10.09:09.59    2974   0.0
PROGRAM dsm         1990.09.27:16.25   1990.10.09:08.43    3850   2.8
PROGRAM fosm        1990.10.09:07.40   1990.10.09:08.45    2986   2.3
PROGRAM bfs         1990.09.19:14.45   1990.10.09:08.50    5512   2.5
PROGRAM lfs         1990.10.01:17.49   1990.10.09:08.53    5300   2.7
PROGRAM fs          1987.03.25:15.31   1990.10.12:16.38    3168   0.0
PROGRAM saidriver   1990.11.02:16.25   1990.11.02:16.27    4916   1.7
PROGRAM fsevent     1987.10.13:13.30   1990.10.09:09.48    5164   0.0
PROGRAM resetsai    1990.08.28:09.33   1990.10.12:17.24    1642   1.1
PROGRAM lf3502      1990.10.17:13.46   1990.10.17:13.52   10946   2.23
PROGRAM initboot    1990.09.13:13.10   1990.09.27:15.55     248   6.2
PROGRAM boot        1990.09.20:12.07   1990.09.27:16.25   36252   11.2
```

The output has the interpretation:

1. program kind
2. program name
3. date of source
4. date of compilation
5. size of object code in bytes
6. version

PRINT (<base>) <first disp> (<last disp>

(<no of words per line> (<delta> $)_0^1$ $)_0^1$ $)_0^1$ $)_0^1$ $)_0^1$ <nl>

outputs the specified memory area with a fixed format (see the example): <base>, <first disp>, and <last disp> are hexadecimal.

<delta> defines the default increase of <first disp> if <last disp> is not specified. After the PRINT command <first disp> := <last disp> + 2.

E.g.:

```
prin c0 0 10 2
 address
c0.0000 0204     516    2   4           0314     788    3  20
c0.0004 0000       0    0   0           000d      13    0  13
c0.0008 7f53   32595  127  83     S     4010   16400   64  16    @
c0.000c ff00    -256  255   0           8000  -32768  128   0
c0.0010 0000       0    0   0
```

PRIORITY <integer>

initializes the current value for priority used by the START or RUN command.

Default: priority = -2.

E.g.: PRIORITY -1

REMOVE <process>

removes the child <process> of ADAM and the associated subtree.

E.g.: REMOVE S

RENAME <old program> <new program>

changes the name of <old program> to <new program>. Only loaded programs (as opposed to autoloaded), which are not accessed by other programs in the system, may be renamed.

E.g.: RENAME MYPROG XMYPROG

RESOURCE (<process>) $^{n}_{1}$ <nl>

lists the memory resources occupied by the process tree with root <process>. The process tree is searched for like the LIST command.
The allocated and used memory for process stack and process heap are listed, besides the messages allocated by the process in terms of no. of messages with bufsize = 0 (Msg0), no. of messages with bufsize > 0 (MsgN) and a sum of all bufsizes (bufsizeN).

RESTART

removes ADAM and the whole application tree, whereafter a link, create and start of ADAM is executed.

RUN <process> AS <program>

links, creates, and starts an process with name <process> of the program <program>.

If <program> is omitted, the program is supposed to have the same name as the process.

E.g.: RUN T1 AS TEST RUN MIRROR

SERVER <servername>

initializes the current value for the FTS server name, used when loading from LAN.

SETSWITCHES <hex>

updates the current internal value of the four switches marked BUS on the Processor Front Panel. The BOOT program may be controlled without changing the physical switches.

SIZE <integer>

initializes the current value of SIZE in bytes used when creating ADAM children.

Default: SIZE = 0.

(Note: SIZE = 0 will trigger the use of the default create size for the program).

E.g.: SIZE 1526

START <process>

starts the ADAM child <process> with the current value of PRIORITY as priority.

E.g.: START S

STOP <process>

stops the ADAM child <process>.

E.g.: STOP S

TRACE <process>

the child <process> is stopped while the current stack of routine activation records of the process is listed.

UNLINK <program>

unlinks a program with name <program> from a program declaration in ADAM.

E.g.: UNLINK CPUSE

UNLOAD <program> <nl>

unlinks, if necessary, and deletes the programs from the LINKER catalog, if the programs are not referenced by other programs. If other programs become not referenced after the delete, these programs are also deleted.

E.g.: UNLOAD CPUSE TEST

USER <username>

updates the current value of the FTS username, used when loading from the current FTS server.

3.1.3 Messages from OPSYS

The rest of the command line is skipped if any of the following messages appear:

*** loader not ready

- the LOADER or the LAN Load Driver, LLD, is not included in the system or is unable to run owing to lack of memory.

*** command not implemented

- the command is not available in this version of OPSYS

*** syntax error

- misspelling of a command

*** programname missing

*** unknown process

*** unknown program

*** programname busy

- incarnations of this program still exist

*** processname missing

*** name in use

*** no free programdeclarations

- you must release a program declaration in ADAM by the command UNLINK

*** program not loaded

- the LINKER catalog does not contain the stated program

*** program parameters not equal

- a program with the stated name exists in the LINKER catalog, but the parameter list does not fulfil the declaration

```
        PROCESS pip;
```

*** size too small or too large

- use the SIZE command to adjust the SIZE parameter

*** program not linked

- use the LINK command

*** unknown program

- the program is not in the LINKER catalog. The program may be loaded by the LOAD command

*** program busy

- the program is still accessed by other programs in the system

3.1.4 Messages from LOADER

*** install more ram memory

- the LOADER cannot get enough memory to run

*** loaddriver no stack

- the driver cannot be created due to lack of memory

*** level reservation trouble

- the interrupt level requested for load is occupied by another program incarnation in the system

scan no: x from fpa in xxxx

- initialize load from RC8000 when the first scan is announced (Note: xxxx is decimal). Later scans are performed by the loader itself with no need for operator assistance.

expected: xxxx
received: xxxx

- the crc16 data check reports an error. The programs should be reloaded

end loader

- normal finis message from the LOADER. A list of the loaded programs is printed with name and compilation date. The list may be extended with the information

*** warning: versionerror

- the program should be recompiled, but loading continues

*** loadfile unintelligible

- the loadfile has the wrong format or is garbage.

*** allready loaded

- the program is already in the LINKER catalog. The program in the catalog is used instead and the loaded program deleted

*** not defined

- the program was not in the LINKER catalog or amongst the loaded programs.

3.2 Debug Mode

3.2.1 Activation

When the MODE switch is set in the range 0 to 7, the console may be set into Debug-mode (D-mode) at any time by pressing the BELL key (CTRL and G keys). This will not change the processor execution state.
A number of display and control commands are available for technical purposes.

NOTE: In D-mode the commands must be typed with capital letters (e.g. with alpha lock activated).

An illegal command will be displayed as an asterix.

In below description the commands marked *RC5050* are available for the RC5050 processor only.

3.2.2 Display Commands

Display commands cause the display of 8 words or 16 bytes of data. RC5050 will display the data as ascii characters as well.

The following display commands are available:

M <addr> Modify Memory
Displays the contents of the 8 memory words starting at <addr>.

W <regset> Modify Working Registers

Displays the contents of the 8 working registers comprising registerset <regset>.

P <regset> Modify Register Stack

Displays the contents of the register stack associated registerset <regset>. At most 8 register stack elements are displayed.

L <level> Modify Working Registers

Displays the level number, the registerset, and the contents of the 8 working registers comprising the registerset connected to <level>.

Y <yaddr> Modify Control Microprocessor RAM

Displays the contents of the 8 control microprocessor RAM bytes starting at <yaddr>.

I C <no> Inspect Control Registers *RC5050*

Displays the contents of the 8 control registers starting with <no>. Control register x is equal to Work Memory address xFF.

I F <f><addr> Inspect Flash Memory *RC5050*

Display the contents of the 16 bytes of flash memory <f> (EEPROM) starting at address <addr>. Flash Memory 0 contains the Service Processor program and Flash memory 1 contains the microprogram.

I N <addr> Inspect Non-Volatile Memory *RC5050*

Display the contents of the 16 bytes of NVM memory starting at address <addr>. The format of the NVM is described in Appendix N. See also the command I ? .

I R <addr> Inspect Register *RC5050*

Display the contents of the 8 registers starting at address <addr>, which is an absolute register address. E.g. the command W 1 is equal to I R 8 .

I S <addr> Inspect Service Processor RAM *RC5050*

Display the contents of the 16 bytes of Service Processor Memory starting at address <addr>.

I W <wno> Inspect Work Memory *RC5050*

Display the contents of the 8 work registers starting at <wno>. <wno> is entered as three hexadecimal digits in the format baa, where b is the bank number in the range 0..F, and aa is the work register number in the range 00..7F. See appendix M for a description of work register utilization.

I Y Inspect Y-RAM Formatted *RC5050*

Display the memory area normally accessed by the Y command formatted and commented. Data cannot be updated with this command.

I ? Inspect Non-Volatile Memory Formatted *RC5050*

Display the Non-Volatile Memory (NVM) normally accessed by the I N command formatted and commented. Data cannot be updated with this command.

? Display all Service Processor commands *RC5050*

Display a list of all accepted Service Processor commands.

Display commands are executed, when a display command is entered.

It is now possible to modify the displayed data by entering new data in the same positions on the following line. Pressing the space bar will move the cursor one position to the right. All updates are byte per byte.

When a P command is terminated (by CR, +, or -) the cursor position defines the number of register stack elements. If the number has been changed, a H is displayed. Note, that a cursor on the first position does not empty the register stack. This is done by the # key (see later).

A display command is terminated by pressing one of the following keys:

The # key (ascii 23 hex) terminates the P command with an empty register stack. Otherwise # is blind.

CR The CR key terminates the current display command. The console will await the next command.

+ The + key terminates the current display command and executes a display command for the succeeding 8 words (M), 8 bytes (Y), up to 8 elements (P), or the 8 registers on the succeeding level (W,L).

- The - key terminates the current display command and executes a display command for the preceding 8 words (M), 8 bytes (Y), up to 8 elements (P), or the 8 registers on the preceding level (W,L).

. The . key terminates the current display command and executes a display command for the same 8 words (M), 8 bytes (Y), up to 8 elements (P), or the 8 registers on the same level (W,L).

ESC The ESC key terminates the current display command, but no data modification takes place in the M, W, P, and L commands. The text <ESC> is displayed. The console will await the next command.

3.2.3 Control Commands

The following control commands are available:

R Run

The processor will start instruction execution.

S Instruction Step

The processor will execute one instruction, stop, and display the current levelno, the registerset, and the contents of the 8 working registers, and reactivate the console. Modification of the displayed data is not possible. The watchdog timer will stop, if started.

S <steps> Multi-Instruction Step

The processor will execute <steps> instructions, stop and reactivate the console.

Z Instruction Step

The command works as the S command. The P display command is performed implicitly, so the contents of the current register stack is also displayed.

Z <steps> Multi-Instruction Step

The command works as the S <steps> command, but a P display command is performed implicitly for each instruction step.

T <testno> Single Selftest

The processor will execute a single selftest, in a loop mode, according to the following table. If testno is chosen as C1-D5 for RC5040 and 81-8F for RC5050, then there will be no errormessage, and the test will continue even if an error occurs. The memory test is performed on 64K bytes blocks according to the RAM configuration bit map. See app K.
The test can be terminated by use of the ESC key.
Errorno + info are explained in 2.2.2.

In below test descriptions the function marked "stop at error" is to stop and output an error message at first occurance of an error while the function marked "ignore error" is to detect errors, but not output any error message and not terminate the repitive execution of the test.

RC5040:

testno stop at error	testno ignore error	err no	test
00C1	0081	00	fifo test
00C3	0083	-	7.5 interrupt test
00C5	0085	02	W-register address test
00C7	0087	03	W-register data test
00C9	0089	04	memory address test
00CB	008B	05	memory data test
00CD	008D	06	internal intr. test
00CF	008F	07	schedule test
00D1	0091	08	intmap test
00D3	0093	09	prefetch test
00D5	0095	0A	register stack test
00D7	0097	0B	8085 EPROM sum test
00DF	009F	-	'power' restart, but no reset of controllers

RC5050:

testno stop at error	testno ignore errors	test
1	81	SP / MP Interrupt Test
2	82	Register Address Test
3	83	Register Data Test
4	84	Working Register Address Test
5	85	Working Register Data Test
6	86	Gate Array Register Test
7	87	Scheduler RAM Test
8	88	Scheduler Nxs Test
9	89	Interrupt Chain Test
MM0A	MM8A	Memory Address Test
MM0B	MM8B	Memory Data Test
C	8C	HW Stack Test
D	8D	MicroSec Counter Test
E	8E	FIFO Test
MM0F	MM8F	PreFetch Test

Tests marked MM will use the high order byte of the <testno> command parameter as module address, e.g: T C28F will start the Prefetch Test in module C2 and ignore errors encountered.
A 16 bits pass counter can be used in single test execution. It will be set to zero, when the T command is used, and is incremented by one before each pass. The value may be examined, when the T command has terminated after ESC or after an error. For RC5040 the command Y65 gives the LSB, and Y66 gives the MSB of the pass counter. For RC5050 the pass counter value is displayed at test termination.

T ? Display list of selftests *RC5050*

Display a list of available selftests.

C G Control: Gate Array Reset *RC5050*

This command will issue a Gate Array reset.

C I Control: Restart SP Initiator *RC5050*

This command will cause the processor to stop and enter the Service Processor Initiator to restart, i.e. micro program load, test programs and autoload will not be initiated.

C R Control: Restart Processor *RC5050*

This command will cause the processor system to issue a bus reset and restart the system exactly like a power on restart.

C W Control: Watch Dog Restart Processor *RC5050*

This command will cause the processor system to restart exactly like a Watch Dog restart.

3.2.4 Command Parameters

All numbers entered or displayed are hexadecimal.

At any time the entering of an empty command (i.e. pressing the CR key) will cause the previous command to be repeated.

An address (<addr>) is entered using one of the following formats:

```
          <base> : <disp>
or
                 : <disp>
```

<base> is the leftmost 8 bits of the 24-bit address.

<disp> is the displacement within the selected memory module, i.e. the rightmost 16 bits of the address.

If the second format (: <disp>) is used, the last entered address base will be echoed and used.

4. AUTOLOADING

The autoload function may be initiated by:

- Power Restart
- Watchdog Restart

Power Restart

Power Restart happens:

- when power ON is performed manually on the OCP or on the power supply,

- by temporary power failure,

- by manual activation of the autoload button on the OCP or the AUTO push button on the power supply.

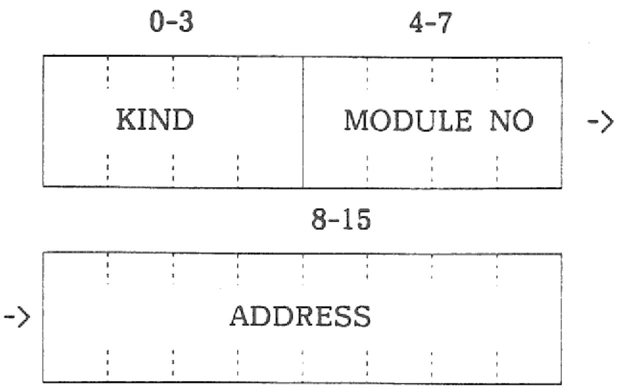
The built-in test programs are activated controlled by the 'MODE' switch, the CPU initializes the registers, whereafter control is passed to the autoload program residing on the first memory module.

Watchdog Restart

Watchdog Restart may be activated both manually by means of the 'Y' or C W debug console command and from the software. See app. K.
The CPU initializes the registers, whereafter control is given to the autoload program. No built-in test programs are activated.

4.1 Autoload Switch Format

The autoload program interprets the four BUS switches on the Processor Front Panel according to the format:



The **autoload KIND** field defines which algorithm the autoload program executes, whereas the **MODULE number** and **ADDRESS** fields have different interpretations depending on the value of the KIND field. The description in this section is ordered according to the value of the KIND field. That is, a subsection for each value of the KIND field. The kind values 8, 9, 12, 14 and 15 (decimal) are currently not used.

4.1.1 KIND = 0: LAN + X.25/3

Autoload is from RcLAN. If the autoload fails, an X.25/3 autoload via COM204 is initiated. After autoload the software in EPROMs is included.

When loading from LAN, **MODULE number** controls the USER name, when the RC5000 tries to LOGON to an FTSSERVER.
FTSUSER name

MODULE	FTSUSER NAME
0	cp<ID register>
1-E	cp1..cp14
F	cp

The ID register postfix is four hexadecimal digits. The FTSSERVER's are requested for autoload of the file 'boot9330' N times in a cyclic way. When this limit is reached, the basic FTSUSER name is changed from cp to pp and the cyclic file requesting is continued N times. If no success, a permanent fall back to X.25/3 autoload is performed.

When a X.25/3 autoload is performed with kind = 0, **MODULE number** is pretended to be 0, causing the X.25/3 autoload to try the possible X.121 adresses in a cyclic way.

For a detailed description of an X.25/3 autoload, please refer to section 4.1.8 below.

Conventions:

1. COM204 interruption level must be 72 (48 hex).
2. MFC20X interruption level must be 16 (10 hex).

Example

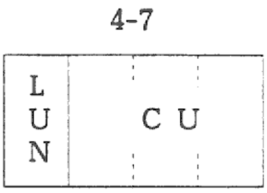
Autoload from RcLAN via MFC20X with FTSUSER = cp7 on interruption level 16 (decimal). If the autoload fails after requsting N FTS servers, change FTSUSER to pp7 and continue with N requests. If the autoload still fails, continue with X.25/3 autoload via COM204, as if the switch settings were 70xx.

70xx

4.1.2 KIND = 1: WD

Autoload is from Winchester Disc, and the software in EPROMs is included.

The **MODULE number** field has the interpretation:



CU - defines the Control Unit number of the Disc Controller on the SCSI bus.

LUN- defines the Logical Unit Number of the Winchester Disc at the Disc Controller.

The **ADDRESS** field specifies the SAI201 DATA interruption level.

Example

Autoload from Winchester Disc with LUN = 1 via disc controller with CU = 2, SAI201 data interruption level 83 (decimal) and SAI201 control interruption level 82 (decimal):

1A53

4.1.3 KIND = 2: FPA (Autoload Enabled)

Autoload is from FPA, and the software in EPROMs is included. Drivers controlling external devices may autoload the RC5000 by activating the watchdog function (autoload is enabled).

The **MODULE number** has no significance.

The **ADDRESS** field specifies the FPA100 REC interruption level.

Example

Autoload from FPA in channel 81 (decimal)

2x51

4.1.4 KIND = 3: WD + COM204

Autoload is from Winchester Disc. If the autoload fails, a COM204 autoload is initiated. After autoload the software in EPROMs is included.

The **MODULE number** and **ADRESS** fields are only used to specify the COM204 autoload.
Therefore be aware of the following conventions concerning the WD autoload:

1. The Control Unit, CU, of the disc controller on the SCSI bus must be 0.

2. The Logical Unit Number, LUN, of the winchester disc at the disc controller must be 0.

3. The SAI201 data interruption level must be 87 (decimal).

4. The SAI201 control interruption level must be 86 (decimal)

If the WD autoload fails, a COM204 autoload is initiated. Then the interpretation of the **MODULE number** field is:

4-7

F	F A N

F - defines the type of load request transmitted on the HDLC line:
0 PAXNET loadrequest
1 X.25/3 loadrequest

FAN- defines the maximum relative channel number used for autoload in a cyclic way if a channel fails. Channels 0,1,...FAN relative to the start channel are used.

The **ADDRESS** field specifies the CHANNEL*128 + COM204 interruption level. That is, the leftmost bit of the ADRESS field specifies the channel (A/B corresponding to 0/1) and the rightmost 7 bits specify the COM204 interruption level.

Conventions:

1. Start address of the reference controller must be 90.0000.

2. Interruption level of the reference controller must be 72.

3. The controllers must be consecutive and increasing memorywise and according to interruption levels (see the following table).

	address	level	channel	
Reference Controller	90.0000	72	A B	
	90.8000	73	A B	<- 3)
	92.0000	74	A B	
First Controller	92.8000	75	A B	<- 3) <- 1)
Last Controller	94.0000	76	A B	 <- 2)
	94.8000	77	A B	<- 3)
	96.0000	78	A B	
	96.8000	79	A B	<- 3)

1) first channel in fan
2) last channel in fan
1) For COM204 strapped to 32 kbyte Dual Port memory only.

Example

Autoload from Winchester Disc with LUN = 0 via disc controller with CU = 0, SAI201 data interruption level 87 (decimal) and SAI201 control interruption level 86 (decimal). If the autoload fails, autoload is from COM204 address 92.8000 level 75 (decimal) channel B using PAXNET loadrequest and a fan consisting of 3 channels:

32CB

4.1.5 KIND = 4: LAN

Autoload is from RcLAN, and the software in EPROMs is included.

MODULE number controls the USER name, when the RC5000 tries to LOGON to an FTSSERVER.

MODULE	FTSUSER NAME
0	cp<ID register>
1-E	cp1..cp14
F	cp

The ID register postfix is four hexadecimal digits.

The FTSSERVER's are requested for autoload of the file 'boot9330' indefinitly in a cyclic way.

The ADDRESS field specifies the MFC20X interruption level.

Example

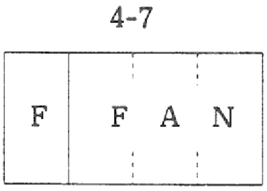
Autoload from RcLAN via MFC20X with FTSUSER = cp on interruption level 16 (decimal).

4F10

4.1.6 KIND = 5: COM204

Autoload is from COM204 (Intelligent HDLC Controller), and the software in EPROMs is included.

The MODULE number field has the interpretation:



F - defines the type of load request transmitted on the HDLC line:
    0 PAXNET loadrequest
    1 X.25/3 loadrequest

FAN- defines the maximum relative channel no. used for autoload in a cyclic way if a channel fails. Channels 0,1,...FAN relative to the start channel are used.

The ADDRESS field specifies the CHANNEL*128 + COM204 interruption level. That is, the leftmost bit of the ADRESS field specifies the channel (A/B corresponding to 0/1) and the rightmost 7 bits specify the COM204 interruption level.

Conventions:

1. Start address of the reference controller must be 90.0000.
2. Interruption level of the reference controller must be 72.
3. The controllers must be consecutive and increasing memorywise and according to interruption levels (see the following table).

	address	level	channel	
Reference Controller	90.0000	72	A B	
	90.8000	73	A B	<- 3)
	92.0000	74	A B	
First Controller	92.8000	75	A B	<- 3) <- 1)
Last Controller	94.0000	76	A B	 <- 2)
	94.8000	77	A B	<- 3)
	96.0000	78	A B	
	96.8000	79	A B	<- 3)

1) first channel in fan
2) last channel in fan
1) For COM204 strapped to 32 kbyte Dual Port memory only.

Example

Autoload from COM204 address 92.8000 level 75 (decimal) channel B using PAXNET loadrequest and a fan consisting of 3 channels:

52CB

4.1.7 KIND = 6: EPROM

No autoload from external device. Only software in EPROMs is included.

In this case neither the MODULE number nor the ADRESS fields are used.

Example

No autoload, only inclusion of software in EPROMs.

6xxx

4.1.8 KIND = 7: X.25/3

Autoload is X.25/3 via COM204, and the software in EPROMs is included.

MODULE number controls the X.25/3 calling algorithm. If the switch is zero, the possible X.121 addresses are tried in a cyclic way. If the switch differs from zero, the X.121 address with index in the internal X.121 address table equal to the switch value is used permanently for repeated calls.

The ADDRESS field has the following interpretation:

8	9 - 15
A / B	LEVEL

where

A/B 0 indicates COM204 channel A and 1 channel B.

LEVEL defines the interruption level of the load controller.

Convensions for controller start address and interrupt level are identical to KIND 3 and KIND 5.

Example

X.25/3 autoload via COM204 at interrupt level 72 (48 hex) channel A using the X.121 address with internal index 2.

7248

4.1.9 KIND = 8: Not used

4.1.10 KIND = 9: Not used

4.1.11 KIND = 10 (A hex): FPA (Autoload Disabled)

Autoload is from FPA, and the software in EPROMs is included. Drivers controlling external devices may **not** autoload the RC5000 by activating the watchdog function. Autoload is disabled.

The **MODULE number** has no significance.

The **ADDRESS** field specifies the FPA100 REC interruption level.

Example

Autoload from FPA in channel 81 (decimal)

Ax51

4.1.12 KIND = 11 (B hex): WD + MEGA

Autoload is from Winchester Disc. If the autoload fails, a MEGA autoload is initiated. After autoload the software in EPROMs is included.

The **MODULE number** and ADRESS fields are only used to specify the MEGA autoload.
Therefore be aware of the following conventions concerning the WD autoload:

1. The Control Unit, CU, of the disc controller on the SCSI bus must be 0.

2. The Logical Unit Number, LUN, of the winchester disc at the disc controller must be 0.

3. The SAI201 data interruption level must be 87 (decimal).

4. The SAI201 control interruption level must be 86 (decimal)

If the WD autoload fails, a MEGA autoload is initiated. Then the interpretation of the MODULE number field is:

4-7

F	F A N

F - not used

FAN- defines the maximum number of connectors used for autoload. If a connector fails the next connector is tried in a cyclic way. Connectors 0,1,...FAN relative to the first connector are used.

The ADDRESS field specifies the first connector to be tried and ADDRESS must be grester than 1.

Example:

B002

A Winchester Disk autoload is attempted. If it fails a Down Line Load from MEGA connactor 2 is attempted. If that fails it is reattempted.

4.1.13 KIND = 12 (C hex): Not used

4.1.14 KIND = 13 (D hex): MEGA

Autoload is via Down Line Load from one or more MEGA line connectors.

The interpretation of the MODULE number field is:

4-7

F	F A N

F - not used

FAN- defines the maximum number of connectors used for autoload. If a connector fails the next connector is tried in a cyclic way. Connectors 0,1,...FAN relative to the first connector are used.

The ADDRESS field specifies the first connector to be tried and ADDRESS must be grester than 1.

Example:

B103

Down Line Load from MEGA connactor 3 is attempted. If it fails a Down Line Load from Mega connector 4 is attempted. If that fails MEGA connector 3 is reattempted etc.

4.1.15 KIND = 14 (E hex): Not used

4.1.16 KIND = 15 (F hex): Not used

4.2 Autoload Message

boot version x.x

```
Autoload from mfc level xxH
      user : <FTS User Name>    file : boot9330
```

- FTS User Name is 'cp' appended the contents of the ID Register if the 'module number' switch is zero. '1' to '14' is appended if the 'module number' switch takes the values 1 to E (hexadecimal). If the 'module number' switch is F nothing is appended.

calling : <X.121 address>

calling : cyclic starting at : <X.121 address>

** end print

- Announcement of autoload is no longer performed. The autoload procedures continue.

Autoload from fpa in xxH

Autoload from com204 addr xx.xxxx level xxH channel xxH

Autoload from eprom

```
Autoload from wd in xxH (D) xxH (C) CU xx lun xx
    file : /autoloadcat/<entryname>
```

- <entryname> is taken from the Control Microprocessor RAM. If the first character in this name is zero, the default name 'boot3502' appended the CU of the RC5000 is used.

*** undefined switchkind xxH

....

- a full stop is printed for every program loaded.

:::

- a colon is printed for every RAM module, which contains programs with a checksum catalog.

;;;

- a semicolon is printed for every PROM module, which contains programs with a checksum catalog.

boot *** exception: xxxx at: xx.xxxx

- consult appendix I for interpretation of the exception code. The autoload is restarted after one minute.

*** level not installed : xx

- the requested interruption level is not installed. The autoload program is restarted.

*** install more memory at : xx

- more RAM memory must be installed to hold the autoloaded programs. The autoload is restarted.

**** warning: versionerror at xx.xxxx

- the program identified by the address must be recompiled to .be autoloaded. Consult the output from CROSSLINK for identification. The autoload is restarted.

**** warning: sumerror module : xx

- the crc16 data check reports an error in the EPROM module xx. Operation continues.

**** warning: <program> not included

- the basic program <program> should be included in the autoloaded programs or in EPROMs. Operation continues.

*** sumerror module : xx

- the crc16 data check reports an error in the RAM module xx. The autoload program will start all over again.

last-block xxxxH
new-block xxxxH

- block sequence error. The autoload program will start all over again.

*** harderror : xx

- a harderror occurred during autoload from Winchester Disc. The autoload program is restarted.

  02 SCSI bus phase error in the selection phase.

  09 No connection to the SAI201.

  0a Timeout of interrupt from SAI201 control channel in the selection phase.

  0b Hard error on SAI201. Reset of SAI201 failed.

  12 Parity error on the SCSI bus in the selection phase.

22 SCSI bus phase error in the command phase.

2a Timeout of interrupt from SAI201 control channel in the command phase.

32 Parity error on the SCSI bus in the command phase.

42 SCSI bus phase error in the data input phase.

4a Timeout of interrupt from SAI201 control channel in the data input phase.

52 Parity error on the SCSI bus in the data input phase.

62 SCSI bus phase error in the data output phase.

6a Timeout of interrupt from SAI201 control channel in the data output phase.

72 Parity error on the SCSI bus in the data output phase.

82 SCSI bus phase error in the status phase.

8a Timeout of interrupt from SAI201 control channel in the status phase.

92 Parity error on the SCSI bus in the status phase.

a2 SCSI bus phase error in the message phase.

aa Timeout of interrupt from SAI201 control channel in the message phase.

b2 Parity error on the SCSI bus in the message phase.

*** filelimit : root xxxx

- The catalog with name 'autoloadcat' is not found in the root catalog. The autoload program is restarted.

*** filelimit : autoloadcat xxxx

- The announced autoloadfile is not found in the catalog 'autoloadcat'. The autoload program is restarted.

*** filelimit : <autoloadfile> xxxx

- The announced autoloadfile is unintelligible. The autoload program is restarted.

4.3 Generating Autoload Files

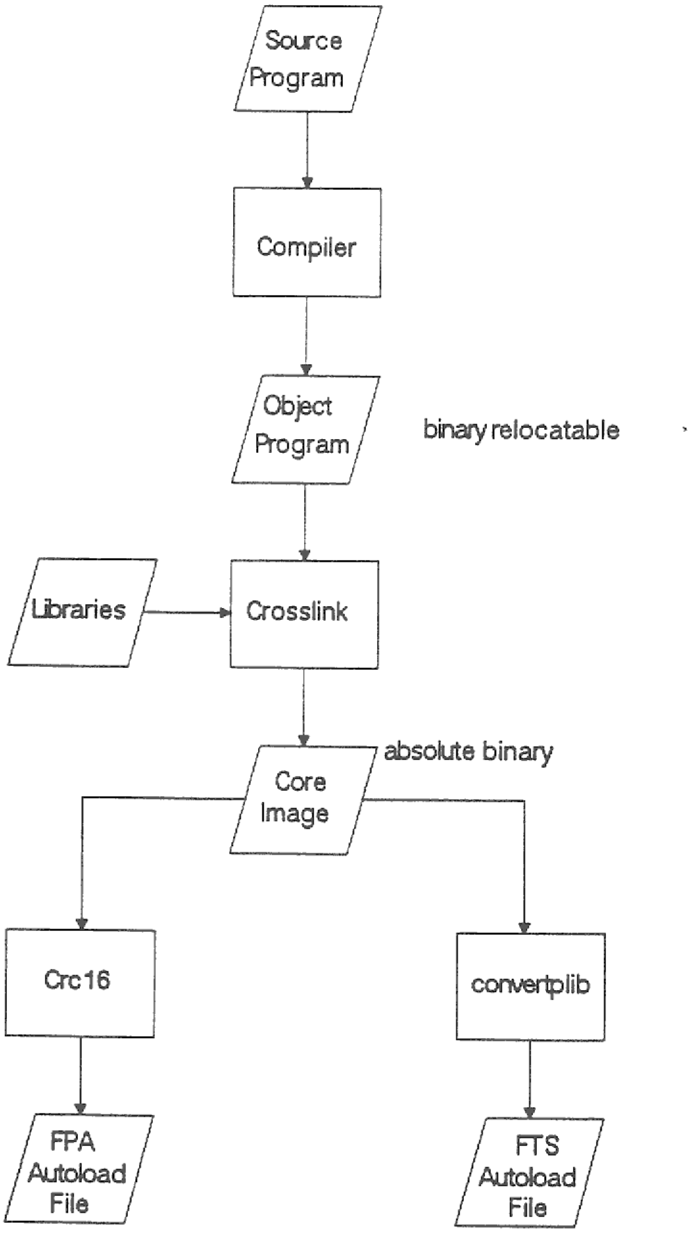


Fig. 4.1. Autoload File Generation

4.3.1 Generating an FTS Autoloadfile

If the file 'coreimage' has been produced on RC8000 by the 'CROSS-LINK' program, the following call will generate a file with a format for autoload via RcLAN by the FTS protocol. The file 'boot9330' must be placed at the FTS Server under the proper catalog bases or directory. The FTS Username is controlled by the 'module number' switch on RC5000.

```
boot9330 = convertplib coreimage
```

4.3.2 Generating an FPA Autoload

If the file 'coreimage' has been produced on RC8000 by the 'CROSS-LINK' program, the following call will autoload the RC5000 if connected via the process 'main35001':

```
main35001 = crc16 coreimage
```

4.3.3 Generating TES202 Eproms

Consult ref. 2.

5. ERROR PROCEDURES

For use in error situations it might be useful to get testoutput from the autoload program 'BOOT' or from the program MONITOR which starts and stops program incarnations.

5.1 Monitor Testoutput

The monitor stack address is obtained by the LIST MONITOR command to OPSYS or from the testoutput from BOOT (see 5.2).

If a bit is set in the byte, the testoutput associated the bit will be generated.

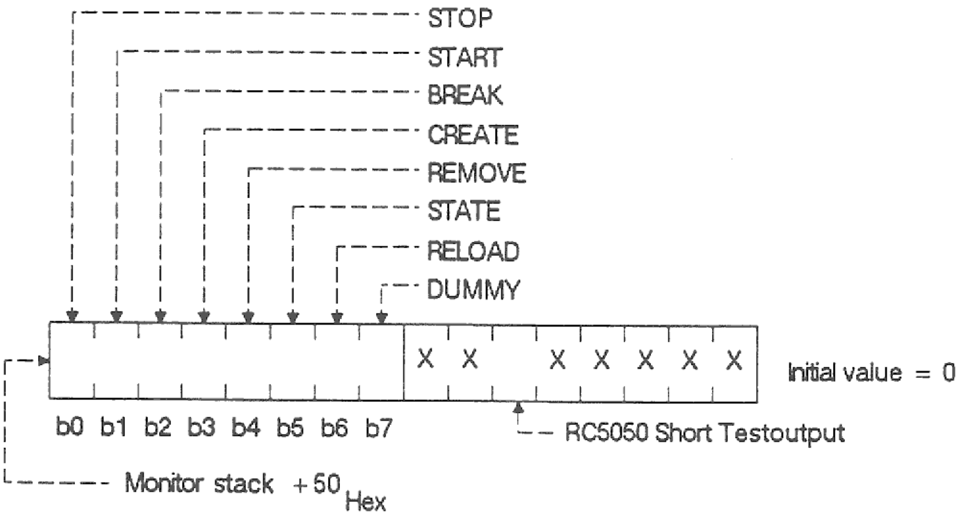


Fig. 5.1. Monitor testoutput

5.2 Boot Testoutput

The BOOT stack address is xx.F850 where xx is the base address of the last RAM memory module in the RC5000. If testoutput during autoload from FPA is wanted, you must increase the software time out in the FPA xmt driver in RC8000 from 1 sec. to at least 5 sec.

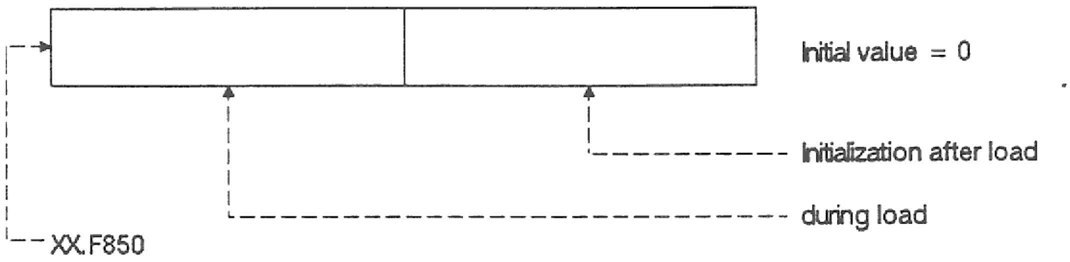


Fig. 5.2. Boot testoutput

A. REFERENCES

1. PN: 99000994
   RC3502-2 Real-Time Pascal Reference Manual

2. PN: 99103221
   RC3502 PROM Blasting Program User's Guide

3. RCSL No. 52-AA1177
   CPU212-219 Technical Manual

4. RCSL No. 52-AA1192
   RC3502/2 Reference Manual

   PN: 99001344
   RC5050 Reference Manual Addendum

5. PN: 99000990
   Debugger Listing ROB984

B. OPSYS COMMANDS

```
break <process>

check <module base>

create <process> (as <program>)

date  <year>.<month>.<day> <hour>.<minute>.<seconds>

excode <integer>

free (<module base>)

from lan
from fpa

help

in <inchannel>

kill <process>

link <program>

list <process> <nl>

listmailbox <process> (<mailbox>)

load <program> <nl>
load <filename> <nl>       if LAN

log <module base>

lookup <program> <nl>

print <base> <firstdisp> <lastdisp>
        <no_of_words_per_line> <delta>
```

priority <integer>

remove <process>

rename <oldname> <newname>

resource (<process>)

restart

run <process> (as <program>)

server <servername>

setswitches <hex>

size <stacksize>

start <process>

stop <process>

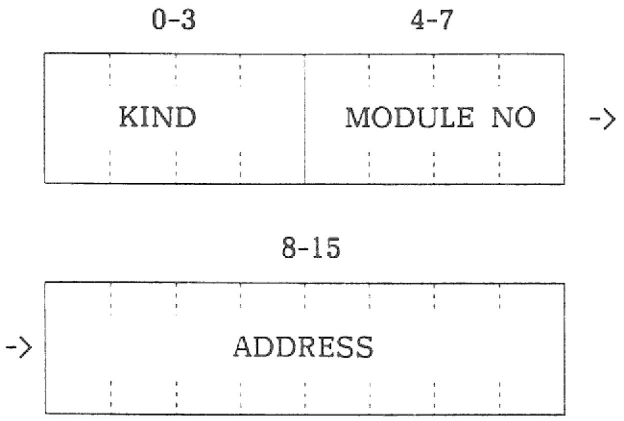
trace <process>

unlink <program>

unload <program> <nl>

user <username>

C. AUTOLOAD SWITCH LAYOUT



KIND

0	LAN + X.25/3	8	Not used
1	WD	9	Not used
2	FPA (Autoload enabled)	A	FPA (Autoload disabled)
3	WD + COM204	B	WD + MEGA
4	LAN	C	Not used
5	COM204	D	MEGA
6	EPROM	E	Not used
7	X.25/3	F	Not Used

MODULE number

LUN x 8 + CU	if WD
Load request*8+fan	if COM204 or WD+COM204
Cyclic or specific call	if X.25/3
FTS User Name	if LAN or LAN+X:.25/3
Fan	if MEGA or WD+MEGA

ADDRESS

SAI201 DATA intrpt level	if WD
Interruption level	if FPA or LAN
Channelx128+intrpt level	if COM204 or WD+COM204
Channelx128+offsetx16+table	if X.25/3 or LAN+X.25/3
First connector	if MEGA or WD+MEGA

D. SECRET VECTOR

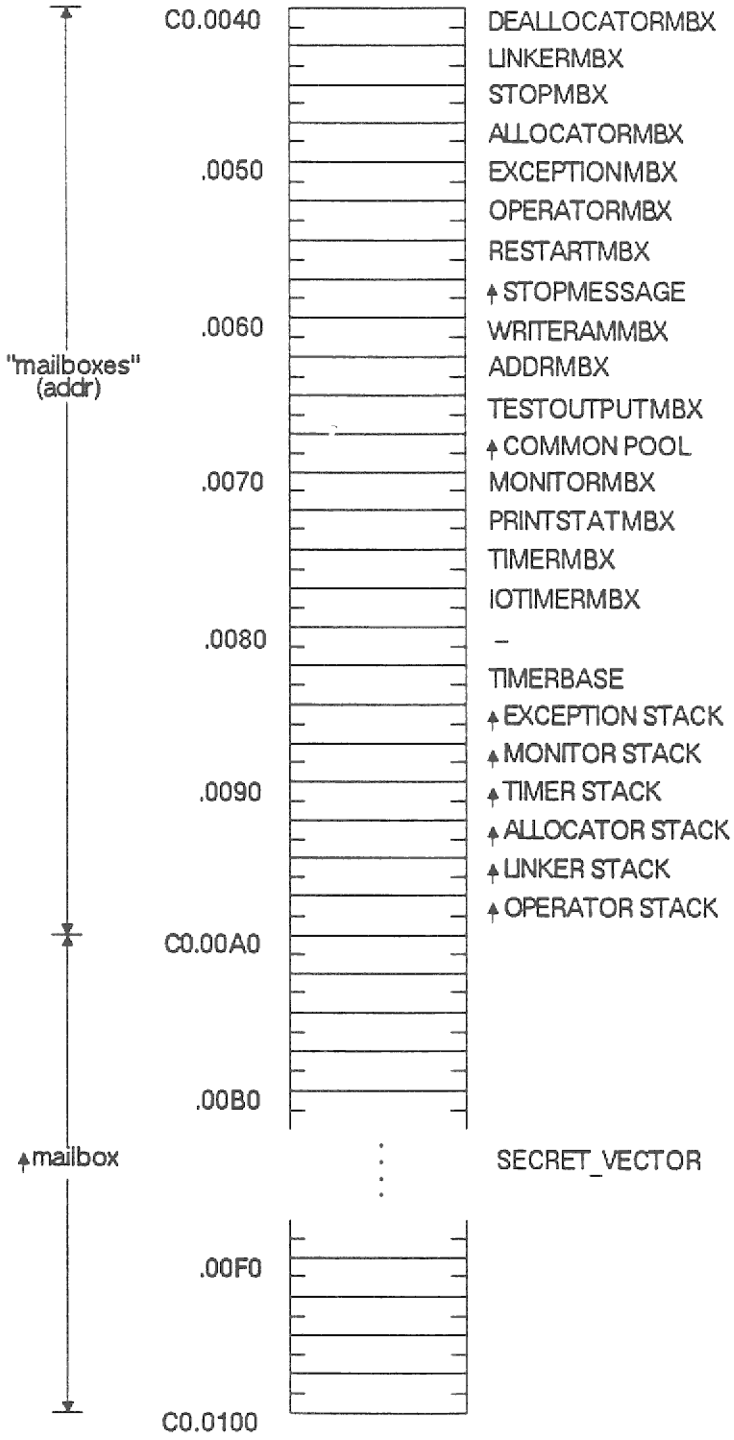


Fig. D.3.)

E. INSTRUCTION CODES

rec1	01		lt	34		jmprw	68		stvsa	9c		mbset	d0	
rec2	02		gt	35		jmphc	69		svsb5	9d		mbtes	d1	
rec3	03		le	36		jmpga	6a		stvsd	9e		stvga	d2	
rec4	04		ge	37		jmcht	6b		svsb7	9f		revga	d3	
rec5	05		indh0	38		intrs	6c		mwit	a0		ststc	d4	
rec6	06		indh1	39		index	6d		mcit	a1		restc	d5	
rec7	07		topen	3a		intpa	6e		svsw2	a2		stvla	d6	
rec8	08		tlock	3b		inpdv	6f		rvsw2	a3		revla	d7	
rec9	09		teqad	3c		iorbbc	70		svsw4	a4		stvsf	d8	
rec10	0a		tnill	3d		iorbb	71		rvsw4	a5		revsf	d9	
rec11	0b		mmul	3e		iowbbc	72		svsw6	a6		rvlbns	da	
rec12	0c		madd	3f		iowbb	73		rvsw6	a7		rvlwns	db	
rec13	0d		cwait	40		iorbwc	74		rechw	a8		rvlans	dc	
rec14	0e		msub	41		iorbw	75		revgws	a9		rvldns	dd	
rec15	0f		uadd	42		iowbwc	76		stvlws	aa		crram	de	
csign	10		usub	43		iowbw	77		revlws	ab		cwram	df	
crele	11		add	44		pcals0	78		moveb	ac		mwist	e0	
movebs	11	1	sub	45		pcals1	79		indh02	ad		mcist	e1	
escape	12	2	umul	46		pcals	7a		moveg	ae		reaid	e2	
cinwq	13	3	udiv	47		pexit	7b		revpw	af		rvsd2	e3	
coutwq	14	3	umod	48		lpush	7c		readw	b0		reald	e4	
csens	15		mul	49		lpop	7d		crput	b1		rvsd4	e5	
lrele	16		div	4a		lrese	7e		stvgw	b2		reasd	e6	
mtime	17		mod	4b		llock	7f		revgw	b3		uadhw	e6	
mhalt	18		and	4c		mwt	80		reagd	b4		rvsd6	e7	
csell	1a		or	4d		mrecha	81	3	reaisd	b5		rechd	e8	
cstop	1b		xor	4e		svsb2	82		stvlw	b6		revgds	e9	
cstart	1c		crc16	4f		rvsb2	83		revlw	b7		stvlds	ea	
iocci	1d		neg	50		svsb4	84		revsb	b8		revlds	eb	
cslev	1e		abs	51		rvsb4	85		rvsb1	b9		setst	ec	
cgreg	1f		compl	52		svsb6	86		revsw	ba		indh12	ed	
mwi	20		shc	53		rvsb6	87		rvsb3	bb		stcea	ee	
iowc	21		shc8	54		rec0	88		revsa	bc		revpd	ef	
iors	22		not	55		revgbs	89		rvsb5	bd		rvsb0	f0	
iorw	23		setcr	56		stvlbs	8a		revsd	be		svsb0	f1	
ioww	24		settm	57		revlbs	8b		rvsb7	bf		rvsw0	f2	
iogo	25	3	setatm	58		stnhb	8c		mwst	c0		svsw0	f3	
iogi	26	3	seteq	59		indh01	8d		inth0	c1		rvsa0	f4	
ionci	27		setsb	5a		renpb	8e		svsa2	c2		svsa0	f5	
mrest	28		setsp	5b		renhb	8f		rvsa2	c3		rvsd0	f6	
mstst	29		setun	5c		readb	90		svsa4	c4		revld	f7	
iocda	2a		setin	5d		crget	91		rvsa4	c5		stvld	f8	
ioibx	2b		setdi	5e		stvgb	92		svsa6	c6		revgd	f9	
mtrh	2c		setad	5f		revgb	93		rvsa6	c7		stvgd	fa	
iorsc	2d		mwis	60		setre	94		rechws	c8		reaad	fb	
mtrs	2e		mcis	61		resta	95		revgas	c9		reard	fc	
mnoop	2f		jmzeq	62		stvlb	96		stvlas	ca		reaxd	fd	
reasd1	30		jmzne	63		revlb	97		revlas	cb		cexch	fe	
uadhw1	30		jmzlt	64		stvsb	98		revsm	cc				
ult	31		jmzgt	65		svsb1	99		indh11	cd				
eq	32		jmzle	66		stvsw	9a		reagds	ce				
ne	33		jmzge	67		svsb3	9b		realds	cf				

1. RC5040 only = moveb.   2. RC5050 only.   3. RC5040 only.

F. INSTRUCTION MNEMONICS

abs	51		iowbw	77		pcals	7a	revld	f7	ststc	d4
add	44		iowbwc	76		pcals0	78	revlds	eb	stvga	d2
and	4c		iowc	21		pcals1	79	revlw	b7	stvgb	92
cexch	fe		ioww	24		pexit	7b	revlws	ab	stvgd	fa
cgreg	1f		jmcht	6b		reaad	fb	revpd	ef	stvgw	b2
cinwq	13	3	jmpga	6a		readb	90	revpw	af	stvla	d6
compl	52		jmphc	69		readw	b0	revsa	bc	stvlas	ca
coutwq	14	3	jmprw	68		reagd	b4	revsb	b8	stvlb	96
crc16	4f		jmzeq	62		reagds	ce	revsd	be	stvlbs	8a
crele	11		jmzge	67		reaid	e2	revsf	d9	stvld	f8
crget	91		jmzgt	65		reaisd	b5	revsm	cc	stvlds	ea
crput	b1		jmzle	66		reald	e4	revsw	ba	stvlw	b6
crram	de		jmzlt	64		realds	cf	rvlans	dc	stvlws	aa
csell	1a		jmzne	63		reard	fc	rvlbns	da	stvsa	9c
csens	15		le	36		reasd	e6	rvldns	dd	stvsb	98
csign	10		llock	7f		reasd1	30	rvlwns	db	stvsd	9e
cslev	1e		lpop	7d		reaxd	fd	rvsa0	f4	stvsf	d8
cstart	1c		lpush	7c		rec0	88	rvsa2	c3	stvsw	9a
cstop	1b		lrele	16		rec1	01	rvsa4	c5	sub	45
cwait	40		lrese	7e		rec10	0a	rvsa6	c7	svsa0	f5
cwram	df		lt	34		rec11	0b	rvsb0	f0	svsa2	c2
div	4a		madd	3f		rec12	0c	rvsb1	b9	svsa4	c4
eq	32		mbset	d0		rec13	0d	rvsb2	83	svsa6	c6
escape	12	2	mbtes	d1		rec14	0e	rvsb3	bb	svsb0	f1
ge	37		mcis	61		rec15	0f	rvsb4	85	svsb1	99
gt	35		mcist	e1		rec2	02	rvsb5	bd	svsb2	82
index	6d		mcit	a1		rec3	03	rvsb6	87	svsb3	9b
indh0	38		mhalt	18		rec4	04	rvsb7	bf	svsb4	84
indh01	8d		mmul	3e		rec5	05	rvsd0	f6	svsb5	9d
indh02	ad		mnoop	2f		rec6	06	rvsd2	e3	svsb6	86
indh1	39		mod	4b		rec7	07	rvsd4	e5	svsb7	9f
indh11	cd		moveb	ac		rec8	08	rvsd6	e7	svsw0	f3
indh12	ed		movebs	11	1	rec9	09	rvsw0	f2	svsw2	a2
inpdv	6f		moveg	ae		rechd	e8	rvsw2	a3	svsw4	a4
inth0	c1		mrecha	81	3	rechw	a8	rvsw4	a5	svsw6	a6
intpa	6e		mrest	28		rechws	c8	rvsw6	a7	teqad	3c
intrs	6c		mstst	29		renhb	8f	setad	5f	tlock	3b
iocci	1d		msub	41		renpb	8e	setatm	58	tnill	3d
iocda	2a		mtime	17		resta	95	setcr	56	topen	3a
iogi	26	3	mtrh	2c		restc	d5	setdi	5e	uadd	42
iogo	25	3	mtrs	2e		revga	d3	seteq	59	uadhw	e6
ioibx	2b		mul	49		revgas	c9	setin	5d	uadhw1	30
ionci	27		mwi	20		revgb	93	setre	94	udiv	47
iorbb	71		mwis	60		revgbs	89	setsb	5a	ult	31
iorbbc	70		mwist	e0		revgd	f9	setsp	5b	umod	48
iorbw	75		mwit	a0		revgds	e9	setst	ec	umul	46
iorbwc	74		mwst	c0		revgw	b3	settm	57	usub	43
iors	22		mwt	80		revgws	a9	setun	5c	xor	4e
iorsc	2d		ne	33		revla	d7	shc	53		
iorw	23		neg	50		revlas	cb	shc8	54		
iowbb	73		not	55		revlb	97	stcea	ee		
iowbbc	72		or	4d		revlbs	8b	stnhb	8c		

1. RC5040 only = moveb. 2. RC5050 only. 3. RC5040 only.

G. PROCESS DESCRIPTOR LAYOUT

Dec		Hex	
0	timer	0	68
2	level	2	6A
4	delaychain	4	6C
6	exceptioncode	6	6E
8	exic	8	70
10		A	72
2	exceptionpoint	C	74
4		E	76
6	maxstack	10	78
8	regset	2	
20	mregset	4	
2	incstate	6	
4	programref	8	
6		A	
8	mbxchain	C	
30	refchain	E	
2		20	
4	processchain	2	
6	msgchain	4	
8		6	
40	programchain	8	
2	workmbx	A	
4		C	
6	statistic	E	
8		30	
50	secret pointer	2	
2	workref	4	
4		6	
6	heap	8	
8		A	
60	plinetable	C	
2	ownname	E	
4		40	
6		2	
8		4	
70		6	
2		8	
4	father	A	
6		C	
8	processret	E	

e347

H. MESSAGE HEADER LAYOUT

Dec			Hex
0	chain		0
2			2
4	u1	u2	4
6	u3	u4	6
8	messagekind		8
10	size		A
12	start		C
14			E
16	owner		10
18			12
20	answer		14
22	msgchain		16
24			18
26	stackchain		1A

I. HEAP LAYOUT

heap_head:

Dec		Hex
0	size	0
2	next_heap_head	2
4		4
6	next_free	6
8	no_of_free_bytes	8

heap_root:

Dec		Hex	
0	size	0	
2	first_heap_head	2	
4		4	
6	last_heap_head	6	
8	name_cat	8	
10		A	
12	free_names	C	
14		E	
16	.	10	heap_ret_chain
18	.	12	
	.		

name_entry:

Dec		Hex
0	next	0
2		2
4	mbxptr	4
6		6
8		8
10	name	A
12		C
14		E
16		10

J. EXCEPTION CODES

Code (Hex)	Meaning	
1	- parity error	
2	- registerstack error	
3	- undefined opcode	
4	- odd number of bytes	
5	- stack overflow	
6	- pointer = nil	
7	- signal:	reference = nil
	- push:	first param = nil
	- pop:	second param = nil
	- lock:	reference = nil
	- reference = nil	
8	- wait:	reference <> nil
	- pop:	first param <> nil
9	- push:	param locked
	- pop:	second param locked
	- signal:	reference locked
	- reference locked	
A	- lock overflow	
B	- arithmetic overflow	
C	- index out of bounds	
	- subrange out of bounds	
D	- illegal zonestate	
E	- field overflow	
F	- move wraparound	
10	- push: identical arguments	
11	- push: first param not empty	
12	- lock: size error	
	- size too small	
13	- top <= offset	
14	- lock: not data message	
	- not data message	
15	- not channel message	
16	- word block i/o: odd number of bytes	
17	- block i/o at level 0	
18	- setcr: first limit negative	
19	- setad: truncation error	
1A	- no resources	
1B	- file does not exist	
1C	- position outside file	
1D	- wrong answer	
1E	- setpriority: illegal priority	
1F	- pool: no core	
20	- process = nil	
21	- arithmetic overflow	
22	- double conversion exception	
23	- system error	
24	- illegal switch in case construction	

Code (Hex)	Meaning
25	- upper limit in call of succ
26	- lower limit in call of pred
27	- with: size error
28	- lockdata: top < computed top
29	- local reference variable not nil at routine exit
2A	- local process variable not nil at routine exit
2B-2E	- system error
2F	- break by father

K. RC5040 REGISTER LAYOUT

Description of the working registers: (for a more detailed description, refer to the reference manual, ref. 4)

W:	Index:	
00	000-007	regset 0
01	008-00F	regset 1
-	-	
79	3C8-3CF	regset 121
7A	3D0-3D7	work regset for multiplications
7B	3D8-3DF	masks0
7C	3E0-3E7	masks1
7D	3E8-3EF	breakpointset
7E	3F0-3F7	monitorset
7F	3F8-3FF	com8085

regset 0 thru regset 121:

1st reg.	ps
2nd reg.	sb
3rd reg.	gf
4th reg.	lf
5th reg.	lu
6th reg.	lm
7th reg.	ib
8th reg.	ic

breakpointset:

1st reg.	breakpointmode (8000 means breakpoint active)
2nd reg.	breakpointbase
3rd reg.	breakpointdisp
4th reg.	unused
5th reg.	unused
6th reg.	unused
7th reg.	unused
8th reg.	unused

monitorset:

1st reg.	memregsetbase
2nd reg.	memregsetdisp
3rd reg.	waitqueuelast
4th reg.	waitqueuefirst
5th reg.	monitorlevel
6th reg.	unused
7th reg.	unused
8th reg.	unused

com8085:

1st reg.	parityerrorregset
2nd reg.	fifo0, fifo1
3rd reg.	fifo2, fifo3
4th reg.	fifo4, fifo5
5th reg.	cow (value, disp)
6th reg.	message.errorcode
7th reg.	parityerrorbase
8th reg.	parityerrordisp

L. RC5050 REGISTER LAYOUT

Description of the registers: (for a more detailed description, refer to the reference manual, ref. 4)

W:	Index:	
00	000-007	regset 0
01	008-00F	regset 1
-	-	
79	3C8-3CF	regset 121
7A	3D0-3D7	not used for compatibility reasons
-	-	
7E	3F0-3F7	not used for compatibility reasons
7F	3F8-3FF	com8085, for compatibility reasons
80	400-407	regset 128
-	-	
1FF	FF8-FFF	regset 511
200	1000-1007	regset 0 extra registers
-	-	
3FF	1FF8-1FFF	regset 511 extra register

regset 0 thru regset 121 and 128 thru 511:

1st reg.	ps	Process Status Word
2nd reg.	sb	Stack Base
3rd reg.	gf	Global Frame
4th reg.	lf	Local Frame
5th reg.	lu	Last Used
6th reg.	lm	stack LiMit
7th reg.	ib	Instruction Base
8th reg.	ic	Instruction Counter

regset 512 thru regset 1023:

1st reg.	th	cpu Timer High
2nd reg.	tl	cpu Timer Low
3rd reg.	ch	register CHain
4th reg.	rb	Reference Base
5th reg.	rd	Reference Displacement
6th reg.	hb	Head Base
7th reg.	hd	Head Displacement
8th reg.	cw	Control Word

M. CONTROL MICRO RAM LAYOUT

00 RTC level:

Real Time Clock interruption level.

01 TTO level:

Console output interruption level.

02 TTI level:

Console input interruption level

04-03 Timer:

Low	High
3	4

RC5000 is interrupted on level 'RTC level' every (256 x High + Low) x 2.5 mS.

06-05 Watchdog:

Low	High
5	6

Count down is performed every 2.5 mS. RC5000 is reset, when (if) watchdog.high is decremented from 1 to 0.

07 TTI input:

When an input character from the console is ready, the character is delivered here and RC5000 interrupted on level 'TTI level'.

08 TTO output:

RC5000 delivers an output character for the console here, and is interrupted on level 'TTO output', when the character is printed.

09 Version:

Model	Micro

9

Model is 1 for RC3502-1, 2 for RC5040 (RC3502-2) and 5 for RC5050. Micro is the micro program version number.

0A-0B Switches:

10		11	
0-3	4-7	8-11	12-15
A		B	

Contains the current value of the autoload switches. For RC5040 this is the four switches marked BUS on the Processor Front Panel. For RC5050 this is a displayable NVM parameter.

0C-1B Interruption Levels:

Describes the 128 interruption level configuration. Level i exists if bit no i is set in the Interruption Level bitmask. Levels 0-7 are described in byte 12, Levels 8-15 in byte 13, etc.

1C-21 RAM Modules:

Is a bitmask, describing the RAM module configuration. A module is 64Kb. A module exists, if the corresponding bit is set in the bitmap.

Modules A0, A2, ..., AE are described in byte 28, modules B0, B2, ..., BE in byte 29, etc.

22-27 EPROM Modules:

Is a bitmask, describing the EPROM module configuration. A module is at most 64Kb. A module exists, if the corresponding bit is set in the bitmap. Modules A0, A2, ..., AE are described in byte 34, modules B0, B2, ..., BE in byte 35, etc.

28-29 ID register:

40	41
0-7	8-15
28	29

The contents of the F505 (IDR202) register or for RC5050 the NVM value may be examined here.

2A COM204 mask:

Controls the COM204 channels during down line load. If bit i is set the channel is skipped, otherwise a load request frame is transmitted on the channel.

2B Result:

If a down line load is successful, this byte contains the resultant relative channel plus 8, where 0 <= channel <= 7.

If autoload is from Winchester Disc, the CU address of the RC5000 on the SCSI bus is passed here.

2C-33 Entryname:

Controls autoload from Winchester Disc.

If the fist character in Entryname is zero, the default name 'boot3502x' is used, where x is the CU address of the RC5000 on the SCSI bus.
Otherwise Entryname is used for lookup in the catalog 'Autoloadcat'.

34-35 Cswitches:

Contains a copy of the field switches.

The contents of switches is transferred to Cswitches, whenever the autoloadprogram is started, and the first byte of Cswitches equals zero. The contents of Cswitches controls the autoload program.

36 Printcount:

Controls the number of times autoload announcements is performed.

N. RC5050 NON-VOLATILE MEMORY LAYOUT.

NVM	Y-mem	Note	Description
00 .. 01	0a..0b		Load switches.
02	-		Mode Switch w.o. baudrate.
03	-		Baud Rate.
04 .. 05	28..29		ID-register, this ID register value will be used only in the absence of a hardware ID register such as F505 (IDR202).
06	-		Enter Degraded Mode at restart, internal SP usage only.
07	-		System mode, normal value zero. +80 Generate CWRAM testoutput. +40 Do not Run after startup and CW command. +20 Disable Watch Dog restart. +01 Disable compatible mode.
08	-	3	Gate Array SMEM high byte.
09	-	3	Gate Array SMEM low byte.
0a	-	4	Gate Array mode.
0b	-	5	First DMA module.
0c	-	1	Latest Error Type.
0d	-	1	Latest Error No.
0e .. 0f	-	1	Latest Error Address.
10 .. 11	-	1	Latest Error Data.
12 .. 13	-	1	Latest Error Module.
14	-	2	Oldest Error Type.
15	-	2	Oldest Error No.
16 .. 17	-	2	Oldest Error Address.
18 .. 19	-	2	Oldest Error Data.
1a .. 1b	-	2	Oldest Error Module.
1c .. 1d	-		Flash 0 burn count.
1e .. 1f	-		Flash 1 burn count.
20 .. 2f	-	6	Extended Memory Inclusion Table
32 .. 61	-		Reserved.
62 .. 63	-		CheckSum for addr 0..61.
64 ..127	-		FCO's followed by 0's, to be maintained by service organization.
128 ..255	-		SW EEPROM area, to be maintained by system software.

Notes:

1. These locations are set to zero when the Service Processor starts or restarts. Whenever a testprogram or hardware error occurs these locations are set accordingly, see section 2.3.1.3 .

2. Whenever a testprogram or hardware error occurs and these locations are zero they are set to the old value of the locations marked 1.

3. The Gate Array control register SMEM controls the usage of streaming memory. After microprogram load the SP transfers the contents of these NVM bytes to the Gate Array SMEM register. The default value of these bytes are 00 00.

4. The Gate Array mode register controls various aspects of the Gate Array functions. After microprogram load the SP transfers the contents of this NVM bytes to the Gate Array mode register. The default value of this byte is 28.

5. This location indicates the address of the first module to be addressed by DMA-based controllers. The value is intended for use by the allocator. The default value of this byte is 80.

6. This 16 bytes constitute a boolean array of all possible memory modules. A value of true indicates that the corresponding memory module will be tested for presense and type during CPU initiation. A value of false will prohibit any testing of the corresponding module, i.e. the module will unconditionally be excluded. The default value, which excludes dual-port controllers and extended memory is: 00 00 00 00 00 00 00 00 00 00 ff ff ff ff ff ff.

O. RC5050 WORK MEMORY LAYOUT

The RC5050 CPU is equipped with a fast work memory consisting of 16 x 128 16 bit words. 128 of these are used as interrupt map while the rest is used for various purposes.

Wm	Index	Ws	Description
0	00..7e	-	Interrupt Map,. i.e. Wm0(nxs) == NextCurReg
0	7f	0	Interrupt Map,. i.e. Wm0(7f) == NextCurReg
1	00	-	ParityErrorCode
1	01..7e	-	n/u
1	7f	1	Service Proc Command 1
2	00	-	ParityBase
2	01..7e	-	n/u
2	7f	2	Service Proc Command 2
3	00	-	i.e. ParityDisp
3	01..7e	-	n/u
3	7f	3	Current GA mode
4	00	-	TestMode Register: 0001 Repeat/Resume Instruction does not count down stepcount 0002 Use TestMode exception action 8000 Generate itc at stepmode entry
4	01	-	Esc 3 instruction answer.
4	02..7e	-	n/u
4	7f	4	BreakPointMode
5	00..7e	-	n/u
5	7f	5	BreakPointBase
6	00..7e	-	n/u
6	7f	6	BreakPointDisp
7	00..7e	-	n/u
7	7f	7	SpMsg
8	00..7e	-	n/u
8	7f	8	SpMsg1
9	00..7e	-	n/u
9	7f	9	SpMsg2
10	00..7e	-	n/u
10	7f	10	Current Level
11	00..7e	-	n/u
11	7f	11	StopMode: Current RegSet All of Wm12 survives watchdog restart.
12	00..07	-	Used by SP: ROM locations
12	08..0f	-	Used by SP: RAM locations
12	10..7e	-	Available for SW usage.
12	7f	12	PS_waitbits
13	00..0f	-	Bits ($0001..$8000)
13	10..7e	-	n/u
13	7f	13	Local Work Var
14	00..7e	-	Move / Block Buffer
14	7f	14	Local Work Var
15	00..0f	-	Masks ($0001..$ffff)
15	10..7e	-	n/u
15	7f	15	Local Work Var

P. SERVICE PROCESSOR INITIATOR

The RC5050 Service Processor Initiator is activated after power-up and after a C R or C I command. Normally the Service Processor, which is located in the builtin EPROM in the Service Processor, will execute initial tests and then activate the main Service Processor program located in EEPROM number 0.

In below cases, however, the Service Processor Initiator will not activate the main SP but instead activate a small command interpreter:

1. When a checksum error is detected in EEPROM 0 or 1.

2. When the SP Initiator is activated by the C I command.

3. When one of the front panel switches are pressed when the SP Initiator starts. Note that this will select a console baud rate as well.

The activated command interpreter uses "-*" as prompt and accepts the following commands:

E Load EEPROM.

This command activates a EEPROM load sequence. First the EEPROM number is quiried, then an optional text describind the load is quiried, and finally the load image in Intel MCS-8 format is read.
For EEPROM 0 (Service Processor program) record types 0 and 1 are accepted only.
For EEPROM 1 (Microprogram) record types 1, 80, 81, 82, and 83 are accepted.

F Force full Service processor function.

This command will cause an unconditional transfer to the Service Processor main program in EEPROM 0.

I N Inspect NVM.

Inspect Non-Volatile Memory. This command will display and optionally change a single byte location in the NVM.

I S Inspect SP-RAM.

This command will display 16 bytes of SP RAM memory. Update is not possible.

N Clear NVM memory.

All of the NVM except speed and mode are set to zero.

? Display available SP Initiator commands.

Q. INSTALLATION STANDARDS AND RECOMMENDATIONS

This is a recommendation concerning installation of hardware modules in the RC5000. The guidelines concern interruption levels, input/output priorities, module number selections, DMA priority.

Q.1 Input/Output Modules

The priorities and interruption levels in the following table should be followed according to input/output channels and priority. The first module in a group should be the lowest channel number.

Q.2 Memory Modules

Memory modules are strapped to cover one or more module addresses in the memory address space. They must not overlap. There must exist at least one RAM memory module with address C0 and at least one PROM memory module containing the BOOT program in module address E0. If the autoloaded programs occupy more than one module, there must be RAM memory modules with address C2, C4, etc. MEM204 RAM memory modules can be placed in the address space C0-DE. MEM205, MEM206, MEM207, and MEM209 RAM memory modules can use 80-FE. A special action can be taken on MEM204 to disable the PROM area, which is always equal to RAM module address + 20 (hex). The PROM memory module address on MEM205 and MEM206 may be set completely free in the whole address space, or may be disabled. The PROM areas E0 or E2 may be disabled on MEM207 and MEM209.

TES201 and TES202 can be strapped in the address space C0-FE. (Note, that C0 must always be used as RAM memory).

Q.3 Controllers

COM204 occupies 2 positions. The one position connects to the Back Plane Bus. In the second position a CBL714 must be placed.

By convention the first COM204 must be strapped to dual port memory module address 90 (hex) and interruption level 72 (decimal). See section 4.1.4.

IMS208 through IMS212 occupies one position.

By convention the first IMS2XX must be strapped to dual port memory module address 80 (hex) and interruption level 8. The succeeding IMS2XX controllers must be set to 80.4000 and level 9, i.e. memory module addresses increase by 4000 (hex) and interruption levels by 1.

Q.4 Summary

CPU215	CPU216	CPU217	2. IOM202	1. IOM202	2. COM204	1. COM204	2. VCO201	1. VCO201

->

->

2. COM205	1. COM205	2. MFC20X	1. MFC20X	2. IMS2XX	1. IMS2XX	2. MEM20X	1. MEM20X	2. TES20X	1. TES20X

CPU221	2. IOM203	1. IOM203	2. COM204	1. COM204	2. VCO201	1. VCO201	2. COM206	1. COM206

->

->

2. COM205	1. COM205	2. MFC20X	1. MFC20X	2. IMS2XX	1. IMS2XX	2. MEM20X	1. MEM20X	2. TES20X	1. TES20X

The following three points should be observed:

1. The interruption level priority chain starts at the CPU and must not be broken by empty positions until the last module, which uses interruption level priority. At the end of the chain, there must be an "interrupt chain end plug" (CBL735).

2. The DMA priority chain starts at the CPU and must not be broken by empty positions until the last module, which uses DMA priority.

3. The module, which is closest to the CPU, has highest priority, both according to interrupt level and DMA.

Module name	Interrupt level (Hex)	Dual Port Memory Base (Hex)	No. of interrupt levels (Decimal)	No. x salesnumber
	Lowest priority			
CPU	00-04	-	5	
SPARE	05-07	-	3	
IMS2XX	08-0F	80-8E	8	8 x RC5330
MFC20X	10-1F	-	16	16 x RC5320
COM206	20-2F	-	16	16 x RC5371
COM205	20-2F	-	16	16 x RC5370
VCO201	30-37	-	8	8 x RC3583
COM204	48-4F	90-96	8	8 x RC5340
IOM202	50-6D	-	32	4 x RC5310
MCC240	6E-6F	-	2	1 x RC5010
SPARE	78-7E	-	4	
	Highest priority			

R. ASCII CHARACTER SET

ASCII	dec	hex	ASCII	dec	hex	ASCII	dec	hex	ASCII	dec	hex
NUL	00	00	space	32	20	@	64	40	`	96	60
SOH	01	01	!	33	21	A	65	41	a	97	61
STX	02	02	"	34	22	B	66	42	b	98	62
ETX	03	03	§	35	23	C	67	43	c	99	63
EOT	04	04	$	36	24	D	68	44	d	100	64
ENQ	05	05	%	37	25	E	69	45	e	101	65
ACK	06	06	&	38	26	F	70	46	f	102	66
BEL	07	07	'	39	27	G	71	47	g	103	67
BS	08	08	(	40	28	H	72	48	h	104	68
HT	09	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	Æ	91	5B	æ	123	7B
FS	28	1C	<	60	3C	Ø	92	5C	ø	124	7C
GS	29	1D	=	61	3D	Å	93	5D	å	125	7D
RS	30	1E	>	62	3E	Ü	94	5E	ü	126	7E
US	31	1F	?	63	3F	_	95	5F	DEL	127	7F

S. INDICES

S.1 Survey of Figures

S.2 Catchword Index