1 Introduction

1.1 Motivation

Perform detailed failure analysis on defective RAM cells:

- Analyse fail modes of one specific part and locate suspicious memory cells or peripheral cells, respectively, in the physical RAM layout.
- Analyse fail modes of one or more parts and perform a statistical analysis of the failure rate of memory cells or peripheral cells, respectively.

1.2 Workflow

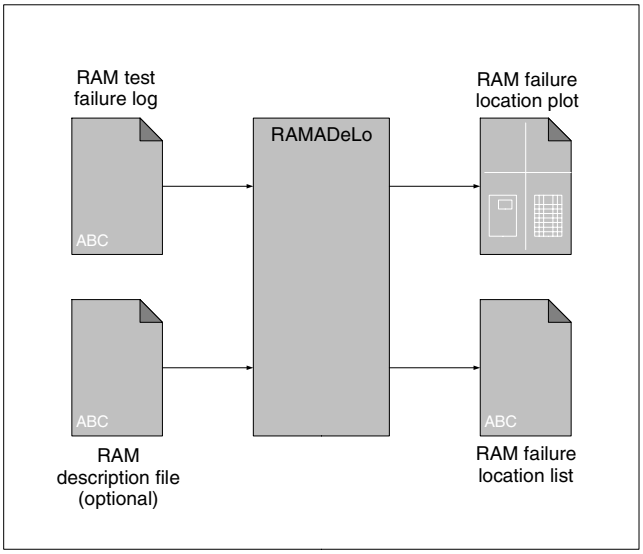


Figure 1-1: Workflow detail analysis

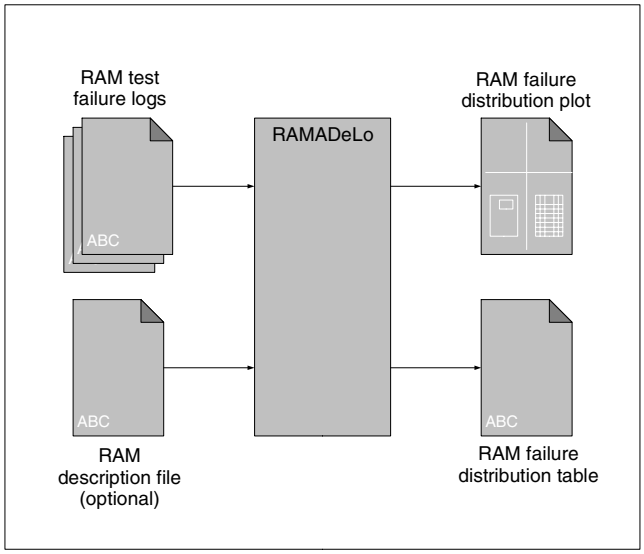


Figure 1-2: Workflow statistical analysis

- Create description of the RAM layout and topology by a project specific control file
- Create failure log of the 6N RAM test data of the device(s) under test
- Run RAMADeLo on these files in order to perform the requested analysis

2 RAM description file

2.1 General file format

- Plain ASCII text file
- Comments and comment lines start with a semicolon
- Empty or blank lines are allowed without any restrictions
- All parameters are case sensitive
- General parameter syntax: <key> = <value>[,<value>]

2.2 Supported parameters

Keyword	Value	Num. format	Unit	Default
CSX	Chip size, X dimension	float	µm	520.2
CSY	Chip size, Y dimension	float	µm	1063.6
ROX	Origin of RAM cell within chip, X coordinate	float	µm	0.0
ROY	Origin of RAM cell within chip, Y coordinate	float	µm	0.0
ROT	Rotation of RAM cell within chip	keyword (0, 90, 180, 270, +X, -X, +Y, -Y)	-	0
OFX	Offset of lower left bit cell within RAM cell, X coordinate ①	float	µm	4.0
OFY	Offset of lower left bit cell within RAM cell, Y coordinate ①	float	µm	1.8
RPX	Width of bit cell, X dimension ①	float	µm	14.2
RPY	Height of bit cell, Y dimension ①	float	µm	27.6
WSX	Width of word decoder cell, X dimension ①	float	µm	57.8
ASY	Height of R/W amplifier cell, Y dimension ①	float	µm	178.6
ESA	Logical start address to be evaluated	integer (dec/hex)	1	0x0000
ESZ	Size of RAM to be evaluated	integer (dec/hex)	1	unlimited
HYP	Hypertrophic fail margin	integer (dec/hex)	1	unlimited
COL	Column mapping between physical and logical bit positions. 1st value is physical column index, 2nd value is logical column index	integer (dec/hex)	1	L08 std. RAM mapping

① Coordinates and dimensions within RAM cell always refer to default rotation of the RAM cell. Please see chapter 2.3 for further details.

2.3 Reference of layout dimensions

The specification of layout dimensions is needed to plot the position of the RAM cell correctly inside the chip boundary and to create the RAM detail view true to scale. Additionally these dimensions will be used to calculate the coordinates in the detailed RAM failure list.

If true to scale plots are not needed, these layout dimensions are not mandatory. If they are not specified by the user, the dimensions of the L08 RAM standard cell will be used by default.

2.3.1 Rotation +X (0° cw)

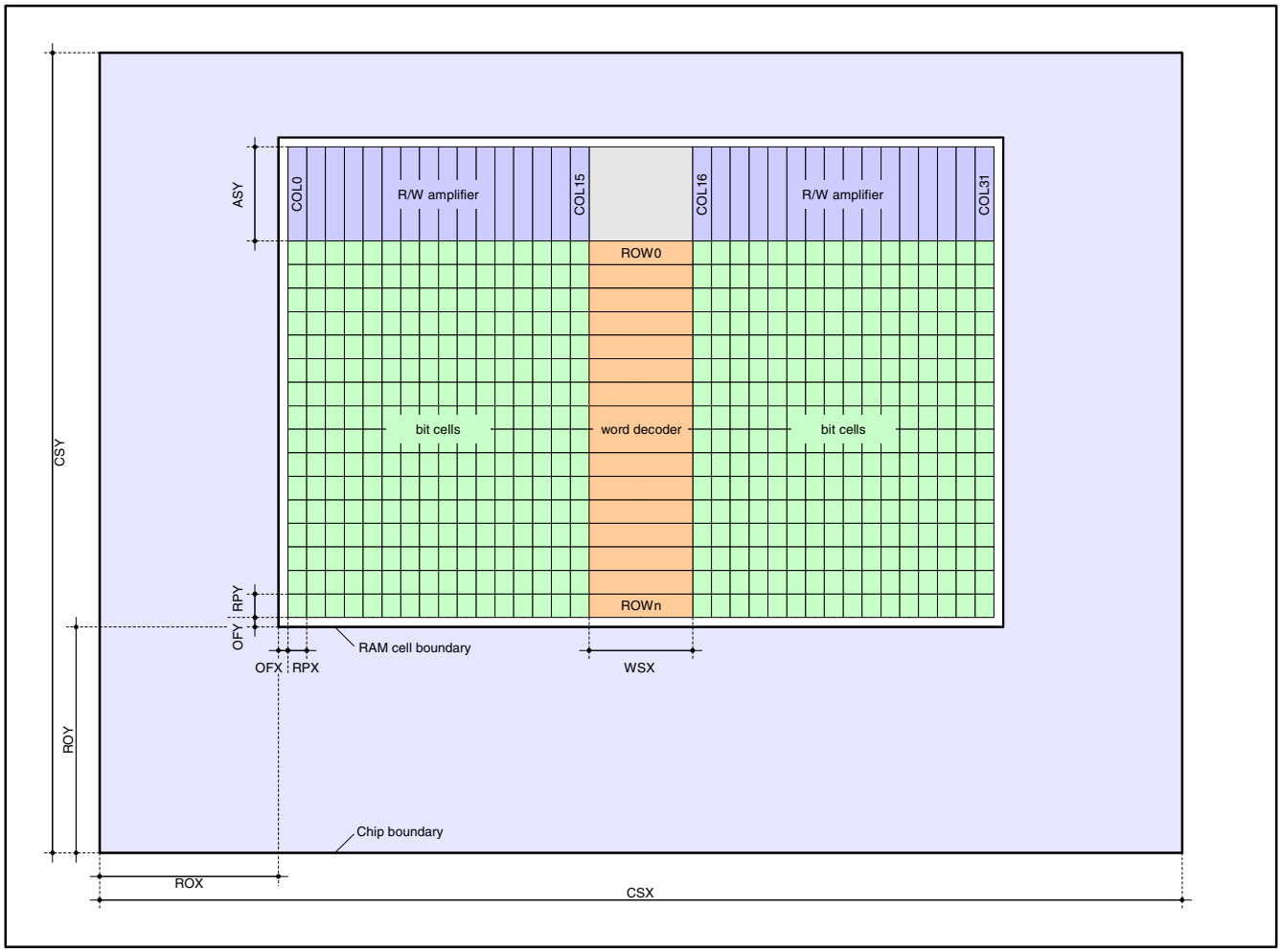


Figure 2-1: Dimensions in orientation +X

2.3.2 Rotation -Y (90° cw)

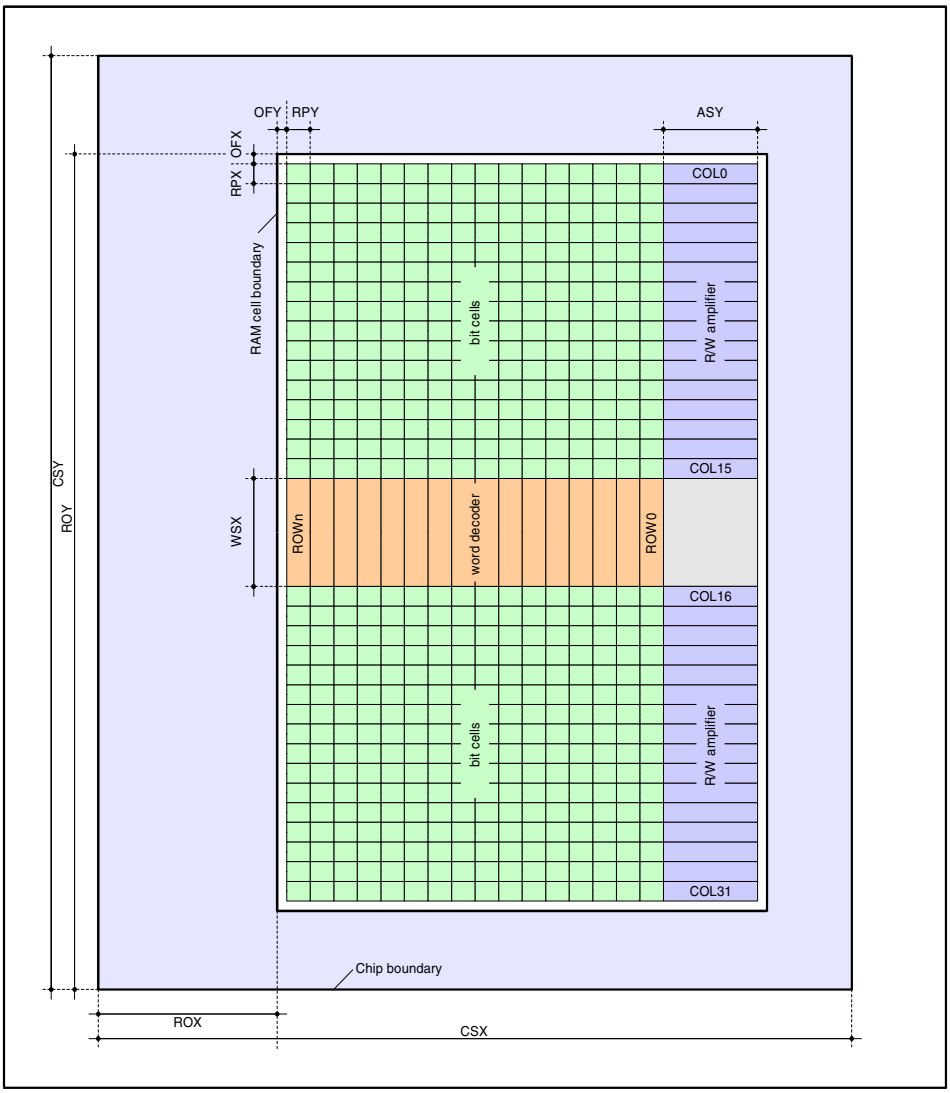


Figure 2-2: Dimensions in orientation -Y

2.3.3 Rotation -X (180° cw)

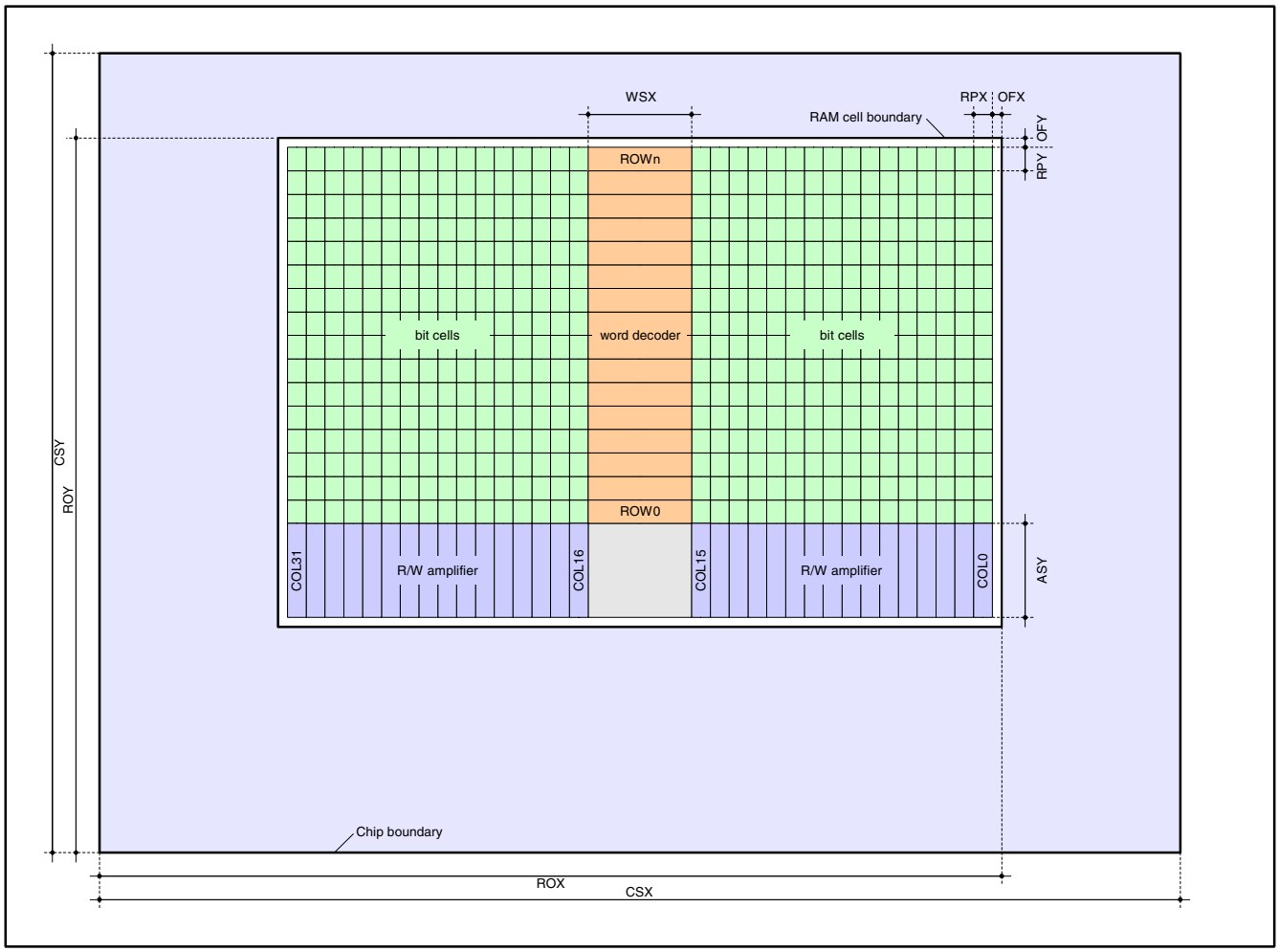


Figure 2-3: Dimensions in orientation -X

2.3.4 Rotation +Y (270° cw)

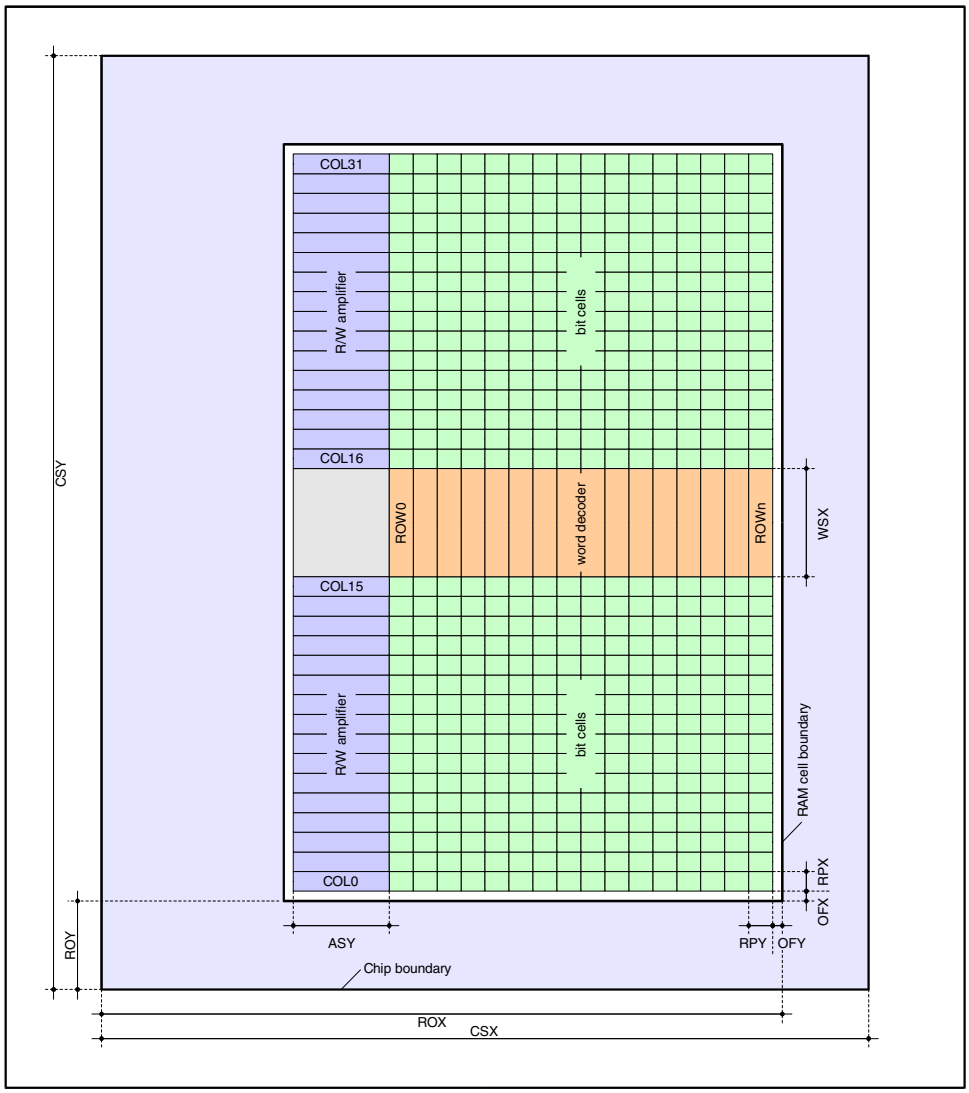


Figure 2-4: Dimensions in orientation +Y

2.4 Specification of address ranges

The logical address range to be evaluated can be defined by the parameters ESA and ESZ. The first RAM address to be evaluated equals ESA, the last address to be evaluated equals (ESA + ESZ -1).

Start address and address range may be specified in decimal notation as well as in hexadecimal notation. Hexadecimal numbers are introduced by the prefix “0x”.

If no start address is specified, 0x0000 is used as default start address. If no address range is supplied, RAMADeLo will evaluate the full address range contained in the RAM test log files.

The address range features have been implemented to be able to support ASICs with more than one RAM block inside. The maximum size of the default RAM is 256 bytes. If more memory space is required, additional RAM blocks may be added to the design, mapped to another location within the microcontroller's address range. In order to restrict the failure evaluation to one certain RAM block, the ESA and ESZ parameters may be used.

2.5 Hypertrophic fail margin

In order to avoid that parts with catastrophic failures are contributing to the evaluation, a hypertrophic fail margin can be supplied. Catastrophic failures may be e.g. a failures within the microcontroller or the test controller which impede a reliable access to the RAM cells.

The hypertrophic fail margin is the maximum number of bit failures that are accepted before a part is considered to be a catastrophic failure and will be dismissed. If the number of bit failures is equal or greater that the hypertrophic fail margin value, the part will be dismissed. If the number of bit failures is less than the hypertrophic fail margin, the part will be evaluated.

The hypertrophic fail margin may be specified in decimal notation as well as in hexadecimal notation. Hexadecimal numbers are introduced by the prefix “0x”.

2.6 Mapping of physical and logical columns

For a proper location of electrical failures within the RAM cell's layout, the mapping between logical bit position and physical RAM layout has to be defined. The physical RAM layout is composed from rows which contain 32 bit cells. These bit cells are located in four consecutive bytes in the address space of the microcontroller.

By definition, the leftmost column in the RAM layout in default rotation (R/W amps at the upper edge, please refer to chapter 2.3.1) has the physical column index 0, while the rightmost column has the index 31. The logical bit position is composed from the two least significant binary digits of the byte address and the bit address according to the following equation:

$$LCOL = [8 * (BYTEADR \mod 4)] + BITADR$$

The COL parameter which is intended to specify the mapping between physical and logical columns takes two arguments. The first argument is the physical column index, the second argument is the logical column index. In order to specify the full correspondence table, the COL parameter has to be appear 32 times with different pairs of physical/logical column indices.

For a complete correspondence table each physical column index from 0 to 31 and each logical column index from 0 to 31 should to be contained in the argument list. In case of an ambiguous or incomplete mapping the software will terminate with an error message.

If no mapping information is specified at the command line or in the configuration file, the default mapping of the L08 RAM will be applied.

The following table shows the default mapping between physical and logical columns.

Phys. column	Log. column	Byte address	Bit value
0	16	xxxxxxxx xxxxxx10$_b$	00000001$_b$
1	17	xxxxxxxx xxxxxx10$_b$	00000010$_b$
2	18	xxxxxxxx xxxxxx10$_b$	00000100$_b$
3	19	xxxxxxxx xxxxxx10$_b$	00001000$_b$
4	20	xxxxxxxx xxxxxx10$_b$	00010000$_b$
5	21	xxxxxxxx xxxxxx10$_b$	00100000$_b$
6	22	xxxxxxxx xxxxxx10$_b$	01000000$_b$
7	23	xxxxxxxx xxxxxx10$_b$	10000000$_b$
8	24	xxxxxxxx xxxxxx11$_b$	00000001$_b$
9	25	xxxxxxxx xxxxxx11$_b$	00000010$_b$
10	26	xxxxxxxx xxxxxx11$_b$	00000100$_b$
11	27	xxxxxxxx xxxxxx11$_b$	00001000$_b$
12	28	xxxxxxxx xxxxxx11$_b$	00010000$_b$
13	29	xxxxxxxx xxxxxx11$_b$	00100000$_b$
14	30	xxxxxxxx xxxxxx11$_b$	01000000$_b$
15	31	xxxxxxxx xxxxxx11$_b$	10000000$_b$
16	7	xxxxxxxx xxxxxx00$_b$	10000000$_b$
17	6	xxxxxxxx xxxxxx00$_b$	01000000$_b$
18	5	xxxxxxxx xxxxxx00$_b$	00100000$_b$
19	4	xxxxxxxx xxxxxx00$_b$	00010000$_b$
20	3	xxxxxxxx xxxxxx00$_b$	00001000$_b$
21	2	xxxxxxxx xxxxxx00$_b$	00000100$_b$
22	1	xxxxxxxx xxxxxx00$_b$	00000010$_b$
23	0	xxxxxxxx xxxxxx00$_b$	00000001$_b$
24	15	xxxxxxxx xxxxxx01$_b$	10000000$_b$
25	14	xxxxxxxx xxxxxx01$_b$	01000000$_b$
26	13	xxxxxxxx xxxxxx01$_b$	00100000$_b$
27	12	xxxxxxxx xxxxxx01$_b$	00010000$_b$
28	11	xxxxxxxx xxxxxx01$_b$	00001000$_b$
29	10	xxxxxxxx xxxxxx01$_b$	00000100$_b$
30	9	xxxxxxxx xxxxxx01$_b$	00000010$_b$
31	8	xxxxxxxx xxxxxx01$_b$	00000001$_b$

2.7 Sample RAM description file

Sample RAM description file m14013c.cnf

```
;-------------------------------------------------------------------------;
;                        RAM configuration file for 140.13                ;
;-------------------------------------------------------------------------;

;-------------;
; Layout data ;
;-------------;

CSX = 7510.0   ; Chip size X [um]
CSY = 7510.1   ; Chip size Y [um]
ROX =   33.9   ; RAM origin X [um]
ROY = 5788.0   ; RAM origin Y [um]
ROT =     90   ; RAM rotated 90° CW
OFX =    9.6   ; Offset to first bit cell X [um]
OFY =    8.8   ; Offset to first bit cell Y [um]
RPX =   32.3   ; RAM pitch X [um]
RPY =   20.9   ; RAM pitch Y [um]
WSX =  120.5   ; RAM WORD64 block width X [um]
ASY =  235.9   ; RAM R/W amplifier block height Y [um]

;----------------------;
; Logical RAM size data ;
;----------------------;

ESA = 0x0000   ; Logical start address for evaluation
ESZ = 0x0080   ; Logical RAM size to be evaluated
HYP =   1000   ; Hypertrophic fail margin

;---------------------------------------;
; Mapping of logical bit to physical bits ;
;---------------------------------------;

COL = 00,16    ; Physical column  0 mapped to logical column 16 (byte 2, bit 0)
COL = 01,17    ; Physical column  1 mapped to logical column 17 (byte 2, bit 1)
COL = 02,18    ; Physical column  2 mapped to logical column 18 (byte 2, bit 2)
COL = 03,19    ; Physical column  3 mapped to logical column 19 (byte 2, bit 3)
COL = 04,20    ; Physical column  4 mapped to logical column 20 (byte 2, bit 4)
COL = 05,21    ; Physical column  5 mapped to logical column 21 (byte 2, bit 5)
COL = 06,22    ; Physical column  6 mapped to logical column 22 (byte 2, bit 6)
COL = 07,23    ; Physical column  7 mapped to logical column 23 (byte 2, bit 7)
COL = 08,24    ; Physical column  8 mapped to logical column 24 (byte 3, bit 0)
COL = 09,25    ; Physical column  9 mapped to logical column 25 (byte 3, bit 1)
COL = 10,26    ; Physical column 10 mapped to logical column 26 (byte 3, bit 2)
COL = 11,27    ; Physical column 11 mapped to logical column 27 (byte 3, bit 3)
COL = 12,28    ; Physical column 12 mapped to logical column 28 (byte 3, bit 4)
COL = 13,29    ; Physical column 13 mapped to logical column 29 (byte 3, bit 5)
COL = 14,30    ; Physical column 14 mapped to logical column 30 (byte 3, bit 6)
COL = 15,31    ; Physical column 15 mapped to logical column 31 (byte 3, bit 7)
COL = 16,07    ; Physical column 16 mapped to logical column  7 (byte 0, bit 7)
COL = 17,06    ; Physical column 17 mapped to logical column  6 (byte 0, bit 6)
COL = 18,05    ; Physical column 18 mapped to logical column  5 (byte 0, bit 5)
COL = 19,04    ; Physical column 19 mapped to logical column  4 (byte 0, bit 4)
COL = 20,03    ; Physical column 20 mapped to logical column  3 (byte 0, bit 3)
COL = 21,02    ; Physical column 21 mapped to logical column  2 (byte 0, bit 2)
COL = 22,01    ; Physical column 22 mapped to logical column  1 (byte 0, bit 1)
COL = 23,00    ; Physical column 23 mapped to logical column  0 (byte 0, bit 0)
COL = 24,15    ; Physical column 24 mapped to logical column 15 (byte 1, bit 7)
COL = 25,14    ; Physical column 25 mapped to logical column 14 (byte 1, bit 6)
COL = 26,13    ; Physical column 26 mapped to logical column 13 (byte 1, bit 5)
COL = 27,12    ; Physical column 27 mapped to logical column 12 (byte 1, bit 4)
COL = 28,11    ; Physical column 28 mapped to logical column 11 (byte 1, bit 3)
COL = 29,10    ; Physical column 29 mapped to logical column 10 (byte 1, bit 2)
COL = 30,09    ; Physical column 30 mapped to logical column  9 (byte 1, bit 1)
COL = 31,08    ; Physical column 31 mapped to logical column  8 (byte 1, bit 0)

;-------------------------------------------------------------------------;
; End of configuration file                                               ;
;-------------------------------------------------------------------------;
```

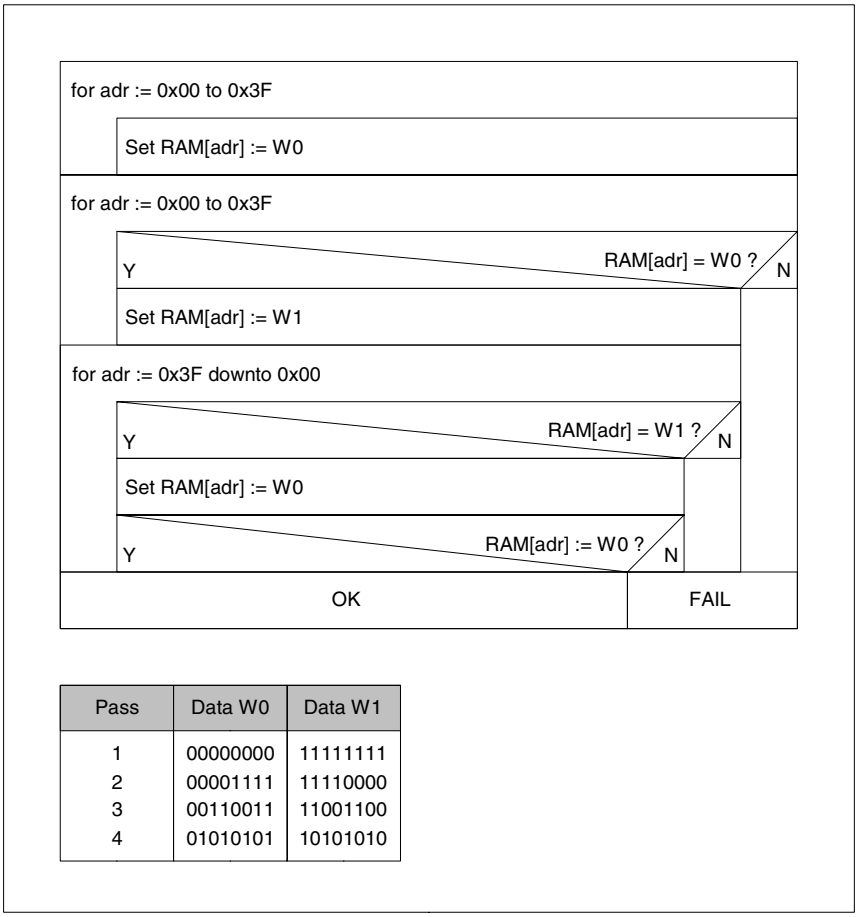
3 RAM failure log file

3.1 General file format

- Plain ASCII text file
- Shall be generated automatically by the ATE equipment from the 6N RAM test return data
- Each valid data line contains the byte address, twelve return data samples from the algorithm and a PASS/FAIL rating of the test results of this byte.
- Each line which contains the keyword “PASS” or “FAIL” is interpreted as data line, each line which does not contain one of these keywords is interpreted as comment lines
- Whitespace or “|” characters may be used to format the log file
- Empty or blank lines are allowed without any restrictions

3.2 6N RAM test algorithm

The 6N RAM test algorithm to be implemented in the ATE test program is shown in the following flow chart. The algorithm itself will be run four times using different sets of data for W0 and W1. This algorithm is capable of detecting stuck-at failures of the RAM cells as well as cross connections between different bytes and bits within a common byte. Also the RAM address decoding is covered by this algorithm.



Pass	Data W0	Data W1
1	00000000	11111111
2	00001111	11110000
3	00110011	11001100
4	01010101	10101010

Figure 3-1: 6N RAM test algorithm

In this example the RAM size is 64 bytes, which means that the loop counter goes from 0x00 to 0x3F. For other RAM sizes this loop counter will have to be adapted.

Sample PASCAL code for 6N RAM test

```
{***********************}
{* Compose test values *}
{***********************}

wd0[0] := 16#00;
wd1[0] := 16#FF;
wd0[1] := 16#0F;
wd1[1] := 16#F0;
wd0[2] := 16#33;
wd1[2] := 16#CC;
wd0[3] := 16#55;
wd1[3] := 16#AA;

for adr := 0 to 63 do
   fflg[adr] := FALSE;

{*****************************}
{* Run 6N RAM test algorithm *}
{*****************************}

for pass := 0 to 3 do
begin
   {*******************************}
   {* Write W0 data into RAM cells *}
   {*******************************}

   for adr := 0 to 63 do
      PokeRegister(adr,wd0[pass]);

   {***********************************************}
   {* Read and check RAM contents and write W1 data *}
   {***********************************************}

   for adr := 0 to 63 do
   begin
      rd0[i,adr] := PeekRegister(adr);
      if (rd0[i,adr]<>wd0[pass]) then
      begin
         fflg[adr] := TRUE;
         if (adr<fadr) then
            fadr := adr;
      end;

      PokeRegister(adr,wd1[pass]);
   end;

   {***********************************************}
   {* Read and check RAM contents and write W0 data *}
   {***********************************************}

   for adr := 63 downto 0 do
   begin
      rd1[pass,adr] := PeekRegister(adr);
      if (rd1[pass,adr]<>wd1[pass]) then
      begin
         fflg[adr] := TRUE;
         if (adr<fadr) then
            fadr := adr;
      end;

      PokeRegister(adr,wd0[pass]);

      rd2[pass,adr] := PeekRegister(adr);
      if (rd2[pass,adr]<>wd0[pass]) then
      begin
         fflg[adr] := TRUE;
         if (adr<fadr) then
            fadr := adr;
      end;
   end;
end;
```

3.3 Sample RAM failure log file

Sample RAM failure log

	PASS1 (0x00/0xFF)			PASS2 (0x0F/0xF0)			PASS3 (0x33/0xCC)			PASS4 (0x55/0xAA)			
ADR	RD0	RD1	RD2	RD0	RD1	RD2	RD0	RD1	RD2	RD0	RD1	RD2	P/F
0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x01	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x02	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x03	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x04	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x05	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x06	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x07	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x08	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x09	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x0A	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x0B	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x0C	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x0D	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x0E	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x0F	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x10	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x11	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x12	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x13	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x14	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x15	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	FAIL
0x16	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x17	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x18	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x19	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	FAIL
0x1A	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x1B	0xFF	0x00	0xFF	0xF0	0x0F	0xF0	0xCC	0x33	0xCC	0xAA	0x55	0xAA	FAIL
0x1C	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x1D	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x1E	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x1F	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x20	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x21	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x22	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x23	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x24	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x25	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x26	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x27	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x28	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x29	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2A	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2B	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2C	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2D	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2E	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x2F	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x30	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x31	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x32	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x33	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x34	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x35	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x36	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x37	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x38	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x39	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3A	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3B	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3C	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3D	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3E	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS
0x3F	0x00	0xFF	0x00	0x0F	0xF0	0x0F	0x33	0xCC	0x33	0x55	0xAA	0x55	PASS

4 Running RAMADeLo

4.1 General operating modes

4.1.1 Online help mode

In order to obtain an online help the command line option **-h** may be used. In online help mode the tool will generate an command summary of RAMADeLo. Please refer to chapter 4.5 for a complete dump of the online help output.

4.1.2 Revision codes mode

The revision codes mode is intended to display information about the revision control IDs contained in the different source code modules. This operating mode is selected by the command line option **-r**. The output of this operating mode will look as shown below:

Sample output of the revision codes operating mode

```
(pm@schalke)> ramadelo -r
RAMADeLo - RAM analysis and defect location tool (c) PM 2007
$Id: ramadelo.c,v 1.11 2007/03/16 10:38:39 pm Exp $
$Id: ramplot.c,v 1.3 2007/03/16 10:38:39 pm Exp $
$Id: ramdoc.c,v 1.2 2007/03/16 10:38:39 pm Exp $
$Id: postscript.c,v 1.1 2007/02/26 16:00:36 pm Exp $
$Id: pslogo.c,v 1.2 2007/03/14 16:21:07 pm Exp $
$Id: global.c,v 1.1 2007/02/26 16:00:36 pm Exp $
```

4.2 Command line options for analysis modes

4.2.1 RAM configuration commands

The command line options used to specify the RAM configuration information are initiated by the keyword **-R**, followed by the ID of a parameter and its value. This option is allowed to appear more than one time in the command line in order to specify different RAM parameters.

If the RAM configuration parameters are summarised in a RAM configuration file, this file may by passed to the RAMADeLo tool using the **-C** command line option, followed by the name of the configuration file.

For further details please refer to chapter 2, which contains a full description of the parameters required to describe the RAM configuration and a specification of the configuration file format.

4.2.2 RAM fail mode filter

The evaluation of RAM failures may be restricted to certain fail modes. By default, all types of failures will be evaluated. Using the command line option **-F** this evaluation can be restricted to stuck-at-0 failures (option **-F0**) or stuck-at-1 failures (option **-F1**).

4.2.3 Output format specification

There are three different output formats supported by RAMADeLo. The default output format is plain ASCII and will be written to the standard output if no output file name has been specified. This corresponds to the command line option **-Otxt**.

Alternatively two Postscript formats may be selected for a graphical display of the analysis results. Using the option **-Ops** the output format will be a standard Postscript format which could be sent directly to a Postscript printer or which could be converted to PDF. The option **-Oeps** will create an encapsuled Postscript format which is intended to be included in other documents like StarOffice texts.

If no output file name is specified, the Postscript output format will create an output file called “ramadelo.ps” or “ramadelo.eps”, respectively.

4.2.4 Output file name specification

If the text output of RAMADeLo shall be written directly into a text file or the name of a Postscript output shall be different from the default names, the command line option **-o** can be used. This option requires one argument, which will be used as file name for the analysis results.

Absolute and relative path names are allowed, but not mandatory.

4.3 Detail analysis mode

The detail analysis mode is intended to investigate the failures of one RAM cell. As a consequence only one RAM failure log file has to be supplied to the tool. If more RAM failure log files are supplied, only the first one will be evaluated, all other files will be ignored.

In detailed analysis the pass/fail information is displayed for each bit cell, each row decoder and each R/W amplifier. If the fail mode of a bit cell is non-ambiguous, it will be displayed as stuck-at-0 failure or struck-at-1 failure in the RAM map. If the behaviour of the RAM cell differs between the passes of the 6N RAM test, it will be flagged as failed cell, but with an ambiguous fail mode.

If all cells in one row are found failed, the fail flag of the related row decoder will be set. If these cells additionally show the same fail mode, the row decoder will inherit this mode. If the bit cells within the row show different fail modes, the row decoder will be flagged as failed, but with an ambiguous fail mode.

If all cells in one column are found failed, the fail flag of the related R/W amplifier will be set. If these cells additionally show the same fail mode, the R/W amplifier will inherit this mode. If the bit cells within the column show different fail modes, the R/W amplifier will be flagged as failed, but with an ambiguous fail mode.

The detail analysis mode is selected by the command line option **-D**. This is the default mode of RAMADeLo.

4.3.1 Text output of a detail analysis

The text output of a detail analysis can be divided in three sections: In the first section the general settings of this analysis are listed. Any filters or evaluation limits are shown here. Additionally the some statistics about the number of evaluated files and the numbers of failures are listed in this section.

Sample text output of detail analysis mode

```
(pm@schalke)> ramadelo -D -C14013.cnf -Otxt RAMData/c23132w19x10y10.ram
RAMADeLo - RAM analysis and defect location tool (c) PM 2007

*********************************
* General settings and statistics *
*********************************

Evaluated fail modes                     :        all
Hypertrophic fail margin                 :  unlimited
Logical start address                    :     0x0000
RAM size evaluated                       :     0x0080

RAM result files processed               :          1
RAM result files evaluated               :          1
RAM result files dismissed               :          0
Total count of bit fails                 :          9
Total count of column fails              :          0
Total count of row fails                 :          0
Max. count of bit fails per bit position :          1
Max. count of column fails               :          0
Nax. count of row fails                  :          0
```

In the next section the fail map of the RAM is shown. Each bit cell, each row decoder and each R/W amplifier is represented by one symbol. A dot "." represents a functional cell, "0" and "1" represent cells with non-ambiguous fail modes stuck-at-0 or stuck-at-1, respectively. The symbol "A" finally represents a failing cell with an ambiguous fail mode.

Sample text output of detail analysis mode (cont'd)

```
************************
* RAM defect locations *
************************

        +--------------------------------+---+--------------------------------+
        | C C C C C C C C C C C C C C C C |   | C C C C C C C C C C C C C C C C |
        | O O O O O O O O O O O O O O O O |   | O O O O O O O O O O O O O O O O |
        | L L L L L L L L L L L L L L L L |   | L L L L L L L L L L L L L L L L |
        | 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 |   | 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 |
        | 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 |   | 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 |
        +--------------------------------+---+--------------------------------+
        | . . . . . . . . . . . . . . . . |   | . . . . . . . . . . . . . . . . |
+-------+--------------------------------+---+--------------------------------+
| ROW00 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . 1 . . |
| ROW01 | . . . . . . . . . . . 1 . . . . | . | . . . . . . . . . . . . . . . . |
| ROW02 | . . . . . . . . . . . 1 . . . . | . | . . . . . . . . . . . . . . . . |
| ROW03 | . . . . . . . . . . . 1 . . . . | . | . . . . . . . . . . . . . . . . |
| ROW04 | . . . . . . . . . . . 1 . . . . | . | . . . . . . . . . . . . . . . . |
| ROW05 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW06 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW07 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW08 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW09 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . 1 . . . |
| ROW10 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW11 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW12 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW13 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW14 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW15 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW16 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW17 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW18 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW19 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW20 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW21 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW22 | . . . . . . . . . . . 1 . . . . | . | . . . . . . . . . . . . . . . . |
| ROW23 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . 1 . |
| ROW24 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . 1 . |
| ROW25 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW26 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW27 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW28 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW29 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW30 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
| ROW31 | . . . . . . . . . . . . . . . . | . | . . . . . . . . . . . . . . . . |
+-------+--------------------------------+---+--------------------------------+

********************
* RAM defect list *
********************

+--------+--------+-----------+-----------------------------------------+
| COLUMN |  ROW   | FAIL MODE |               COORDINATES               |
+--------+--------+-----------+-----------------------------------------+
|     11 |      1 | STUCK1    | (   63.60, 6101.40) (   84.50, 6133.70) |
|     11 |      2 | STUCK1    | (   84.50, 6101.40) (  105.40, 6133.70) |
|     11 |      3 | STUCK1    | (  105.40, 6101.40) (  126.30, 6133.70) |
|     11 |      4 | STUCK1    | (  126.30, 6101.40) (  147.20, 6133.70) |
|     11 |     22 | STUCK1    | (  502.50, 6101.40) (  523.40, 6133.70) |
|     27 |      9 | STUCK1    | (  230.80, 6738.70) (  251.70, 6771.00) |
|     29 |      0 | STUCK1    | (   42.70, 6803.30) (   63.60, 6835.60) |
|     29 |     23 | STUCK1    | (  523.40, 6803.30) (  544.30, 6835.60) |
|     29 |     24 | STUCK1    | (  544.30, 6803.30) (  565.20, 6835.60) |
+--------+--------+-----------+-----------------------------------------+
```

The last section contains a list of all detected failures with their column and row indices, the fail mode and the coordinates where the failing cell can be found in the chip layout (lower left corner and upper right corner). These coordinates are related to the origin of the chip layout, the unit is µm. In order to get correct cell coordinates, the RAM configuration has to be supplied. Please refer to chapter 2 for more details.

4.3.2 Postscript output of a detail analysis

The postscript output of a detail analysis contains the same information as the text output, but the fail map and the location of the analysed RAM cell within the chip layout are drawn in graphical format. If the RAM information is supplied correctly, all drawings in this plot will be true to scale.

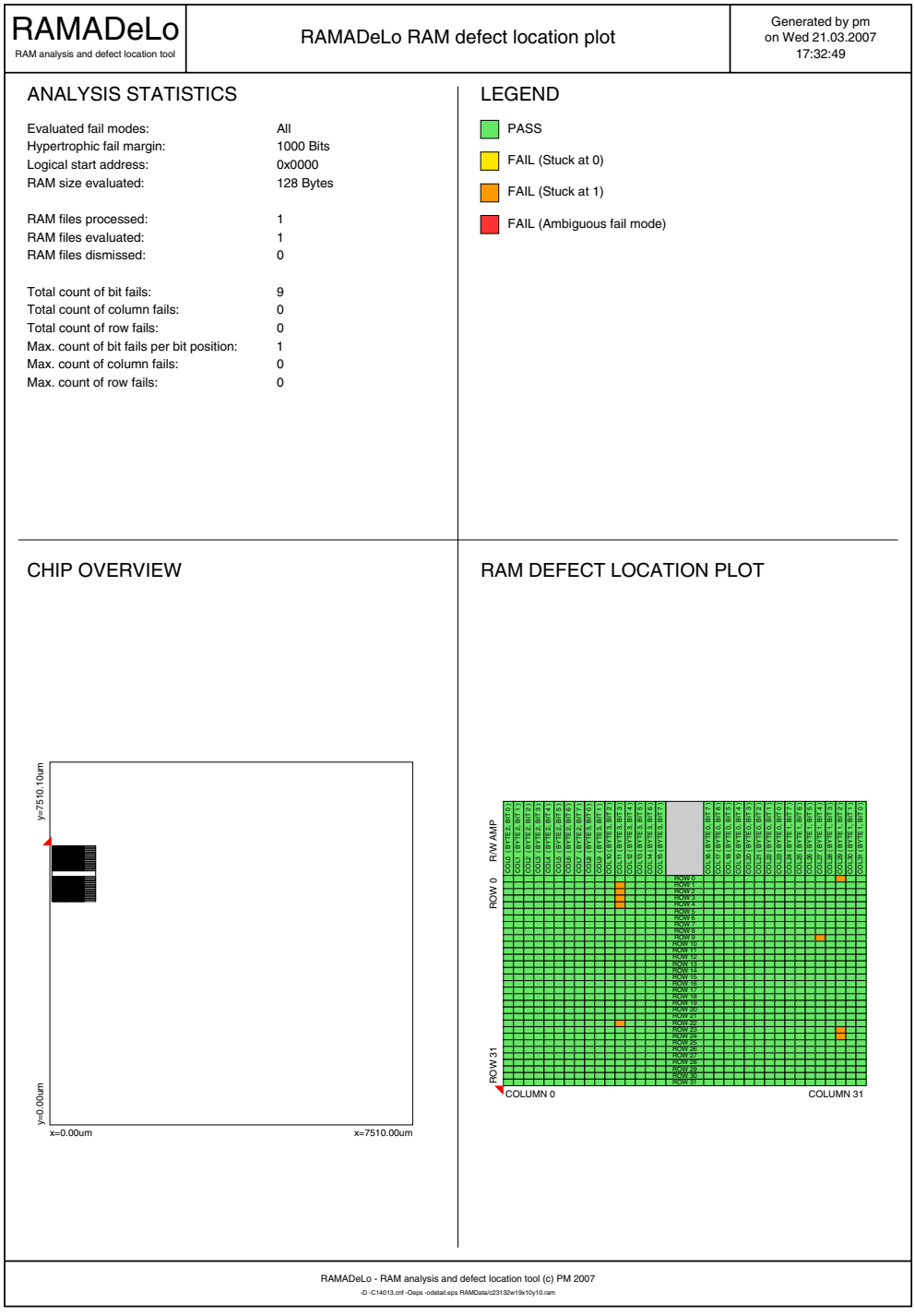


Figure 4-1: Sample plot of a detail analysis

4.4 Summary analysis mode

The summary analysis mode is intended to investigate the failures of numerous RAM cells in order to obtain statistical information of the failure probability. The number of RAM failure log files is not limited by the RAMADeLo tool and may only be bounded by the length of the command line allowed by the operating system.

In summary analysis the failure probability information is displayed for each bit cell, each row decoder and each R/W amplifier. In contrast to to the detail analysis, the different fail modes will not be distinguished here.

If all cells in one row are found failed, the fail flag of the related row decoder will be set. If all cells in one column are found failed, the fail flag of the related R/W amplifier will be set.

The summary analysis mode is selected by the command line option **-S**.

4.4.1 Text output of a summary analysis

The text output of a summary analysis can be divided in two sections: In the first section the general settings of this analysis are listed. Any filters or evaluation limits are shown here. Additionally the some statistics about the number of evaluated files and the numbers of failures are listed in this section.

Sample text output of summary analysis mode

```
(pm@schalke)> ramadelo -S -C14013.cnf -Otxt RAMData/*
RAMADeLo - RAM analysis and defect location tool (c) PM 2007

**********************************
* General settings and statistics *
**********************************

Evaluated fail modes                    :        all
Hypertrophic fail margin                :       1000 Bits
Logical start address                   :     0x0000
RAM size evaluated                      :     0x0080

RAM result files processed              :        122
RAM result files evaluated              :        119
RAM result files dismissed              :          3
Total count of bit fails                :        957
Total count of column fails             :          4
Total count of row fails                :          0
Max. count of bit fails per bit position :          8
Max. count of column fails              :          1
Nax. count of row fails                 :          0
```

In the second section the fail counts of the RAM are shown in tabular format. The format of this section complies to the CSV (comma separated values) standard, so that it can be imported into spreadsheet documents for further evaluation. The table fields are separated from each other by a semicolon. Each bit cell, each row decoder and each R/W amplifier is represented by one table field. The contents of these fields is the total number of fails found in the related RAM element.

The column named “ROWTOT” contains the number of rows which completely failed, the row named “COLTOT” contains the number of columns which completely failed. In the sample output shown below the lines have been truncated after the column “COL10” for better readability. In the real output all columns up to “COL31” will be contained in the table.

Sample text output of summary analysis mode (cont'd)

```
**************************
* RAM defect distribution *
**************************

      ;ROWTOT;  COL0;  COL1;  COL2;  COL3;  COL4;  COL5;  COL6;  COL7;  COL8;  COL9; COL10;   ...
COLTOT;       ;     0;     0;     1;     0;     0;     0;     0;     0;     0;     0;     1;   ...
  ROW0;     0;     0;     2;     1;     0;     0;     0;     0;     0;     0;     2;     1;   ...
  ROW1;     0;     1;     2;     1;     4;     0;     2;     0;     0;     0;     1;     2;   ...
  ROW2;     0;     1;     1;     1;     3;     0;     1;     0;     0;     0;     1;     1;   ...
  ROW3;     0;     1;     3;     1;     6;     0;     1;     0;     0;     0;     2;     1;   ...
  ROW4;     0;     1;     3;     1;     3;     0;     2;     0;     0;     0;     3;     1;   ...
  ROW5;     0;     0;     1;     1;     1;     0;     2;     0;     0;     0;     2;     1;   ...
  ROW6;     0;     0;     1;     1;     1;     0;     2;     0;     0;     0;     0;     1;   ...
  ROW7;     0;     1;     1;     1;     1;     1;     0;     0;     1;     0;     0;     2;   ...
  ROW8;     0;     0;     1;     1;     2;     0;     1;     0;     1;     0;     0;     2;   ...
  ROW9;     0;     0;     3;     1;     0;     0;     1;     0;     1;     0;     1;     2;   ...
 ROW10;     0;     0;     1;     1;     4;     1;     1;     0;     2;     0;     1;     2;   ...
 ROW11;     0;     0;     4;     1;     2;     1;     4;     0;     2;     0;     0;     1;   ...
 ROW12;     0;     0;     1;     1;     2;     0;     4;     0;     2;     0;     0;     1;   ...
 ROW13;     0;     0;     1;     1;     2;     0;     2;     0;     1;     0;     1;     1;   ...
 ROW14;     0;     0;     1;     1;     3;     0;     2;     0;     2;     0;     2;     1;   ...
 ROW15;     0;     1;     1;     1;     2;     0;     2;     1;     1;     0;     0;     1;   ...
 ROW16;     0;     0;     1;     1;     4;     0;     2;     1;     1;     0;     1;     1;   ...
 ROW17;     0;     0;     3;     1;     1;     0;     2;     0;     1;     0;     2;     1;   ...
 ROW18;     0;     0;     3;     1;     0;     0;     1;     0;     0;     0;     2;     1;   ...
 ROW19;     0;     0;     2;     1;     2;     0;     2;     1;     1;     0;     0;     1;   ...
 ROW20;     0;     0;     2;     1;     3;     0;     2;     0;     2;     0;     1;     1;   ...
 ROW21;     0;     0;     2;     1;     1;     0;     1;     0;     1;     1;     0;     1;   ...
 ROW22;     0;     0;     1;     1;     1;     0;     3;     0;     0;     0;     0;     1;   ...
 ROW23;     0;     0;     2;     1;     2;     0;     1;     0;     0;     0;     2;     1;   ...
 ROW24;     0;     0;     1;     1;     1;     0;     4;     0;     0;     0;     1;     1;   ...
 ROW25;     0;     0;     0;     1;     1;     0;     2;     0;     0;     0;     0;     1;   ...
 ROW26;     0;     0;     1;     1;     3;     0;     0;     0;     0;     0;     2;     1;   ...
 ROW27;     0;     0;     1;     1;     4;     0;     0;     0;     0;     0;     1;     1;   ...
 ROW28;     0;     0;     1;     1;     1;     0;     1;     0;     1;     0;     2;     2;   ...
 ROW29;     0;     0;     1;     1;     0;     0;     2;     0;     0;     0;     1;     1;   ...
 ROW30;     0;     0;     1;     1;     1;     0;     0;     0;     1;     0;     0;     1;   ...
 ROW31;     0;     0;     0;     1;     0;     0;     0;     0;     0;     0;     0;     1;   ...
```

4.4.2 Postscript output of a summary analysis

The postscript output of a summary analysis contains the same information as the text output, but the fail map and the location of the analysed RAM block within the chip layout are drawn in graphical format. If the RAM information is supplied correctly, all drawings in this plot will be true to scale.

The failure probabilities for the different RAM elements are displayed as a colour scale. Please note that there are individual scale factors for bit cells, rows and columns. The color scale legend is found in the upper right part of the plot

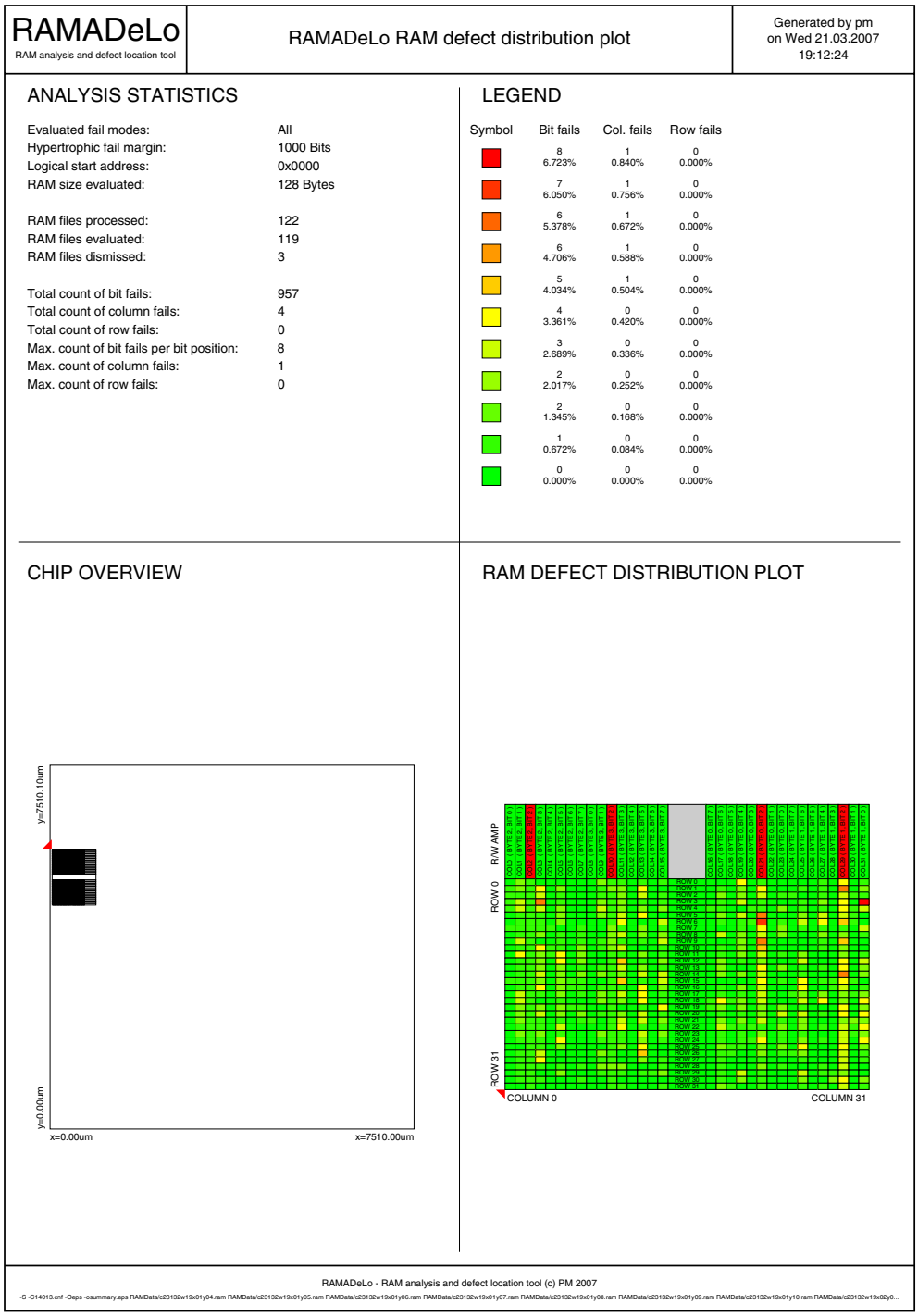


Figure 4-2: Sample plot of a summary analysis

4.5 Command summary, online help

Command summary, online help

```
(pm@schalke)> ramadelo -h
RAMADeLo - RAM analysis and defect location tool (c) PM 2007

NAME
     RAMADeLo - RAM analysis and defect location tool

SYNOPSIS
     ramadelo [options] [ <logfile> ... ]

DESCRIPTION

OPTIONS

     -h
          Display this help

     -r
          Display RCS ID strings of all modules

     -D
          Run detail analysis on one RAM log file.
          If more than one log file is supplied, only the first one will
          be used for analysis. In the detail analysis each defective bit
          cell, column or row, respectively, will be displayed together
          with the individual fail mode (stuck0, stuck1, ambiguous).

     -S
          Run summary analysis on one or more RAM log files.
          In the summary analysis the failure rate for each bit position,
          column or row, respectively, will be calculated with respect
          to the number of RAM log files evaluated. The fail mode
          information will not be available in this summary.

     -F<fmode>
          Apply fail mode filter to RAM log file data.
          <fmode> = 0 : Evaluate only stuck0 failures
          <fmode> = 1 : Evaluate only stuck1 failures

     -R<param>=<val>[,<val>]
          Supply RAM information via command line option.
          The following RAM information parameters are supported:
          CSX = <val> : Chip size X [um]
          CSY = <val> : Chip size Y [um]
          ROX = <val> : RAM cell origin X [um]
          ROY = <val> : RAM cell origin Y [um]
          ROT = <val> : RAM cell orientation, where
            <val> = +X or 0   : RAM cell rotated 0 deg. cw
            <val> = -Y or 90  : RAM cell rotated 90 deg. cw
            <val> = -X or 180 : RAM cell rotated 180 deg. cw
            <val> = +Y or 270 : RAM cell rotated 270 deg. cw
          OFX = <val> : X offset of first bit cell [um]
          OFY = <val> : Y offset of first bit cell [um]
          RPX = <val> : Bit cell size X [um]
          PRY = <val> : Bit cell size Y [um]
          WSX = <val> : Word decoder size X [um]
          ASY = <val> : R/W amplifier size Y [um]
          COL = <pcol>,<lcol> : RAM column mapping, where
            <pcol> = Physical column number [0..31]
            <lcol> = Logical column number [0..31]
          ESA = <val> : Start address to be evaluated [dec/hex]
          ESZ = <val> : RAM size to be evaluated [dec/hex]
          HYP = <val> : Hypertrophic fail margin [dec/hex]
          The default values for these parameters correspond
          to the L08 standard RAM cell. Evaluation starts at
          address 0x0000 and comprises max. 256 Bytes RAM size.
```

Command summary, online help (cont'd)

```
-C<confile>
     Supply RAM information info via config file.
     The set of parameters supported in the config file is
     identical to those described for the -R option.
     Additionally empty lines or comments are allowed in the
     config file. Comment lines start with a semicolon.

-O<format>
     Specify output format for evaulations.
     <format> = txt : ASCII text format (default)
     <format> = ps  : Postcript format
     <format> = eps : Encapsuled Postcript format

-o<outname>
     Specify filename for RAMADeLo output data.
     By default, RAMADeLo will write text output data to
     the standard output, which may be redirected into a file
     by the user. As an alternative this option -o may be be
     used to write text output data into an ASCII file with the
     given name.
     If the Postscript output format is used, RAMADeLo will always
     create output files, which are called "ramadelo.ps" or
     "ramadelo.eps", respectively, depending of the selected
     Postscript style. Here the -o option can be applied to
     supersede these default names by the given file name.
```

5 Tables of contents

5.1 Table of contents

5.2 Table of figures