Proposal for Scan Failure Datalog Format

Ajay Khoche
Verigy Ltd
Santa Clara, USA

1 Introduction

This document describes a proposal to store scan failure datalog in Standard Test Data Format (STDF) using General Data Record (GDR) specifications. The intent of this specification is to facilitate the data transfer from ATE to diagnosis tools to enable volume diagnostics.

2 Requirements

- To provide means to specify all the data required by the diagnosis tool to perform diagnosis
- To provide means to synchronize and validate the fail data
- To minimize the size of the datalog

3 Volume Diagnosis Requirements

This section describes the information that needs to be communicated from ATE to the diagnosis tool. The information can be classified in six categories. The first three categories identify the die, test and the environment for which the data is collected whereas the last three contain the details of captured information.

3.1 Die/Core Identification

This set of information identifies the die/core for which the data is collected. The information includes following elements:

- Wafer lot identifier
- Wafer identifier
- X-Y coordinates for die identification
- Code-id (In case of core based scan data logging

3.2 Test Identification

This set of information identifies the test for which the data collection is done. It consists of:

- Test program version: This is used to keep track of test program revisions
- Test mode: This is used to hold the insertion identification e.g. Wafer sort 1.
- Test num (assuming unique test # for each scan test)
- Pattern set name (any ATPG given name)
- Test suite num (assuming unique test # for each scan test)

3.3 Environment Specification

This information set contains the data about the environment in which a particular test was run. Each test could be run under different environment conditions at different point in the test flow. This information consists of:

- Scan Frequency
- Capture Frequency
- Test Nominal Voltage
- Test Stress Voltage: To be used if any voltage stress in terms of bump or low voltage is applied
- Test temperature
- User specific environment specification

3.4 Datalog Format Specification

This information allows the specification of the format in which the fail data is provided. It consists of following information:

- Fail data format: ATPG (pattern based) vs. ATE (Cycle Based)
- Expected Data flag: (No expected data | with expected data)
- Z Handling flag: This is used to indicate how the Z states are handled and represented in the data log

3.5 Data Capture Information

This set of data is provided to perform data validation and synchronization by the diagnosis tools to ensure that the there is not discrepancy in the data consistency, which may result in useless output from the diagnosis tool. The data in this class consists of:

- First Pattern executed (Required only if fail data format is ATPG and first pattern is other than 0)
- Last Pattern executed (Required only if fail data format is ATPG and the last executed pattern is other than the last pattern in the memory)
- First cycle executed (Required only if fail data format is ATE)
- Last cycle executed (Required only if fail data format is ATE and the last executed cycle is other than the last cycle with entire pattern set)
- Total number of fails
- Total patterns (required only if the fail data format is ATPG.)
- Total cycles (required only if the fail data format is ATE.)
- Pattern 0 fail indication: This is set if there is any fail on any pin in pattern 0
- Buffer full identification: This information will identify for each pin whether there was a buffer overrun.

3.6 Fail data

This is the actual fail data during the test run. It is provide in the format specified by the datalog format specification. The possible formats are:

- ATPG : (pattern_num, offset, pin_num, measured_data, expected data)
- ATPG : (pattern_num, offset, pin_num, measured_data)
- ATE: (cycle_num, pin_num, measured_data, expected data)
- ATE: (cycle_num, pin_num, measured_data)

Here it is assumed that the pin_num is the PMR_INDX that can be obtained from the Pin Map Record (PMR).

In addition, In the case of no expected data being written, the expectation is that for a given setup the expected data will be written out in a special one time only flow for the Diagnosis tool. For that case either of the ATPG or ATE format with expected can be used. If however the consistency in the cycle numbering between ATE and the diagnosis tools is guaranteed by some other means, then the special run can omit the cycle number from the log and a flag can be added in the record to indicate that. Similarly if the consistency in the pin ordering can be achieved through other means then pin number can also be dropped from the log.

4 Data Representation

Table 1 shows the data representation for the information described in the previous section for the Diagnosis GDR record.

Table 1

Information	STDF Field Name	Data Type	Required	Missing Invalid value	Comment
Bytes following the header	REC_LEN	U*2	Y		This will depend on the data to be represented
Record type	REC_TYP	U*1	Y		50
Record Subtype	REC_SUB	U*1	Y		10
Number of Gen_data filed in the record	FLD_CNT	U*2	Y		
Diagnostics Record ID	GEN_DATA(10)	C*14	Y		"DIAGNOSIS_DATA": Field to identify the diagnostics record in the set of GDRs
Wafer Lot Id	GEN_DATA (10)	C*n	Y		n is the length of the ID string
Wafer Id	GEN_DATA (2)	U*2	Y		Assumed to be a number
X-Coord	GEN_DATA (2)	U*2	Y		
Y-Coord	GEN_DATA (2)	U*2	Y		
Core Id	GEN_DATA (2)	U*2	N	0	Assumed to be a number
Test Program ID	GEN_DATA(10)	C*n	N	0	A text field for the test program version identification. A 0 in the n field indicates missing/unspecified field.
Test Mode	GEN_DATA(10)	C*n	N	0	A text filed is for test insertion identification. A 0 in the n field indicates missing/unspecified field
Test Num	GEN_DATA (2)	U*2	Y		Assumed to be a number
Pattern Id	GEN_DATA (10)	C*n	N	0	A 0 in the n filed indicates a missing field
Test Suite Id	GEN_DATA (2)	U*2	Y		Assumed to be a number
Scan Freq	GEN_DATA (3)	U*4	Y	0	
Capture Freq	GEN_DATA (3)	U*4	N	0	
Test Nominal Voltage	GEN_DATA (2)	U*2	Y		

Test Stress Voltage	GEN_DATA (2)	U*2	N	0	
Test Temp	GEN_DATA (5)	I*2	N	32767	To be used if either self-contained GDR is needed or if there is a temperature that is different than the ambient that needs to be captured
User Env Spec	GEN_DATA (10)	C*n	N	0	To specify any custom specification (string type)
ATPG/ATE Format	GEN_DATA (1)	U*1	Y		0: ATPG, 1: ATE
Expected Data Spec	GEN_DATA (1)	U*1	Y		A single field is used to indicate the presence of expected data in the fail records. 1- present, 0 - absent
Z-Handling flag	GEN_DATA(1)	U*1	N		Used to indicate Mapping to Z – states 0:L, 1:H, 2:Z, 3:X, 4-not handled
First Pattern Executed	GEN_DATA (3)	U*4	Y/N	0	Mandatory if format is ATPG and the first pattern executed is not the first pattern of the in the pattern memory
Last Pattern Executed	GEN_DATA (3)	U*4	Y/N		Mandatory if format is ATPG and the last pattern executed is not the last pattern in the pattern memory. If the format is not ATPG then this field will not be in the record
First Cycle executed	GEN_DATA (3)	U*4	Y/N	0	Mandatory if format is ATE and the first cycle is not the fist cycle of the first pattern
Last Cycle executed	GEN_DATA (3)	U*4	Y/N		Mandatory if format is ATE and the last cycle executed is not the last cycle with entire pattern run. If the format is not ATE then this field will not be in the record.
Total Fails	GEN_DATA (3)	U*4	Y		
Total Cycles	GEN_DATA (3)	U*4	N		Used only if ATE format is used
Total Patterns	GEN_DATA (3)	U*4	N		Used only if ATPG format is used
Patten 0 fails	GEN_DATA (1)	U*1	Y		A global field for all the scan out pins
Buffer Full	GEN_DATA (10)	C*n	Y		One flag per scan out pins
Pattern Num	GEN_DATA (3)	D*4	Y/N		Not required if ATE based format is used
Offset	GEN_DATA (3)	D*4	Y/N		Not required if ATE based format is used
Pin Num	GEN_DATA (2)	D*2	Y		Assumes that pin name to pin number mapping is done is some other record if not it can be added to this record itself.
Measured + Expected data	GEN_DATA (1)	D*1	Y/N		Measured data in the higher nibble and expected data in lower nibble. The lower nibble is not required if Expected is not supplied

Cycle Num	GEN_DATA (3)	D*4	Y/N		Not required if ATPG format is used

In the above table D*n is intended to be used with GEN_DATA type 12(D*n). Even though in the original document it is used to encode one bit in each following byte, in our case all the bits will be meaningful. Thus in the case of ATE format with expected data each fail data would be of [4 (cycle num) + 2 (pin num) + 1 (measured+expected) = 7] bytes. Therefore if there were say ten failures then the fail data would take [1 (GEN_DATA 12) + 2 (for indicating 70 bytes to follow) + 70 (for fail data) = 73] bytes. Please note that there is no GEN_DATA type indicator is used for the sub-fields of each data.

5 Diagnostic Record Syntax

This section describes the information is organized in the Diagnosis GDR record using BNF notation.

// Diagnostic Record Specification

Diagnostic_record := <recorder_header><Record_body>

<record_header> := <REC_LEN>
<REC_TYP>
<REC_SUB>;
<REC_LEN> :=U*2;
<REC_TYPE> :=U*1;
<REC_SUB> :=U*1;

<record_body><FLD_CNT>
<Diag_record_id>
<die_core_info>
<Test_info>
<env_info>
<Data_capture_run_info>
<datalog_format_info>
<data_log>;

<FLD_CNT> := U*2;
<Diag_record_id> : =<Gen_data_C*n><Char_arr[n]>; //n is the length of the id tag

// Die/core Identification Specification

<die_core_info> := <Wafer_lot_id><wafer_id><Xcoord><Ycoord><CoreId>;
<Wafer_lot_id> := <Gen_data_C*n><char_arr>;
<wafer_id> := <Gen_data_U*2><Two_bytes>;
<Xcoord> := <Gen_data_U*2><Two_bytes>;
<Ycoord> := <Gen_data_U*2><Two_bytes>;
<CoreId> := <Gen_data_U*2><Two_bytes>;

// Test Identification Specification

<Test_info> : =
<Test_pgm_ver>
<Test_mode>
<Test_num>
<Pattern_Id>
<Test_suite_Id>;

<Test_pgm_ver> := <Gen_data_C*n>><Char_arr[n]>;
<Test_mode> := <Gen_data_C*n>><Char_arr[n]>;
<Test_num> := <Gen_data_U*2><Two_bytes>;
<Pattern_Id> : = <Gen_data_C*n><Char_arr[n]>;
<Test_suite_Id> := <Gen_data_U*2><Two_bytes>;
//Environment Specification
<env_info> := <Scan_freq>
<Capture_freq>
<Nominal_Test_voltage>
<Stress_test_voltage>
<Test_temp>
<User_definded_spec>;
<Scan_freq> := <Gen_data_U*4><Four_bytes>; // in KHz
<Capture_freq> := <Gen_data_U*4><Four_bytes>; // In KHz
<Nominal_Test_voltage> :=<Gen_data_U*2><Two_bytes> ; //in mV
<Stress_test_voltage> := <Gen_data_U*2><Two_bytes> ; //in mV
<Test_temp> := <Gen_data_I*2><Two_bytes>; // In Celsius
<User_defined_spec> := <Gen_data_C*n><Char_arr[n]>;

//Data log Format Specification
<datalog_format_info> := <Fail_format_spec>
<Expected_data_spec>
<Z-handling_spec>;
<Fail_format_spec> := [<ATPG> | <ATE>];
<ATPG> := <Gen_data_U*1>00; //U*1
<ATE> := <Gen_data_U*1> 01; //U*1

<Expected_data_spec> := [<Exp>|<NoExp>];
<Exp> := <Gen_data_U*1>01; //U*1
<NoExp> := <Gen_data_U*1>00; //U*1

<Z-handling_spec> := [<Z-as-Z> | <Z-as-0 >| <Z-as-1>| <Z-as-X>|<Z-not-Handled>];
<Z-as-Z> := <Gen_data_U*1>02 ;//U*1
<Z-as-0> := <Gen_data_U*1>00 ;//U*1
<Z-as-1> := <Gen_data_U*1>01 ;//U*1
<Z-as-X> := <Gen_data_U*1>03;//U*1
<Z-not_handled> := <Gen_data_U*1>04;//U*1
//Data Capture Run Information Specification
<Data_capture_run_info> := <First_pat_exec>

<Last_pat_exec>
<First_cyc_exec>
<Last_cyc_exec>
<Total_fails>
<Total_cycles>
<Total_patterns>
<Pattern_0_fail>
<buffer_full>;
<First_pat_exec> := <Gen_data_U*4><Four_bytes>; // Used only if format is ATPG
<Last_pat_exec> := <Gen_data_U*4><Two_bytes>;// Used only if format is ATPG
<First_cyc_exec> := <Gen_data_U*4><Four_bytes>; // Used only if format is ATE
<Last_cyc_exec> := <Gen_data_U*4><Four_bytes>;// Used only if format is ATE
<Total_fails> := <Gen_data_U*4><Four_bytes>;
<Total_patterns> := <Gen_data_U*4><Four_bytes> ;// Used only if format is ATPG
<Total_cycles> := <Gen_data_U*4><Four_bytes> // Used only if format is ATE
<Pattern_0_fail> := <Gen_data_U*1><One_byte>;
<buffer_full> := <Gen_data_C*n><char_arr[n]>; //n is the number of outputs that were logged.

// Fail Data Specification

<data_log> := <Gen_data_D*n><Fail_data_size><[<Log_element>]+ ;
<fail_data_size> := <two_byte>;
<Log_element> := <Atpg_Exp>|<Atpg_no_Exp>|<ATE_Exp>|<ATE_no_Exp>;
<Atgpg_Exp> := <Pat_num><Offset><Pin_num><Meas_data><Exp_data>;
<Atgpg_No_Exp> := <Pat_num><Offset><Pin_num><Meas_data>;
<ATE_Exp> := <Cycle_num><Pin_num><Meas_data> <Exp_data>;
<ATE_Exp> := <Cycle_num><Pin_num><Meas_data>;
<Cycle_num>:= U*4;
<Pat_num> := U*4;
<Offset> := U*4;
<Meas_data> := N*1; // 1 nibble
<Exp_data> := N*1; // 1 nibble

//Terminal specs: The definitions of U*4,U*2,U*1,C*n can be found the STDF V4

<Gen_data_U*1> := 01;
<Gen_data_U*2> := 02;
<Gen_data_U*4> := 03;
<Gen_data_I*2> := 05;
<Gen_data_C*n> := 0a;
<Gen_data_D*n> := 0c;
<Two_byte> := U*2;
<Four_byte> := U*4;
<One_byte> := U*1;
<Char_arr[n]> := C*n;

6 Examples

This section shows some examples of how the proposed GDR record can be used to represent diagnosis information.

6.1 Example 1

This example consists of a simple design with 3 scan chains with each scan chain being 100 FF deep. A total of 10 (0:9) scan patterns are applied in the test. The other parameters are shown in Figure 1.

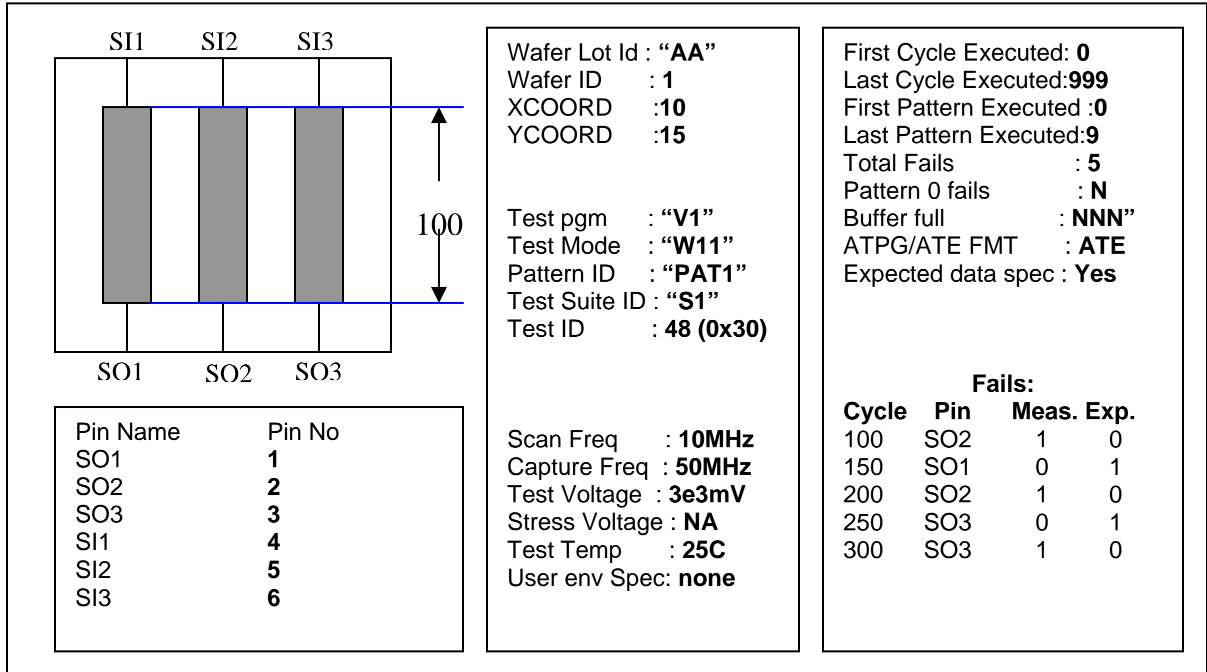

Figure 1: Example with ATE format for failures

The following table shows the GDR record contents for the example in Figure 1.

Line no	Even Byte	Odd Byte	Comment
1	00	9f	159 bytes after header
2	32	0a	Type 50 subtype 10
3	00	2a	42 GEN_DATA fields in the record
4	0a	0e	C*14
5	44	49	"DIAGNOSIS_DATA" string
6	41	48	
7	54	55	
8	59	49	
9	59	5f	
10	44	41	
11	54	41	
12	0a	02	C*2
13	41	41	Wafer lot ID "AA"
14	00*	02	Pad byte (0) and U*2
15	00	01	Wafer ID 01
16	00*	02	Pad byte (0) and U*2
17	00	0a	XCOORD = 10

18	00*	02	Pad byte (0) and U*2
19	00	0f	YCOORD = 15
20	01	00	U*1; Core 0
21	0a	02	C*2
22	56	31	Test Program Version ID “V1”
23	0a	03	C*3
24	57	31	Test Mode ID “W11”
25	31	02	U*2 in the second byte
26	00	30	Test ID = 48 (0x30)
27	0a	04	C*4
28	56	41	Pattern ID = “PAT1” in the four bytes
29	60	31	
30	0a	02	C*2
31	53	31	Test Suite ID = “S1”
32	00*	03	Pad (0) U*4
33	00	00	Scan Freq in four bytes in KHz
34	27	10	(10000 KHz – 00:00:27:10)
35	00*	03	Pad (0) U*4
36	00	00	Capture Frequency in KHz
37	c3	50	50000KHz – (00:00:c3:50)
38	00*	02	Pad (0) U*2
39	0B	B8	Nominal voltage (3000 mV)
40	00*	02	Pad (0) U*2
41	FF	FF	Stress Voltage
42	00*	02	
43	00	19	Test Temperature +25C (00:19)
44	0a	00	No User Env Spec
45	01	01	U*1; ATE format (01)
46	01	01	U*1; Expected data in the log
47	01	04	Z not handled
48	00*	03	Pad (0) U*4
49	00	00	First Cycle 0 (00:00:00:00)
50	00	00	
51	00*	03	Pad (0) U*4
52	00	00	Last Cycle 999 (00:00:03:E7)
53	03	E7	
54	00*	03	Pad(0); U*4
55	00	00	
56	00	05	5 Total fails
57	00*	03	Pad(0) U*4
58	00	00	
59	03	E8	Total cycles 1000(00:00:03:E8)
60	01	00	U*1;No Pattern 0 fail
61	0a	03	C*3
62	54	54	Buffer full “NNN” one for each scan out

63	54	0c	Expected data for fails in the output; D*n
64	00	23	Overusing the 0c format total 35 bytes for 5 failure
65	00	00	First Fail Info
66	00	64	Cycle no 100 (00:00:00:64)
67	00	02	Pin no 2(00:02)
68	10	00	Measured value 1 Exp value 0; Second Fail Info starting at Odd Byte
69	00	00	Cycle no 150(00:00:00:96)
70	96	00	
71	01	01	Pin 1(00:01), measured value 0 and expected value 1
72	00	00	3rd Fail Info on Starting on Even Byte
73	00	C8	Cycle number 200 (00:00:00:C8)
74	00	02	Pin 02 (00:02)
75	10	00	Measured/Expect value in third fail 1/0; 4th Fail Info Starting on Odd byte
76	00	00	
77	FA	00	Cycle number 250 (00:00:00:FA);
78	03	01	Pin 3 (00:03); Measured/expected value in 4th fail 0/1
79	00	00	Fifth Fail Starting on Even byte
80	01	2C	Cycle Number 300 (00:00:01:2C)
81	00	03	Pin Number 3 (00:03)
82	10		Measured/Expected value in 5th fail 1/0
83			

6.2 Example 2

This example is similar to the example in previous section except that the fail data is stored in the ATPG format. The fields that different are highlighted in blue.

SI1 SI2 SI3

100

SO1 SO2 SO3

Pin Name	Pin No
SO1	**1**
SO2	**2**
SO3	**3**
SI1	**4**
SI2	**5**
SI3	**6**

Wafer Lot Id : **"AA"**
Wafer ID : **1**
XCOORD :**10**
YCOORD :**15**

Test pgm : **"V1"**
Test Mode : **"W11"**
Pattern ID : **"PAT1"**
Test Suite ID : **"S1"**
Test ID : **48 (0x30)**

Scan Freq : **10MHz**
Capture Freq : **50MHz**
Test Voltage : **3e3mV**
Stress Voltage : **NA**
Test Temp : **25C**
User Env. Spec**: None**

First Cycle Executed: **0**
Last Cycle Executed:**999**
First Pattern Executed :**0**
Last Pattern Executed:**9**
Total Fails : **5**
Pattern 0 fails : **N**
Buffer full : **NNN"**
ATPG/ATE FMT : **ATE**
Expected data spec : **Yes**

Fails:

Cycle	**Pin**	**Meas.**	**Exp.**
100	SO2	1	0
150	SO1	0	1
200	SO2	1	0
250	SO3	0	1
300	SO3	1	0

Figure 2: Example with Fail Information in ATPG format

Line No	Even Byte	Odd Byte	Comment
1	00	bd	179 bytes following head
2	32	0a	Type 50 subtype 10
3	00	25	37 GEN_DATA fields
4	0a	0e	C*14
5	44	49	"DIAGNOSIS_DATA" string
6	41	48	
7	54	55	
8	59	49	
9	59	5f	
10	44	41	
11	54	41	
12	0a	02	C*2
13	41	41	"AA"
14	00*	02	Pad byte (0) and U*2
15	00	01	Wafer ID 01
16	00*	02	Pad byte (0) and U*2
17	00	0a	XCOORD = 10
18	00*	02	Pad byte (0) and U*2
19	00	0f	YCOORD = 15
20	01	00	U*1; Core 0
21	00*	02	Pad byte (0) and U*2
22	0a	02	C*2
23	56	31	"V1"
24	0a	03	C*3
25	57	31	"W11"
26	31	30	Test ID = 48 (0x30)
27	0a	04	C*4
28	56	41	Pattern ID = "PAT1" in the four bytes
29	60	31	
30	0a	02	C*2
31	59	31	Test Suite ID = "S1"
32	00*	03	Pad byte (0) U*4
33	00	00	Scan Freq in four bytes in KHz
34	27	10	(10000 KHz – 00:00:27:10)
35	00*	03	Pad byte (0) U*4
36	00	00	Capture Frequency in KHz
37	c3	50	50000KHz – (00:00:c3:50)

38	00*	02	Pad byte (0) U*2
39	0B	B8	Nominal voltage (3000 mV)
40	00*	02	Pad byte (0) U*2
41	FF	FF	Stress Voltage
42	00*	02	
43	00	19	Test Temperature +25C (00:19)
44	0a	00	No User Env Specs
45	01	00	U*1; ATPG Format
46	01	01	U*1; Expected data in the log
47	01	04	No Z-handling
48	00*	03	Pad byte(0) U*4
49	00	00	First pattern number 0 (00:00:00:00)
50	00	00	
51	00*	03	Pad byte (0) U*4
52	00	00	Last pattern 9 (00:00:00:09)
53	00	09	
54	00*	03	Pad byte (0); U*4
55	00	00	5 Total fails
56	00	05	
57	00*	03	Pad(0), U*4
58	00	00	
59	00	0a	Total Patterns 10 (00:00:00:0a)
60	01	00	U*1;No Pattern 0 fail
61	0a	03	C*3
62	54	54	Buffer full “NNN” one for each scan out
63	54	0c	Expected data for fails in the output; D*n
64	00	37	Overusing the 0c format; total 55 bytes for 5 failure
65	00	00	First Fail Info
66	00	01	Pattern no 1 (00:00:00:01)
67	00	00	Offset 0
68	00	00	
69	00	02	Pin no 2(00:02)
70	10	00	Meas/Exp value 1/0; Second Fail Info starting at Odd Byte
71	00	00	
72	01	00	Pattern 1 (00:00:00:01)
73	00	00	
74	32	00	Offset 50 (00:00:00:32)
75	01	01	Pin 1; Meas/Expected value 0/1
76	00	00	Third fail start
77	00	02	Pattern 2 (00:00:00:02)
78	00	00	
79	00	00	Offset 0 (00:00:00:01)
80	00	02	Pin 2 (00:02)

81	10	00	Meas./Expected Value 1/0; fourth fail start
82	00	00	
83	02	00	Pattern 2 (00:00:00:02);
84	00	00	
85	32	00	Offset 50 (00:00:00:32)
86	03	01	Pin 3(00:03); Meas/expected value 0/1
87	00	00	Fifth fail start
88	00	03	Pattern 3 (00:00:00:03)
89	00	00	
90	00	00	Offset 0 (00:00:00:00)
91	00	03	Pin 3 (00:03)
92	10		Meas/Expected value 1/0

6.3 Example 3

This example is similar to the example1 in section 5.1 except that no expected data is stored in the record. The fields that are different from the example 1 are highlighted in blue.

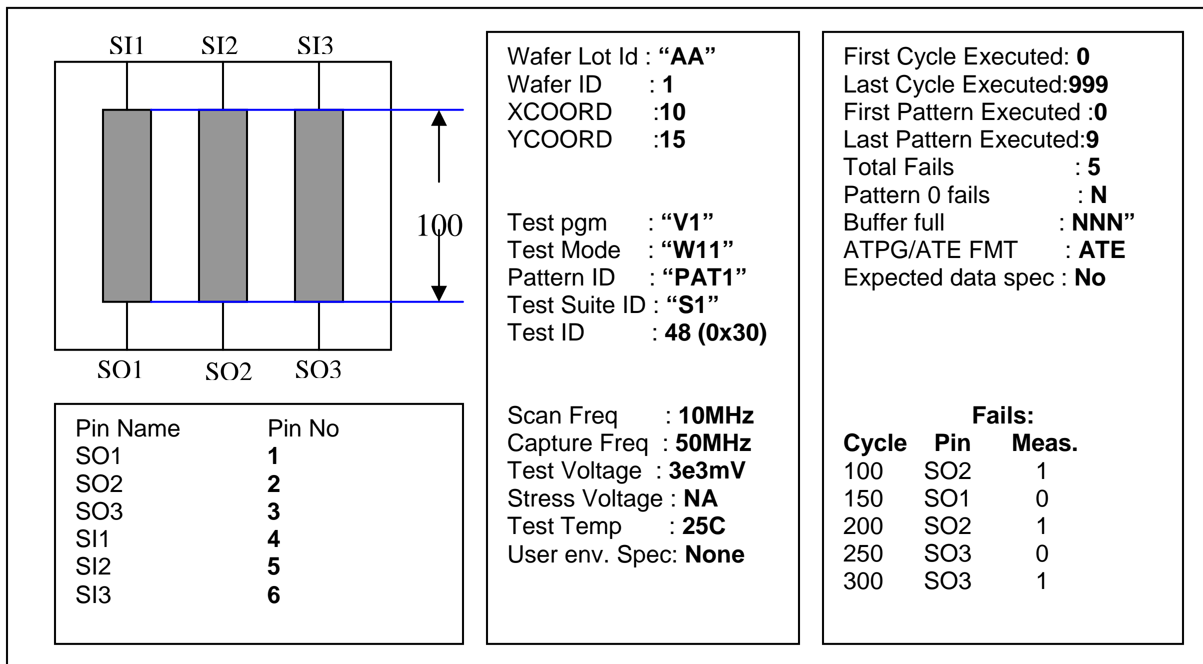

Figure 3: Example with ATE format and no expected data

Line no	Even Byte	Odd Byte	Comment
1	00	9f	159 bytes after header
2	32	0a	Type 50 subtype 10
3	00	2a	42 GEN_DATA fields in the record
4	0a	0e	C*14

5	44	49	“DIAGNOSIS_DATA” string
6	41	48	
7	54	55	
8	59	49	
9	59	5f	
10	44	41	
11	54	41	
12	0a	02	C*2
13	41	41	“AA”
14	00*	02	Pad byte (0) and U*2
15	00	01	Wafer ID 01
16	00*	02	Pad byte (0) and U*2
17	00	0a	XCOORD = 10
18	00*	02	Pad byte (0) and U*2
19	00	0f	YCOORD = 15
20	01	00	U*1; Core 0
21	0a	02	C*2
22	56	31	“V1”
23	0a	03	C*3
24	57	31	“W11”
25	11	02	“1” and U*2
26	31	30	Test ID = 48 (0x30)
27	0a	04	C*4
28	56	41	Pattern ID = “PAT1” in the four bytes
29	60	31	
30	0a	02	C*2
31	53	31	Test Suite ID = “S1”
32	00*	03	Pad (0) U*4
33	00	00	Scan Freq in four bytes in KHz
34	27	10	(10000 KHz – 00:00:27:10)
35	00*	03	Pad (0) U*4
36	00	00	Capture Frequency in KHz
37	c3	50	50000KHz – (00:00:c3:50)
38	00*	02	Pad (0) U*2
39	0B	B8	Nominal voltage (3000 mV)
40	00*	02	Pad (0) U*2
41	FF	FF	Stress Voltage
42	00*	02	
43	00	19	Test Temperature +25C (00:19)
44	0a	00	No User Env Spec
45	01	01	U*1; ATE format (01)
46	01	00	U*1; Expected data in the log
47	01	04	Z not handled
48	00*	03	Pad (0) U*4
49	00	00	First Cycle 0 (00:00:00:00)

50	00	00	
51	00*	03	Pad (0) U*4
52	00	00	Last Cycle 999 (00:00:03:E7)
53	03	E7	
54	00*	03	Pad(0); U*4
55	00	00	
56	00	05	5 Total fails
57	00*	03	Pad(0) U*4
58	00	00	
59	03	E8	Total cycles 1000(00:00:03:E8)
60	01	00	U*1;No Pattern 0 fail
61	0a	03	C*3
62	54	54	Buffer full "NNN" one for each scan out
63	54	0c	No Expected data for fails in the output; D*n
64	00	23	Overusing the 0c format total 35 bytes for 5 failure
65	00	00	First Fail Info
66	00	64	Cycle no 100 (00:00:00:64)
67	00	02	Pin no 2(00:02)
68	10	00	Meas. Data 1; no exp data; Second fail start
69	00	00	Second Fail
70	96	00	At Cycle no 150(00:00:00:96)
71	01	00	Pin 1(00:01); Meas. Data 0.
72	00	00	3rd Fail Info on Starting
73	00	C8	Cycle number 200 (00:00:00:C8)
74	00	02	Pin 02 (00:02)
75	10	00	Meas. Data 1. 4th Fail start
76	00	00	Cycle number 250 (00:00:00:FA);
77	FA	00	Pin 3 (00:03);
78	03	10	Meas. Data 10.
79	00	00	
80	01	2C	Fifth Fail Cycle Number 300 (00:00:01:2C)
81	00	03	Pin Number 3 (00:03)
82	10		Meas. Data 1

6.4 Example 4

This example is similar to example 2 in section 5.2 except that no expected information is stored. The sections that are different are once again highlighted in blue.

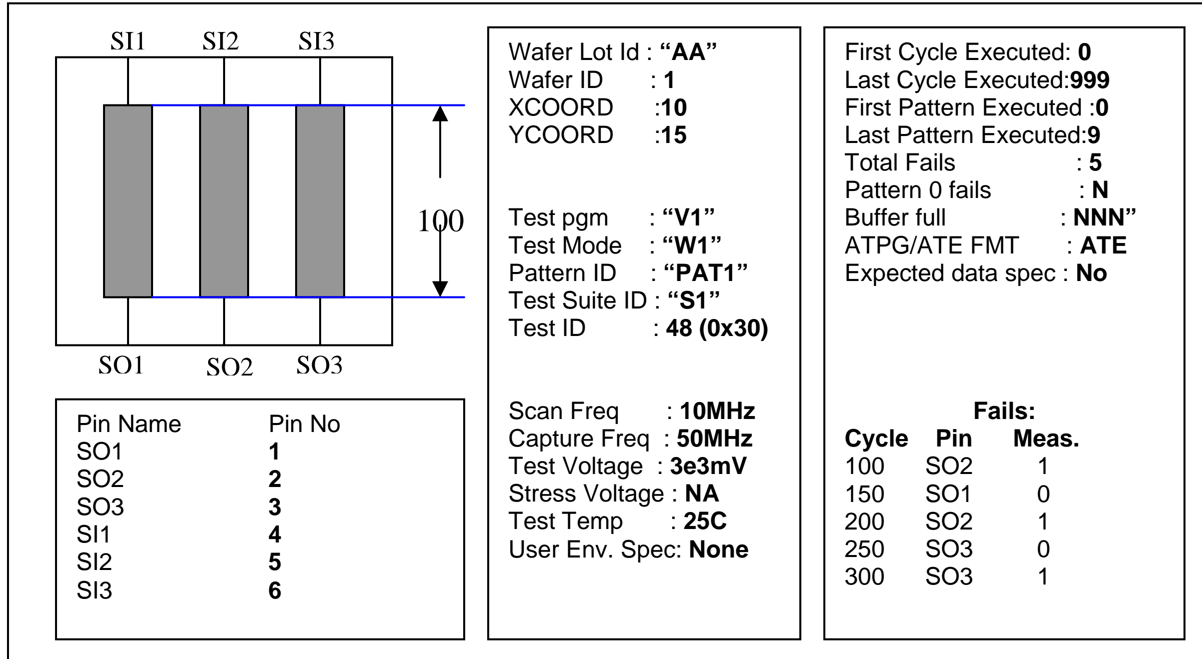

Figure 4: Example 4 with ATPG format and no expected data

Line No	Even Byte	Odd Byte	Comment
1	00	b3	179 bytes following head
2	32	0a	Type 50 subtype 10
3	00	25	42 GEN_DATA fields
4	0a	0e	C*14
5	44	49	"DIAGNOSIS_DATA" string
6	41	48	
7	54	55	
8	59	49	
9	59	5f	
10	44	41	
11	54	41	
12	0a	02	C*2
13	41	41	"AA"
14	00*	02	Pad byte (0) and U*2
15	00	01	Wafer ID 01
16	00*	02	Pad byte (0) and U*2
17	00	0a	XCOORD = 10

18	00*	02	Pad byte (0) and U*2
19	00	0f	YCOORD = 15
20	01	00	U*1; Core 0
21	00*	02	Pad byte (0) and U*2
22	0a	02	C*2
23	56	31	"V1"
24	0a	03	C*3
25	57	31	"W11"
26	31	30	Test ID = 48 (0x30)
27	0a	04	C*4
28	56	41	Pattern ID = "PAT1" in the four bytes
29	60	31	
30	0a	02	C*2
31	59	31	Test Suite ID = "S1"
32	00*	03	Pad byte (0) U*4
33	00	00	Scan Freq in four bytes in KHz
34	27	10	(10000 KHz – 00:00:27:10)
35	00*	03	Pad byte (0) U*4
36	00	00	Capture Frequency in KHz
37	c3	50	50000KHz – (00:00:c3:50)
38	00*	02	Pad byte (0) U*2
39	0B	B8	Nominal voltage (3000 mV)
40	00*	02	Pad byte (0) U*2
41	FF	FF	Stress Voltage
42	00*	02	
43	00	19	Test Temperature +25C (00:19)
44	0a	00	No User Env Specs
45	01	00	U*1; ATPG Format
46	01	00	U*1; No Expected data in the log
47	01	04	No Z-handling
48	00*	03	Pad byte(0) U*4
49	00	00	First pattern number 0 (00:00:00:00)
50	00	00	
51	00*	03	Pad byte (0) U*4
52	00	00	Last pattern 9 (00:00:00:09)
53	00	09	
54	00*	03	Pad byte (0); U*4
55	00	00	5 Total fails
56	00	05	
57	00*	03	Pad(0), U*4
58	00	00	
59	00	0a	Total Patterns 10 (00:00:00:0a)
60	01	00	U*1;No Pattern 0 fail
61	0a	03	C*3
62	54	54	Buffer full "NNN" one for each scan out

63	54	0c	Expected data for fails in the output; D*n
64	00	32	Overusing the 0c format; total 50 bytes for 5 failure
65	00	00	First Fail Info
66	00	01	Pattern no 1 (00:00:00:01)
67	00	00	Offset 0
68	00	00	
69	00	02	Pin 2(00:02)
70	10	00	Meas. Data 1, no Exp data; Second Fail Info starting at Odd Byte
71	00	00	
72	01	00	Pattern 1 (00:00:00:01)
73	00	00	
74	32	00	Offset 50 (00:00:00:32)
75	01	00	Pin 1(00:01); Meas. Data 0
76	00	00	Third fail start
77	00	02	Pattern 2 (00:00:00:02)
78	00	00	
79	00	00	Offset 0 (00:00:00:00)
80	00	02	Pin 2 (00:02)
81	10	00	Meas. Data 1; fourth fail start
82	00	00	Pattern 2 (00:00:00:02);
83	02	00	
84	00	00	Offset 50 (00:00:00:32)
85	32	00	
86	03	00	Pin 3(00:03);Meas. Data 00
87	00	00	
88	00	03	Pattern 3 (00:00:00:03)
89	00	00	
90	00	00	Offset 0 (00:00:00:00)
91	00	03	Pin 3 (00:03)
92	10		Meas. Data 1