SOFTWARE VALIDATION TEST PLAN AND REPORT FOR SAPHIRE VERSION 6.80

Prepared by

B. Dasgupta
F. Ferrante

Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute®
San Antonio, Texas

***Approved by*:**

Asadul H. Chowdhury
Manager, Mining, Geotechnical, and Facility Engineering

11-06-007
Date

CONTENTS

Section Page

1 SCOPE OF THE VALIDATION 1

2 REFERENCES 1

3 ENVIRONMENT 1
3.1 Software 1
3.2 Hardware 2

4 PREREQUISITES 2

5 ASSUMPTIONS AND CONSTRAINTS 2

6 TEST CASES 2
6.1 Test A 2
6.1.1 Test Procedure 2
6.1.2 Test Cases 3
6.1.2.1 Test Case TADLG–1 4
6.1.2.1.1 Test Problem 4
6.1.2.1.2 Test Input and Results 4
6.1.2.2 Test Case TORLG–1 4
6.1.2.2.1 Test Problem 4
6.1.2.2.2 Test Input and Results 4
6.1.2.3 Test Case TORLG–2 5
6.1.2.3.1 Test Problem 5
6.1.2.3.2 Test Input and Results 5
6.1.2.4 Test Case TFT1–A 5
6.1.2.4.1 Test Problem 5
6.1.2.4.2 Test Input and Results 6
6.1.2.5 Test Case TFT2–A 6
6.1.2.5.1 Test Problem 6
6.1.2.5.2 Test Input and Results 7
6.1.2.6 Test Case TFT3–A 7
6.1.2.6.1 Test Problem 7
6.1.2.6.2 Test Input and Results 8
6.1.2.7 Test Case TFT4–A 9
6.1.2.7.1 Test Problem 9
6.1.2.7.2 Test Input and Results 10
6.1.2.8 Test Case TET–1 10
6.1.2.8.1 Test Problem 10
6.1.2.8.2 Test Input and Results 11
6.2 Test B 11
6.2.1 Test Problem 11
6.2.2 Test Procedure 12

CONTENTS (continued)

Section Page

6.2.2.1 Test Results 13

7 NOTES 18

APPENDIX—SAPHIRE VERSION 6.0 FAULT TREE MODEL

FIGURES

Figure Page

1 Fault Tree Model TFT1–A 6

2 Fault Tree Model TFT2–A 7

3 Fault Tree Model TFT3–A 8

4 Fault Tree Model TFT4–A 9

5 Event Tree Model TET–1 10

6 Fault Tree Model for Test B 11

TABLES

Table Page

1 Test Cases Performed for Test A 3

2 Probability Values From Basic Event in Test B 13

3 Results From Test B 13

4 Results From SAPHIRE, MATLAB, and MATHCAD for Monte Carlo Simulation 17

5 Results From SAPHIRE and MATLAB for Monte Carlo and Latin Hypercube Sampling Simulation 17

6 List of Files in CD: SAPHIRE_ValidationTest 18

1 SCOPE OF THE VALIDATION

The Systems Analysis Programs for Hands-on Integrated Reliability Evaluation (SAPHIRE) (Russel, 1993) is a program for conducting systems reliability analysis for probabilistic risk assessments. Sponsored by the Nuclear Regulatory Commission (NRC), the SAPHIRE project is developed and maintained at the Idaho National Engineering and Environmental Laboratory Idaho Falls, Idaho. SAPHIRE is a standalone Microsoft® Windows®-based application software. SAPHIRE Version 6.80 validation testing was performed on a Windows 2000 Professional operating system.

As indicated, SAPHIRE is a general purpose probabilistic risk assessment software; however, selected capabilities were validated commensurate with the review of the preclosure safety analysis in the potential license application for the Yucca Mountain repository. The validation was focused on verifying event tree and fault tree analysis techniques. It was conducted by solving problems developed for this test and verifying the results with either hand calculations or MATLAB® and MATHCAD® software.

SAPHIRE Version 6.80 is incorporated in the current release of the PCSA Tool (Version 3.0.1 Beta C) and can be launched from the PCSA Tool (Maxwell, 2005). Once launched, SAPHIRE Version 6.80 operates independently from the PCSA Tool. There is no exchange of input or output data between the main module of the PCSA Tool and SAPHIRE. For this validation, SAPHIRE Version 6.80 is executed and tested independently of the PCSA Tool.

2 REFERENCES

Maxwell, T., B. Dasgupta, G. Adams, R. Benke, and N. Gisenberg. "PCSA Tool Version 3.0 User Guide." San Antonio, Texas: CNWRA. 2005.

Russel, K.D., C.L. Atwood, W.J. Galyean, M.B. Sattison, and D.M. Rasmuson. NUREG/CR–6616, "Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 5.0." Technical Reference Manual. Idaho National Engineering Laboratory. 1993.

3 ENVIRONMENT

3.1 Software

The SAPHIRE Version 6.80 executable and associated files are installed by downloading them to the target platform from the SAPHIRE website (i.e., http://saphirei.nel.gov). The downloadable version is available in the user area of the website, accessed by member identification and password. SAPHIRE Version 6.80 was downloaded to the host computer file folder c:\saf60. The download consisted of three file folders: (i) demo, (ii) samples, and (iii) tools and two files (i) _isreg32.dll and (ii) Delsl_.isu. The entire subdirectory (i.e., c:\saf60) was copied to the d:\Saphire-Validation folder for the validation test on machine DRACO. Additional folders were created under the d:\Saphire-Validation folder to store the test results. These subfolders contained the test cases discussed in Section 6.0.

3.2 Hardware

Standalone SAPHIRE Version 6.80 validation was performed on Windows personal computers. Computers used for this test were DRACO and GRIFFON. Windows 2000 Professional machine DRACO was used to execute SAPHIRE. Windows XP Professional machine GRIFFON was used to perform MATLAB and MATHCAD analyses.

4 PREREQUISITIES

DRACO was used for this validation test to execute SAPHIRE Version 6.80. The machine DRACO was set up in accordance with Section 3.1 prior to testing.

5 ASSUMPTIONS AND CONSTRAINTS

There were no assumptions associated with the validation test of SAPHIRE Version 6.80.

Software errors were not encountered during the validation test. The test models are designed to be simple to perform verification primarily by hand calculations. Users may, however, encounter operational problems with the software while modeling complex problems.

6 TEST CASES

Two types of tests were conducted for validating SAPHIRE Version 6.80.

Test A: Develop a set of test problems for which solutions are derived by hand calculations considering point estimates of event probabilities and comparing the same with the output from SAPHIRE analysis.

Test B: Develop a test problem for which uncertainty analysis performed in SAPHIRE can be compared to results from routines constructed using the software MATLAB (with the MATLAB Statistical Toolbox included). For these purposes, MATLAB (Version 7.1.0.246 R14 Service Pack3, License Number 301039) was used to generate random numbers and perform the comparison calculations needed. MATLAB is a general purpose programming environment, developed by MathWorks, and suitable for the calculations required in this validation. Furthermore, the Statistical Toolbox included with this MATLAB installation provides some generic statistical commands used to develop the comparison results. MATLAB's plotting capabilities were also used to present results.

6.1 Test A

6.1.1 Test Procedure

The following steps are required:

- Create a test directory d:\Saphire-Validation on the target platform.

- Copy the two folders “tools” and “demo” from c:\programs\SAPHIRE-Version 6.80 to this test directory.
- Create a subdirectory “Validation Test Case A.”
- Run SAPHIRE from directory d:\Saphire-Validation\tools\sapwin.exe.
- Create a SAPHIRE Project “Test” and save the database in directory d:\Saphire-Validation\Validation Test A. SAPHIRE saves the project as TEST.sra, and all files associated with this test case are saved in this folder.
- Create the seven fault tree and one event tree models described in Section 6.1.2.
- Analyze each problem independently.
- Compare the output results of each problem displayed on screen or in reports to the expected output.

6.1.2 Test Cases

Test A consists of eight test cases. Table 1 shows the test details, which includes a test problem identifier, analysis technique tested, and attributes of the analysis technique validated. All analyses in this test are conducted with point estimate probabilities. All the test cases are modeled under one SAPHIRE project. The model, input data, and output are located in d:\Saphire-Validation\Validation Test A. Results from this project are accessed by opening the test.sra file from SAPHIRE.

Table 1. Test Cases Performed for Test A

Test Cases	Analysis Technique	Validation
TADLG–1	Fault Tree	AND Gate
TORLG–1	Fault Tree	OR Gate Case 1
TORLG–2	Fault Tree	OR Gate Case 2
TFT1–A	Fault Tree	Cut Set
TFT2–A	Fault Tree	Cut Set
TFT3–A	Fault Tree	Cut Set
TFT4–A	Fault Tree	Cut Set
TET–1	Event Tree	Event Sequences

6.1.2.1 Test Case TADLG–1

6.1.2.1.1 Test Problem

The TADLG–1 test case tests the AND logic of a fault tree model in SAPHIRE. The AND logic specifies the top event, Q, as the intersection of events A and B [$P(A \cap B)$]. Event Q occurs only if events A and B both occur. Consider P(A) and P(B) as the probabilities of two independent events A and B, respectively.

$$P(Q) = P(A \cap B) = P(A) * P(B)$$

Let $P(A) = 1 \times 10^{-2}$ and $P(B) = 2 \times 10^{-3}$,

$$P(Q) = 1 \times 10^{-2} * 2 \times 10^{-3} = 2 \times 10^{-5}$$

6.1.2.1.2 Test Input and Results

The SAPHIRE AND logic model and SAPHIRE report are shown in the appendix (p. 1). The name of basic events P(A) and P(B) in SAPHIRE are EVENT-AND-1 and EVENT-AND-2, respectively. The top event is TADLG–1, and the Mincut Upper Bound that represents the top event in SAPHIRE is 2×10^{-5}. The minimum cut set top event result agrees with the hand calculation; therefore, the validation test passes.

6.1.2.2 Test Case TORLG–1

6.1.2.2.1 Test Problem

The TORLG–1 test case tests the OR logic of a fault tree model in SAPHIRE with low probability of failure of basic events. The OR logic indicates the probability, Q, of either of two events A and B [$P(A \cup B)$]. Consider P(A) and P(B) as the probabilities of two independent events A and B, respectively.

$$P(Q) = P(A \cup B) = P(A) + P(B) - P(A \cup B)$$

Let $P(A) = 1 \times 10^{-3}$ and $P(B) = 1 \times 10^{-4}$,

$$P(Q) = 1 \times 10^{-3} + 1 \times 10^{-4} - 1 \times 10^{-3} * 1 \times 10^{-4} = 1.1 \times 10^{-3}$$

6.1.2.2.2 Test Input and Results

The SAPHIRE OR logic model and SAPHIRE report are shown in appendix (p. 2). The basic events are EVENT-OR1-1 = P(A) and EVENT-OR1-2 = P(B). The top event is TORLG–1 and the Mincut Upper Bound is 1.1×10^{-3}. The minimum cut set top event result agrees with the hand calculation; therefore, the validation test passes.

6.1.2.3 Test Case TORLG–2

6.1.2.3.1 Test Problem

The TORLG–2 test case tests the OR logic of a fault tree model in SAPHIRE with high probability of failure of basic events. The OR logic indicates the probability Q of either of two events A and B [$P(A \cup B)$]. Consider P(A) and P(B) are the probabilities of two independent events A and B, respectively.

$$P(Q) = P(A \cup B) = P(A) + P(B) - P(A \cup B)$$

Let P(A) = 0.1 and P(B) = 0.2,

$$P(Q) = 0.1 + 0.2 - 0.1 * 0.2 = 0.28$$

6.1.2.3.2 Test Input and Results

The SAPHIRE OR logic model and SAPHIRE report are shown in the appendix (p. 3). The basic events are EVENT-OR2-1 = P(A) and EVENT-OR2-2 = P(B). The top event is TORLG–2 and the Mincut Upper Bound is 2.8×10^{-1}. The minimum cut set top event result agrees with the hand calculation; therefore, the validation test passes.

6.1.2.4 Test Case TFT1–A

6.1.2.4.1 Test Problem

The TFT1–A test case tests the cut set identification and associated calculations of a fault tree model in SAPHIRE. The fault tree model is shown in Figure 1. The fault tree consists of top event **A**, two gates **B** and **C**, and four basic events named **1**, **2**, **3**, and **4**. Probability values for basic events **1**, **2**, **3**, and **4** are assumed to be 1×10^{-4}, 5×10^{-4}, 2×10^{-5}, and 8×10^{-5}, respectively.

Cut set calculations

Top Event = **A**
= **B * C**
= **(1 + 2) * (3 + 4)**
= **1*3 + 1*4 + 2*3 +2*4**

Minimal cut sets and probability values for this model are.

(i) **1** and **3** = 2×10^{-9}
(ii) **1** and **4** = 8×10^{-9}
(iii) **2** and **3** = 1×10^{-8}
(iv) **2** and **4** = 4×10^{-8}

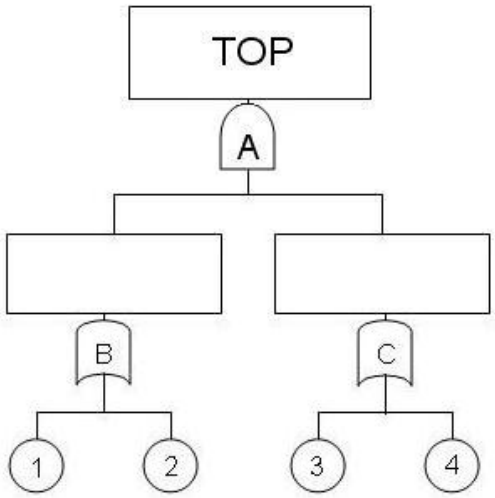


Figure 1. Fault Tree Model TFT1–A

6.1.2.4.2 Test Input and Results

The SAPHIRE model and SAPHIRE report are shown in the appendix (p. 4). The basic events are EVENT-1, EVENT-2, EVENT-3, and EVENT-4, corresponding to the basic events **1**, **2**, **3**, and **4**, respectively, shown in Figure 1. The cut sets and the probability results agree with the hand calculation; therefore, the validation test passes. Note that SAPHIRE displays tabulated values of the cut sets as "frequency," even though the calculated result could be probability.

6.1.2.5 Test Case TFT2–A

6.1.2.5.1 Test Problem

The TFT2–A test case tests the cut sets of a fault tree model in SAPHIRE. The fault tree model is shown in Figure 2. The fault tree consists of top event **A**, two gates **B** and **C,** and four basic events named **1**, **2**, **3**, and **4**. Probability values for basic events **1**, **2**, **3**, and **4** are assumed to be 1×10^{-4}, 5×10^{-4}, 2×10^{-5}, and 8×10^{-5}, respectively.

Cut set calculations

Top Event = **A**
= **B + C**
= **(1 * 2) + (3 * 4)**
= **1 * 2 + 3 * 4**

The cut sets and probability values for this model are:

(i) 1 and 2 = 5×10^{-8}
(ii) 3 and 4 = 1.6×10^{-9}

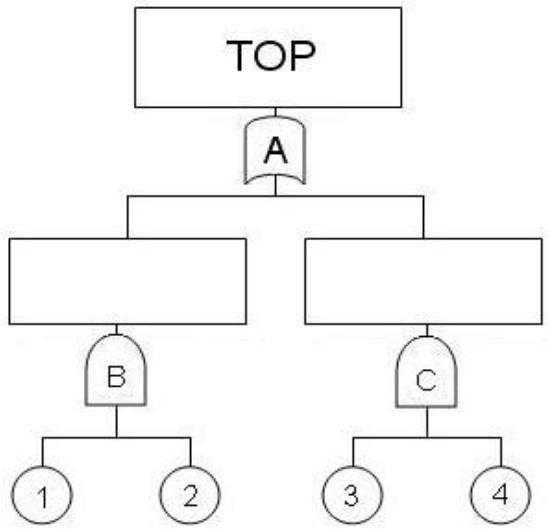


Figure 2. Fault Tree Model TFT2–A

6.1.2.5.2 Test Input and Results

The SAPHIRE model and SAPHIRE report are shown in the appendix (p. 5). The basic events are EVENT-1, EVENT-2, EVENT-3, and EVENT-4, corresponding to the basic events **1**, **2**, **3**, and **4**, respectively, shown in Figure 2. The cut set and the probability results agree with the hand calculation; therefore, the validation test passes.

6.1.2.6 Test Case TFT3–A

6.1.2.6.1 Test Problem

The TFT3–A test case tests the cut set identification and cut set calculations for a fault tree model in SAPHIRE. The fault tree model is shown in Figure 3. The fault tree consists of top event **A**; seven gates **B**, **C**, **D**, **E**, **F**, **G**, and **H**; nine basic events named **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, and **9**. Probability values for basic events **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, and **9** are assumed to be 1×10^{-4}, 5×10^{-4}, 2×10^{-5}, 8×10^{-5}, 1×10^{-3}, 5×10^{-3}, 4×10^{-4}, 3×10^{-5}, and 9×10^{-5}, respectively.

Cut set calculations

Top Event = **A**
= B + C
= 1 * 2 + 3 + D
= 1 * 2 + 3 + E + F
= 1 * 2 + 3 + 4 + G + 5 + 6
= 1 * 2 + 3 + 4 + 7 + H + 5 + 6
= 1 * 2 + 3 + 4 + 7 + 8 + 9 + 5 + 6

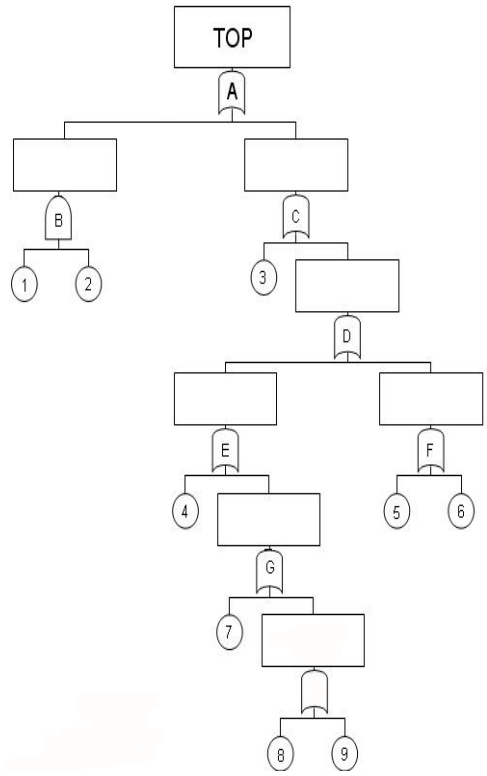


Figure 3. Fault Tree Model TFT3–A

Minimal cut sets and probability values for this model are

(i) **6** = 5×10^{-3}
(ii) **5** = 1×10^{-3}
(iii) **9** = **9** $\times 10^{-5}$
(iv) **8** = 3×10^{-5}
(v) **7** = 4×10^{-4}
(vi) **4** = $8 \times 10^{-5,}$
(vii) **3** = 2×10^{-5}
(viii) **1, 2** = 5×10^{-8}

6.1.2.6.2 Test Input and Results

The SAPHIRE model and SAPHIRE report for TFT3–A are shown in the appendix (p. 6). The basic events are EVENT-1, EVENT-2, EVENT-3, EVENT-4, EVENT-5, EVENT-6, EVENT-7, EVENT-8, and EVENT-9 corresponding to the basic events **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, and **9**, respectively, shown in Figure 3. The cut sets and the probability results agree with the hand calculation; therefore, the validation test passes.

6.1.2.7 Test Case TFT4–A

6.1.2.7.1 Test Problem

The TFT4–A test case tests the cut set determination and cut set calculations for a fault tree model in SAPHIRE. The fault tree model is shown in Figure 4. The fault tree consists of top event **A**, five gates **B**, **C**, **D**, **E**, and **F** and six basic events named as **1**, **2**, **3**, **4**, **5**, and **6**. Probability/frequency values for basic events **1**, **2**, **3**, **4**, **5**, **a**nd **6** are assumed to be 1×10^{-4}, 5×10^{-4}, 2×10^{-5}, 8×10^{-5} ,1×10^{-3}, 1×10^{-3}, and 5×10^{-3} respectively.

Cut set calculations

Top Event = **A**
= **B + 1**
= **C * D + 1**
= **(2 + E) * (5 + F) +1**
= **(2 + 3 *4) + (5 + 6 *1) +1**
= **2 * 5 + 2 * 6 *1 + 3 * 4 * 5 + 3 * 4 * 6 * 1**

Minimal cut sets and frequencies for this model are given below:

i) **1** = 1×10^{-4}
ii) **2**, **5** = 5×10^{-7}
iii) **3, 4, 5** = 1.6×10^{-12}

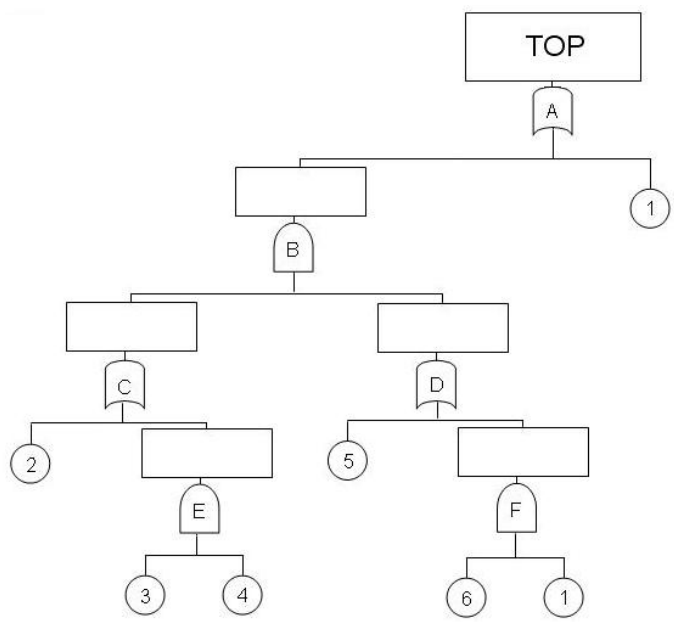


Figure 4. Fault Tree Model TFT4–A

6.1.2.7.2 Test Input and Results

The SAPHIRE model and SAPHIRE report for TFT4–A are shown in the appendix (p. 7). The basic events are EVENT-1, EVENT-2, EVENT-3, EVENT-4, EVENT-5, and EVENT-6, corresponding to the basic events **1**, **2**, **3**, **4**, **5**, and **6**, respectively, shown in Figure 4. The cut sets and the probability results agree with the hand calculation; therefore, the validation test passes.

6.1.2.8 Test Case TET–1

6.1.2.8.1 Test Problem

The TET–1 test case tests an event tree analysis in SAPHIRE. The event tree model is shown in Figure 5. The event tree consists of an initiating event **IN-ET** and subsequent events **EVT–1**, **EVT–2**, and **EVT–3**. As shown in Figure 5, the event tree generates eight event sequences **A** to **H**. The initiating event frequency and the event probabilities follow.

IN-ET frequency f = 0.1
EVT–1 probability p_1 = 0.01
EVT–2 probability p_2 = 0.06
EVT–3 probability p_3 = 0.002

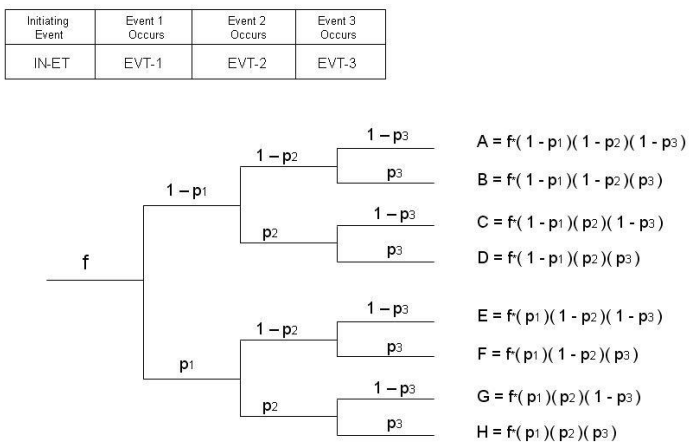


Figure 5. Event Tree Model TET–1

The event sequence analysis is

A = $f*(1 - p_1)*(1 - p_2)*(1 - p_3)$ = 0.1*0.99*0.94*0.998 = 0.09287
B = $f*(1 - p_1)*(1 - p_2)*(p_3)$ = 0.1*0.99*0.94*0.002 = 1.861×10^{-4}
C = $f*(1 - p_1)*(p_2)*(1 - p_3)$ = 0.1*0.99*0.06*0.998 = 5.928×10^{-3}
D = $f*(1 - p_1)*(p_2)*(p_3)$ = 0.1*0.99*0.06*0.002 = 1.188×10^{-5}
E = $f*(p_1)*(1 - p_2)*(1 - p_3)$ = 0.1*0.01*0.94*0.998 = 9.381×10^{-4}
F = $f*(p_1)*(1 - p_2)*(p_3)$ = 0.1*0.01*0.94*0.002 = 1.88×10^{-6}

G = $f*(p_1)*(p_2)*(1 - p_3)$ = 0.1*0.01*0.06*0.998 = 5.988×10^{-5}
H = $f*(p_1)*(p_2)*(p_3)$ = 0.1*0.01*0.06*0.002 = 1.2×10^{-7}

6.1.2.8.2 Test Input and Results

The SAPHIRE model and results for TET–1 are shown in the appendix (p. 8). The initiating events and subsequent events in the SAPHIRE model have the same names stated in Section 6.1.2.8.1. The event sequence frequency result agrees with the hand calculation; therefore, the validation test passes.

6.2 Test B

Test B consists of executing an uncertainty analysis for a generic fault tree using the Monte Carlo sampling routine in SAPHIRE.

6.2.1 Test Problem

Test B tests the uncertainty analysis in the minimal cut set calculations of a generic fault tree in SAPHIRE. The fault tree model is shown in Figure 6. The fault tree consists of the top event **TOP EVENT 1** AND-gate;, the OR-gates **GATE 1**, **GATE 2**, **GATE A**, **GATE B**, and **GATE C**;

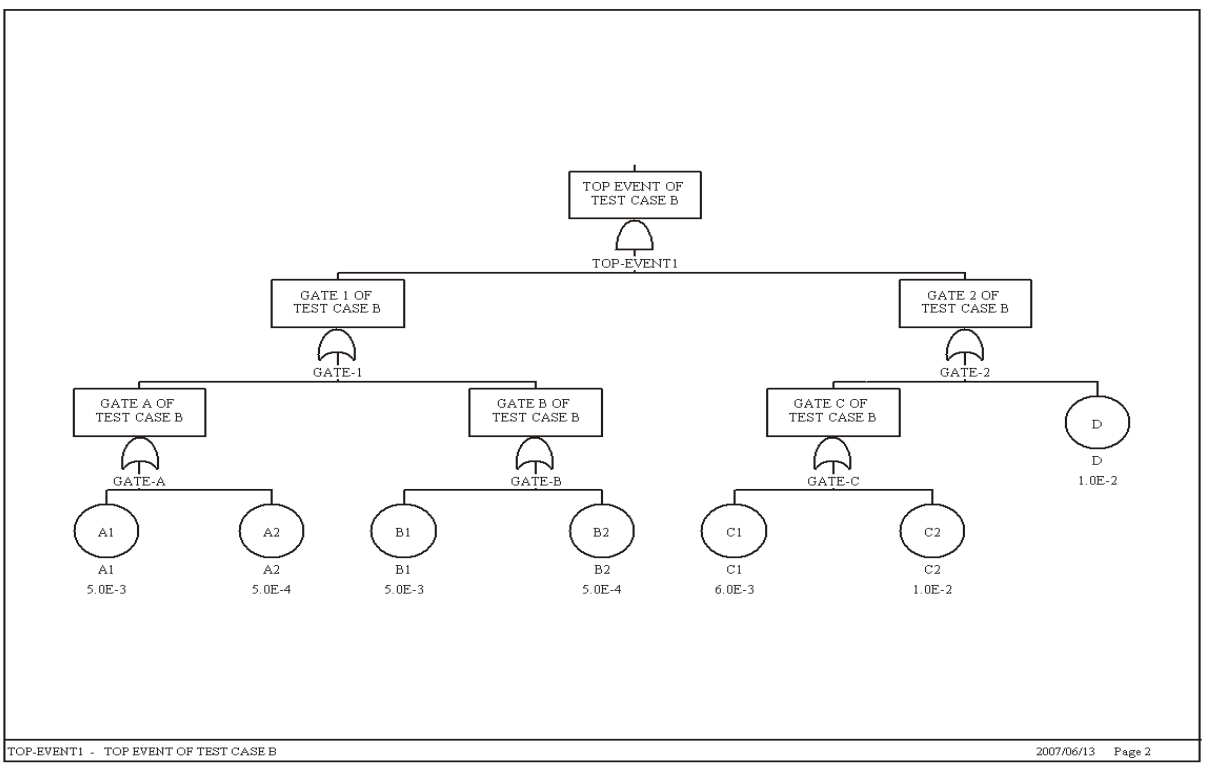


Figure 6. Fault Tree Model for Test B

and six basic events **A1**, **A2**, **B1**, **B2**, **C1**, **C2**, and **D**. Probability values for basic events **A1**, **A2**, **B1**, **B2**, **C1**, **C2**, and **D** are assumed to be 5×10^{-3}, 5×10^{-4}, 5×10^{-3}, 5×10^{-4}, 6×10^{-3}, 1×10^{-2}, and 1×10^{-2}, respectively.

The following minimat cut sets result from the analysis:

Top Event = **TOP EVENT**
= **GATE 1 * GATE 2**
= **(GATE A + GATE B) * (GATE C + D)**
= **(A1 + A2 + B1 + B2) * (C1 + C2 + D)**
= **A1 * C1 + A1 * C2 + A1 * D + A2 * C1 + A2 * C2 + A2 * D + B1 * C1 + B1 * C2 + B1 * D + B2 * C1 + B2 * C2 + B2 * D**

Hand-calculated minimal cut sets and probabilities for this model are

(i) **A1, C1** = 3×10^{-5}
(ii) **A1, C2** = 5×10^{-5}
(iii) **A2, C1** = 3×10^{-6}
(iv) **A2, C2** = 5×10^{-6}
(v) **B1, C1** = 3×10^{-5}
(vi) **B1, C2** = 5×10^{-5}
(vii) **B2, C1** = 3×10^{-6}
(viii) **B2, C2** = 5×10^{-6}
(ix) **A1, D** = 5×10^{-5}
(x) **A2, D** = 5×10^{-6}
(xi) **B1, D** = 5×10^{-5}
(xii) **B2, D** = 5×10^{-6}

For uncertainty purposes, it will be assumed that the probabilities of the basic events **A1**, **A2**, **B1**, **B2**, **C1**, and **C2** follow lognormal distributions with mean and error factors given in Table 2 (where the error factor is the ratio between the 95th percentile and the median value).

6.2.2 Test Procedure

The following steps are performed:

- Run SAPHIRE Version 6.80.
- Build fault tree model shown in Figure 6.
- Input parameters required for uncertainty analysis purposes (i.e., Table 2).
- Solve problem using SAPHIRE minimal cut set analysis capabilities.
- Run uncertainty analysis using SAPHIRE using Monte Carlo sampling capabilities.

Table 2. Probability Values From Basic Event in Test B		
Basic Event	**Mean**	**Error Factor**
A1	5×10^{-3}	3
A2	5×10^{-4}	10
B1	5×10^{-3}	3
B2	5×10^{-4}	10
C1	6×10^{-3}	5
C2	1×10^{-2}	10

- Compare the output results with analysis performed using MATLAB software for independent random sampling estimates and MATHCAD software for approximated theoretical values expected for the fault tree model developed.

6.2.2.1 Test Results

The minimal cut set analysis performed in SAPHIRE results in a probability of occurrence of the TOP EVENT of 2.86×10^{-4}. The following output is obtained for the minimal cut sets (see file MINCUTSETS.txt):

Project-> VALIDATION Fault Tree-> TOP-EVENT1
Mincut Upper Bound -> 2.860×10^{-4} This Partition -> 2.860×10^{-4}

To run the uncertainty analysis in SAPHIRE, the maximum number of samples allowed is chosen (99998) using Monte Carlo sampling. The results obtained are shown in Table 3, along with results generated using MATLAB through the routine “testcase.m” (along with subroutine “genlogrand.m” to generate random variables), which were developed specifically for this test comparison. A MATHCAD file titled Test Case B.mcd shows the approximated values of the mean and standard deviation for the TOP EVENT, based on the first order Taylor expansion approximation method (reproduced on pp. 15 and 16):

Table 3. Results for Test B				
Cut No.	**% Total**	**% Cut Set**	**Frequency**	**Cut Sets**
1	17.5	17.5	5.000×10^{-5}	A1, C2
2	35	17.5	5.000×10^{-5}	B1, C2
3	52.5	17.5	5.000×10^{-5}	A1, D
4	70	17.5	5.000×10^{-5}	B1, D
5	80.5	10.5	3.000×10^{-5}	A1, C1
6	91	10.5	3.000×10^{-5}	B1, C1

Table 3. Results for Test Case B (continued)				
Cut No.	**% Total**	**% Cut Set**	**Frequency**	**Cut Sets**
7	92.7	1.8	5.000×10^{-6}	A2, C2
8	94.5	1.8	5.000×10^{-6}	B2, C2
9	96.2	1.8	5.000×10^{-6}	A2, D
10	98	1.8	5.000×10^{-6}	B2, D
11	99	1.1	3.000×10^{-6}	A2, C1
12	100	1.1	3.000×10^{-6}	B2, C1

The results from SAPHIRE are obtained with the random seed number "12345" with 99998 samples for each parameter, and the project in which the fault tree model shown in Figure 6 was developed and analyzed is saved under "Validate.sra" (along with other supporting files generated). The MATLAB results were obtained with MATLAB's built-in random number generator for 1 million samples and are saved in the file "ValidationMatlab.mat."

The Latin Hypercube Sampling routine in SAPHIRE is used to perform the same minimal cut set analysis as that previously mentioned, and its convergence is compared to SAPHIRE Monte Carlo sampling results as well as MATLAB results. The random seeds "1000," "10000," "20000," "30000," "40000," "50000," "60000," "70000," "80000," and "90000" are used in SAPHIRE for both Monte Carlo and Latin Hypercube Sampling , each generating 50 samples for the parameters in each analysis. Table 4 shows the results for the mean and standard deviation for each of the 10 runs. The standard deviations of the mean probability value from the Monte Carlo and Latin Hypercube Sampling results are 2.800×10^{-5} and 1.539×10^{-5}, respectively, showing better convergence by the latter. Similarly, the spread of the standard deviation probability value from the Monte Carlo and Latin Hypercube Sampling results is also measured (i.e., standard deviations of 1.093×10^{-4} and 7.360×10^{-5}, respectively), with equivalent results. The MATLAB file "testcaseblhs.m" with the subroutine "genlhs.m" developed for Latin Hypercube Sampling of the same SAPHIRE problem are also listed in Table 5 and the results are saved under "ValidationMatlablhs.mat." The standard deviations of the mean and standard deviation probability of TOP EVENT with the MATLAB files result in 1.467×10^{-5} and 9.100×10^{-5}, which are comparable to the spread obtained with the SAPHIRE Latin Hypercube Sampling results.

Mean Probability Values and Error Factor Values (assuming lognormal distribution for all variables, except D which is assumed to be constant):

$a_1 := 5\cdot 10^{-3}$ $a_2 := 5\cdot 10^{-4}$ $b_1 := 5\cdot 10^{-3}$ $b_2 := 5\cdot 10^{-4}$ $c_1 := 6\times 10^{-3}$ $c_2 := 0.01$ $d := 0.01$

$EF_{a1} := 3$ $EF_{a2} := 10$ $EF_{b1} := 3$ $EF_{b2} := 10$ $EF_{c1} := 5$ $EF_{c2} := 10$

Standard Deviations (STD) using the properties of lognormal distributions:

ERROR FUNCTION $EF = \exp\left(\mathrm{qnorm}(0.95,0,1)\cdot\sigma_X\right)$ hence $\sigma_X = \dfrac{\ln(EF)}{\mathrm{qnorm}(0.95,0,1)}$

CONVERSION OF MEAN,

STD FROM LOGNORMAL TO NORMAL SPACE $\mu_X = \ln\left[\mu_Y\cdot\exp\left[-0.5\cdot\left(\dfrac{\ln(EF)}{\mathrm{qnorm}(0.95,0,1)}\right)^2\right]\right]$

CONVERSION OF STD FROM

NORMAL TO LOGNORMAL SPACE $\sigma_Y = \sqrt{\left[\exp\left[\left(\sigma_X\right)^2\right]-1\right]\cdot\exp\left[2\cdot\mu_X+\left(\sigma_X\right)^2\right]}$

$\sigma_X := \dfrac{\ln\left(EF_{a1}\right)}{\mathrm{qnorm}(0.95,0,1)}$ $\sigma_{a1} := \sqrt{\left[\exp\left[\left(\sigma_X\right)^2\right]-1\right]\cdot\exp\left[2\cdot\ln\left[a_1\cdot\exp\left[-0.5\cdot\left(\sigma_X\right)^2\right]\right]+\left(\sigma_X\right)^2\right]}$

$\sigma_{a1} = 3.749\times 10^{-3}$

$\sigma_X := \dfrac{\ln\left(EF_{a2}\right)}{\mathrm{qnorm}(0.95,0,1)}$ $\sigma_{a2} := \sqrt{\left[\exp\left[\left(\sigma_X\right)^2\right]-1\right]\cdot\exp\left[2\cdot\ln\left[a_2\cdot\exp\left[-0.5\cdot\left(\sigma_X\right)^2\right]\right]+\left(\sigma_X\right)^2\right]}$

$\sigma_{a2} = 1.235\times 10^{-3}$

$\sigma_X := \dfrac{\ln\left(EF_{b1}\right)}{\mathrm{qnorm}(0.95,0,1)}$ $\sigma_{b1} := \sqrt{\left[\exp\left[\left(\sigma_X\right)^2\right]-1\right]\cdot\exp\left[2\cdot\ln\left[b_1\cdot\exp\left[-0.5\cdot\left(\sigma_X\right)^2\right]\right]+\left(\sigma_X\right)^2\right]}$

$\sigma_{b1} = 3.749\times 10^{-3}$

$\sigma_X := \dfrac{\ln\left(EF_{b2}\right)}{\mathrm{qnorm}(0.95,0,1)}$ $\sigma_{b2} := \sqrt{\left[\exp\left[\left(\sigma_X\right)^2\right]-1\right]\cdot\exp\left[2\cdot\ln\left[b_2\cdot\exp\left[-0.5\cdot\left(\sigma_X\right)^2\right]\right]+\left(\sigma_X\right)^2\right]}$

$\sigma_{b2} = 1.235\times 10^{-3}$

$$\sigma_X := \frac{\ln(EF_{c1})}{\text{qnorm}(0.95,0,1)} \quad \sigma_{c1} := \sqrt{\left[\exp\left[(\sigma_X)^2\right] - 1\right]\cdot\exp\left[2\cdot\ln\left[c_1\cdot\exp\left[-0.5\cdot(\sigma_X)^2\right]\right]\right] + (\sigma_X)^2}$$

$$\sigma_{c1} = 7.601\times 10^{-3}$$

$$\sigma_X := \frac{\ln(EF_{c2})}{\text{qnorm}(0.95,0,1)} \quad \sigma_{c2} := \sqrt{\left[\exp\left[(\sigma_X)^2\right] - 1\right]\cdot\exp\left[2\cdot\ln\left[c_2\cdot\exp\left[-0.5\cdot(\sigma_X)^2\right]\right]\right] + (\sigma_X)^2}$$

$$\sigma_{c2} = 0.025$$

Calculate mean probability of TOP EVENT based on the 1st order Taylor Series expansion approximation, using the results for the minimal cut set analysis:

$$\mu_{TOPEVENT} := a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d$$

$$\mu_{TOPEVENT} = 2.86\times 10^{-4}$$

Calculate the standard deviation of the probability of TOP EVENT based on the 1st order Taylor Series expansion approximation, again, using the results for the minimal cut set analysis:

$$\sqrt{\begin{aligned}
&\left[\frac{d}{da_1}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{a1}{}^2 \ldots \\
&+\left[\frac{d}{da_2}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{a2}{}^2 \ldots \\
&+\left[\frac{d}{db_1}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{b1}{}^2 \ldots \\
&+\left[\frac{d}{db_2}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{b2}{}^2 \ldots \\
&+\left[\frac{d}{dc_1}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{c1}{}^2 \ldots \\
&+\left[\frac{d}{dc_2}\begin{pmatrix} a_1\cdot c_1 + a_1\cdot c_2 + a_2\cdot c_1 + a_2\cdot c_2 \ldots \\ + b_1\cdot c_1 + b_1\cdot c_2 + b_2\cdot c_1 + b_2\cdot c_2 \ldots \\ + a_1\cdot d + a_2\cdot d + b_1\cdot d + b_2\cdot d \end{pmatrix}\right]^2\cdot\sigma_{c2}{}^2
\end{aligned}} = 3.191\times 10^{-4}$$

Table 4. Results From SAPHIRE, MATLAB, and MATHCAD for Monte Carlo Simulation			
TOP EVENT Parameter	**SAPHIRE**	**MATLAB**	**MATHCAD**
Mean	2.864×10^{-4}	2.857×10^{-4}	2.86×10^{-4}
5th Percentile	8.148×10^{-5}	7.892×10^{-5}	—
Median	2.053×10^{-4}	2.043×10^{-4}	—
95th Percentile	7.251×10^{-4}	7.263×10^{-4}	—
Standard Deviation	3.363×10^{-4}	3.388×10^{-4}	3.191×10^{-4}

Table 5. Results From SAPHIRE and MATLAB for Monte Carlo and Latin Hypercube Sampling Simulation						
RUN	**SAPHIRE (MC)**		**SAPHIRE (LHS)**		**MATLAB (LHS)**	
	MEAN	**STD**	**MEAN**	**STD**	**MEAN**	**STD**
1	3.035×10^{-4}	2.301×10^{-4}	3.005×10^{-4}	3.844×10^{-4}	2.633×10^{-4}	1.775×10^{-4}
2	2.555×10^{-4}	1.663×10^{-4}	3.101×10^{-4}	3.472×10^{-4}	2.716×10^{-4}	2.094×10^{-4}
3	3.159×10^{-4}	5.425×10^{-4}	2.737×10^{-4}	2.095×10^{-4}	2.752×10^{-4}	2.415×10^{-4}
4	3.198×10^{-4}	2.900×10^{-4}	2.840×10^{-4}	2.830×10^{-4}	2.697×10^{-4}	2.574×10^{-4}
5	3.120×10^{-4}	2.485×10^{-4}	2.594×10^{-4}	1.847×10^{-4}	2.671×10^{-4}	2.023×10^{-4}
6	2.466×10^{-4}	1.816×10^{-4}	2.916×10^{-4}	3.083×10^{-4}	2.729×10^{-4}	2.699×10^{-4}
7	3.002×10^{-4}	2.920×10^{-4}	2.729×10^{-4}	1.910×10^{-4}	2.894×10^{-4}	3.267×10^{-4}
8	2.539×10^{-4}	1.71×10^{-4}	2.823×10^{-4}	2.296×10^{-4}	2.727×10^{-4}	2.121×10^{-4}
9	3.049×10^{-4}	2.587×10^{-4}	2.734×10^{-4}	1.967×10^{-4}	3.138×10^{-4}	4.924×10^{-4}
10	3.023×10^{-4}	2.169×10^{-4}	2.972×10^{-4}	3.332×10^{-4}	2.724×10^{-4}	2.331×10^{-4}

7 NOTES

There is one attachment to this validation report. All the files required for conducting SAPHIRE Version 6.80 validation including the test results and output from SAPHIRE analysis are contained on a CD labeled SAPHIRE Version 6.80 Validation (attachment). A short description of the list of files on the CD is provided in Table 6. Note: If the read-only attribute is set for any of the files copied to the test directory, clear this attribute.

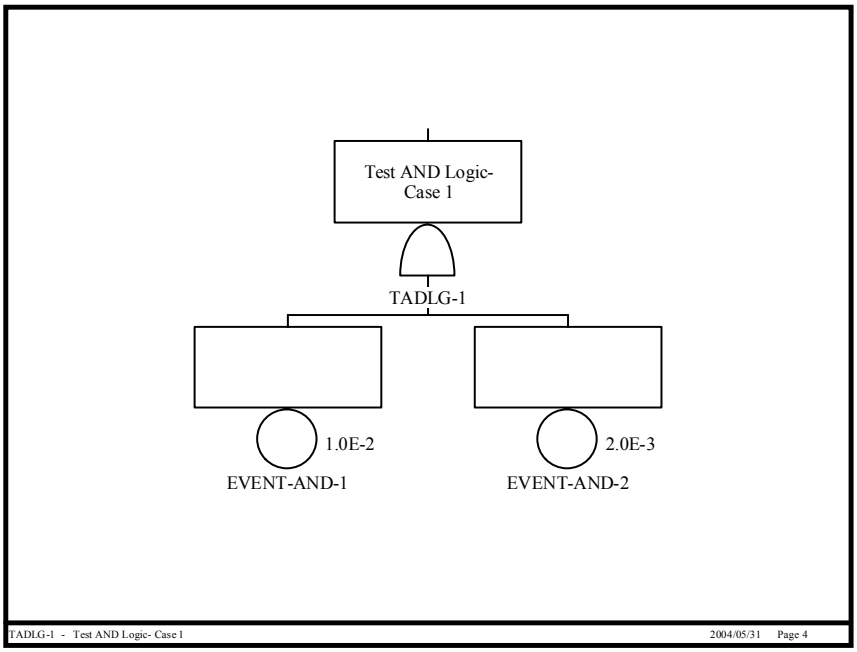
Table 6. List of Files in CD: SAPHIRE_ValidationTest

Directory Name	Contents
Tools	SAPHIRE tools
Demo	Demonstration problem files provided with SAPHIRE installation
Samples	Sample files provided with SAPHIRE installation
Validation Test Case A	SAPHIRE project files for Test A
Validation Test Case B	SAPHIRE project files for Test B MATLAB and MATHCAD routines for verification

APPENDIX
SAPHIRE VERSION 6.80 FAULT TREE MODEL

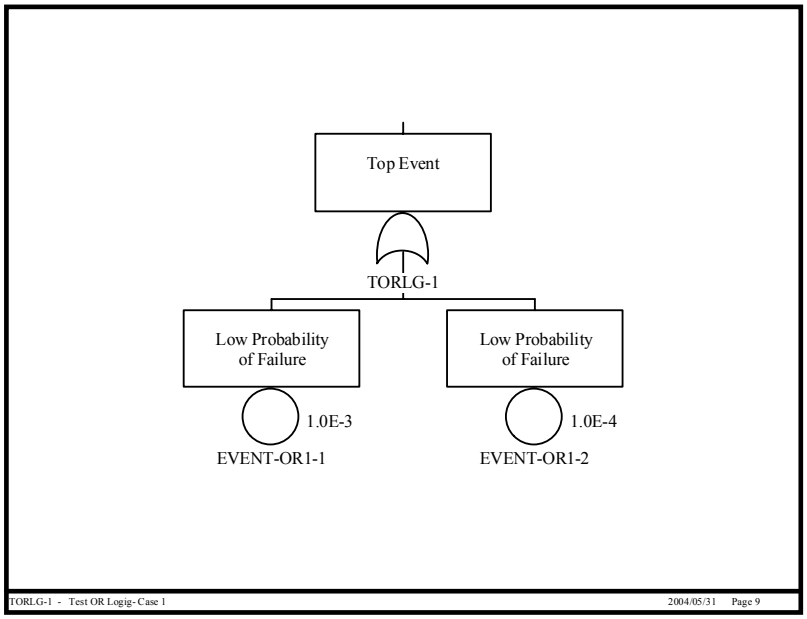
APPENDIX
SAPHIRE VERSION 6.80 OUTPUT FOR TEST CASE A

SAPHIRE Fault Tree Model: TADLG–1




```
                SAPHIRE Cut set Report

                         Sort/Slice Cut Set Report

  Project-> TEST                          Fault Tree-> TADLG-1
  Mincut Upper Bound -> 2.000E-005        This Partition -> 2.000E-005

   Cut      %     % Cut

   No.    Total    Set     Frequency                       Cut Sets

-------- -----  -----  -----------  ------------------------------------------
-
       1 100.0  100.0   2.000E-005  EVENT-AND-1, EVENT-AND-2
```

SAPHIRE VERSION 6.80 Fault Tree Model: TORLG–1

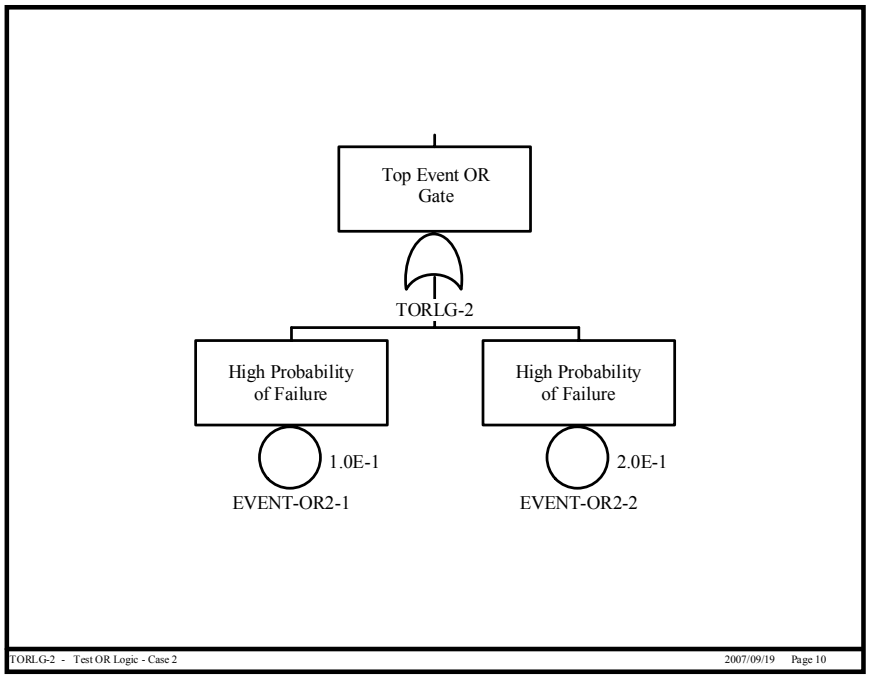


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TORLG-1
Mincut Upper Bound -> 1.100E-003 This Partition -> 1.100E-003

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	90.9	90.9	1.000E-003	EVENT-OR1-1
2	100.0	9.1	1.000E-004	EVENT-OR1-2

SAPHIRE VERSION 6.80 Fault Tree Model: TORLG–2

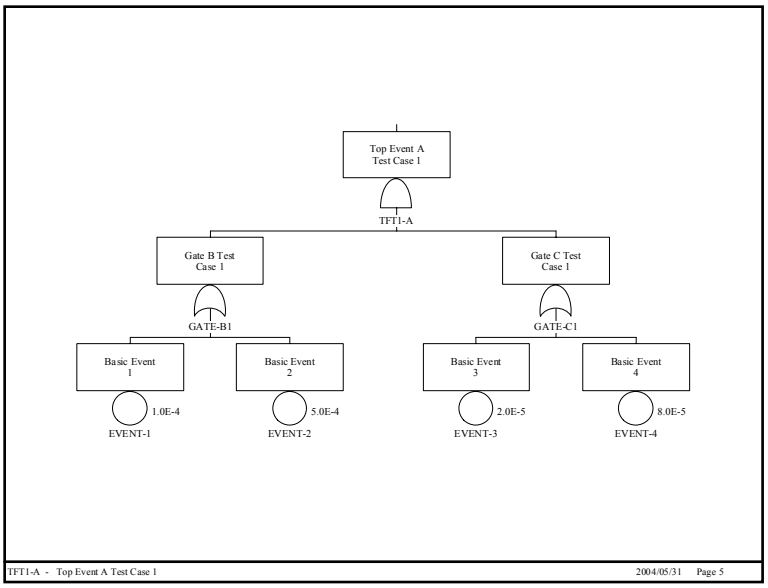


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TORLG-2
Mincut Upper Bound -> 2.800E-001 This Partition -> 2.800E-001

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	71.4	71.4	2.000E-001	EVENT-OR2-2
2	100.0	35.7	1.000E-001	EVENT-OR2-1

SAPHIRE VERSION 6.80 Fault Tree Model: TFT1–A

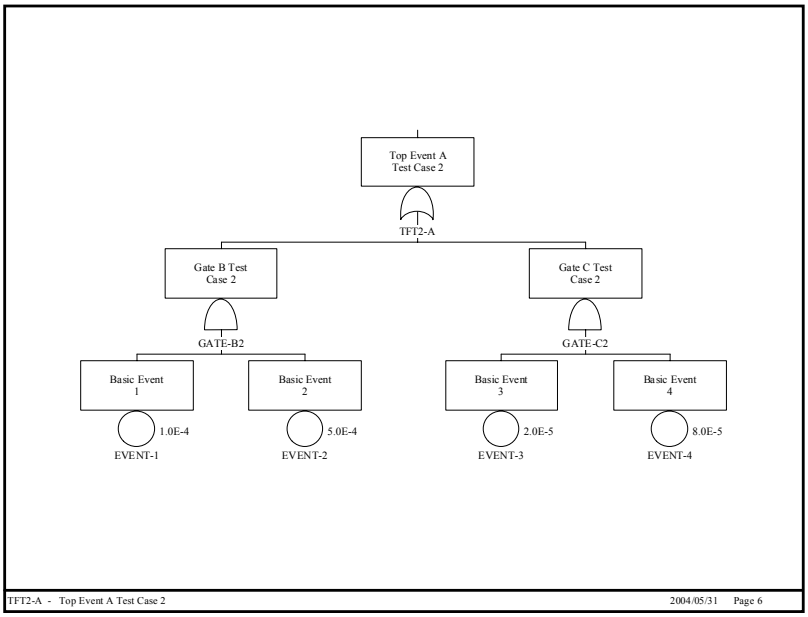


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TFT1-A
Mincut Upper Bound -> 6.000E-008 This Partition -> 6.000E-008

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	66.7	66.7	4.000E-008	EVENT-2, EVENT-4
2	83.3	16.7	1.000E-008	EVENT-2, EVENT-3
3	96.7	13.3	8.000E-009	EVENT-1, EVENT-4
4	100.0	3.3	2.000E-009	EVENT-1, EVENT-3

SAPHIRE VERSION 6.80 Fault Tree Model: TFT2–A

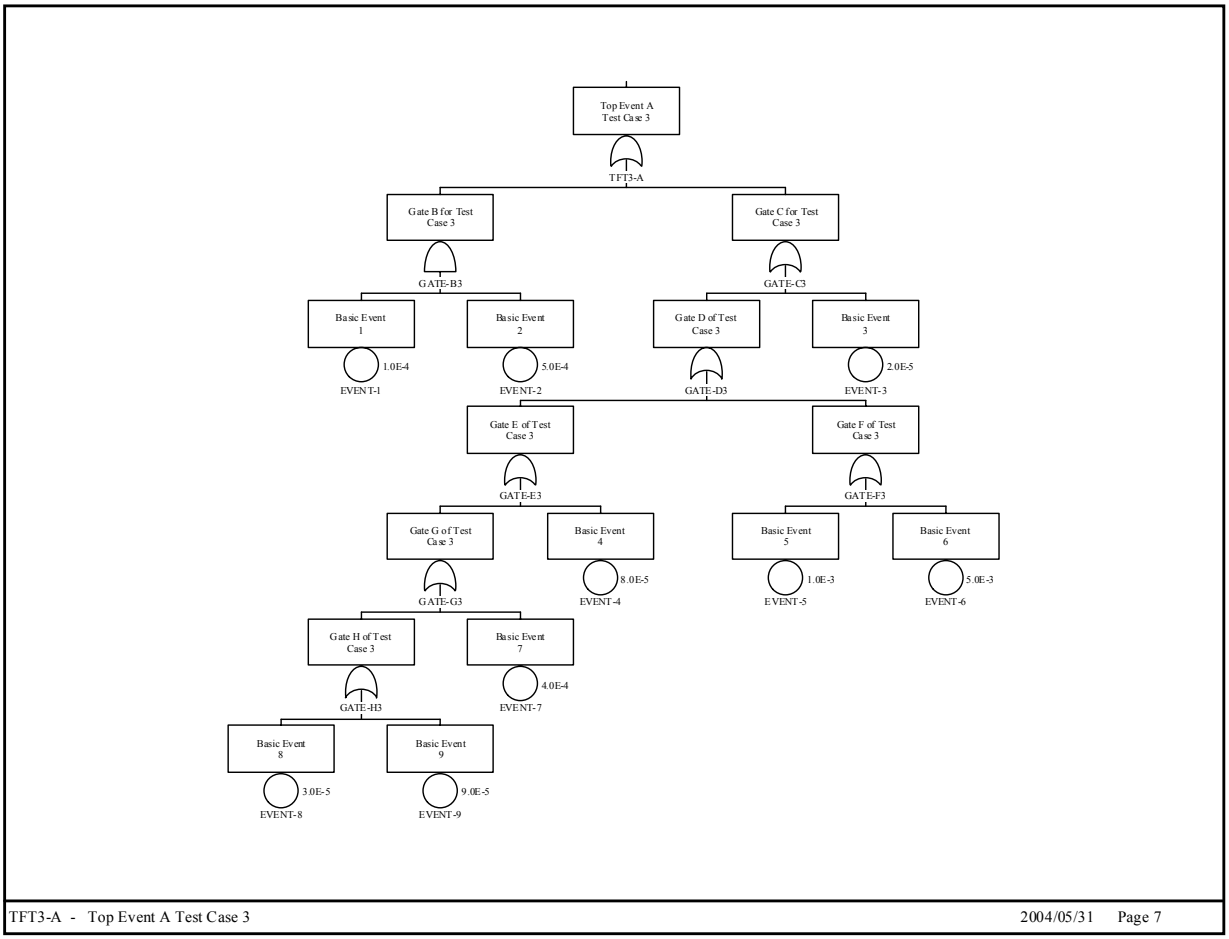


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TFT2-A
Mincut Upper Bound -> 5.160E-008 This Partition -> 5.160E-008

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	96.9	96.9	5.000E-008	EVENT-1, EVENT-2
2	100.0	3.1	1.600E-009	EVENT-3, EVENT-4

SAPHIRE VERSION 6.80 Fault Tree Model: TFT3–A

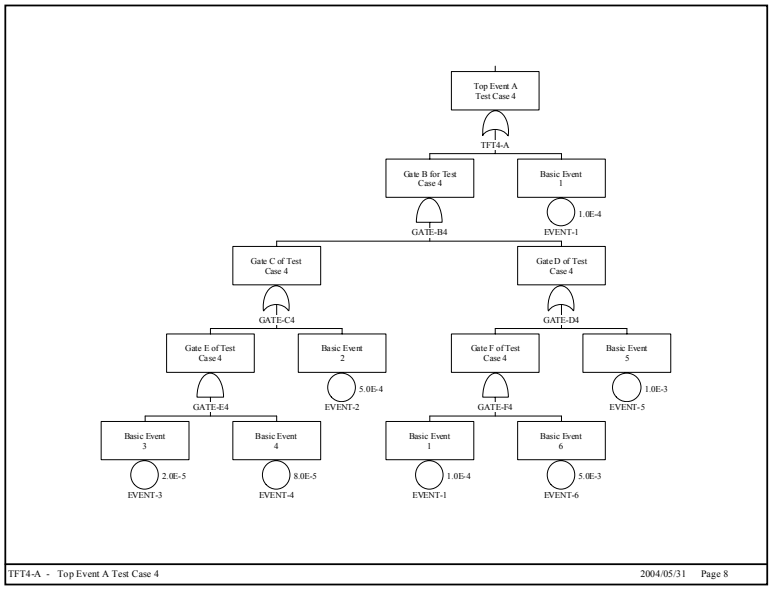


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TFT3-A
Mincut Upper Bound -> 6.611E-003 This Partition -> 6.611E-003

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	75.6	75.6	5.000E-003	EVENT-6
2	90.8	15.1	1.000E-003	EVENT-5
3	96.8	6.1	4.000E-004	EVENT-7
4	98.2	1.4	9.000E-005	EVENT-9
5	99.4	1.2	8.000E-005	EVENT-4
6	99.9	0.5	3.000E-005	EVENT-8
7	100.0	0.3	2.000E-005	EVENT-3
8	100.0	0.0	5.000E-008	EVENT-1, EVENT-2

SAPHIRE VERSION 6.80 Fault Tree Model: TFT4–A

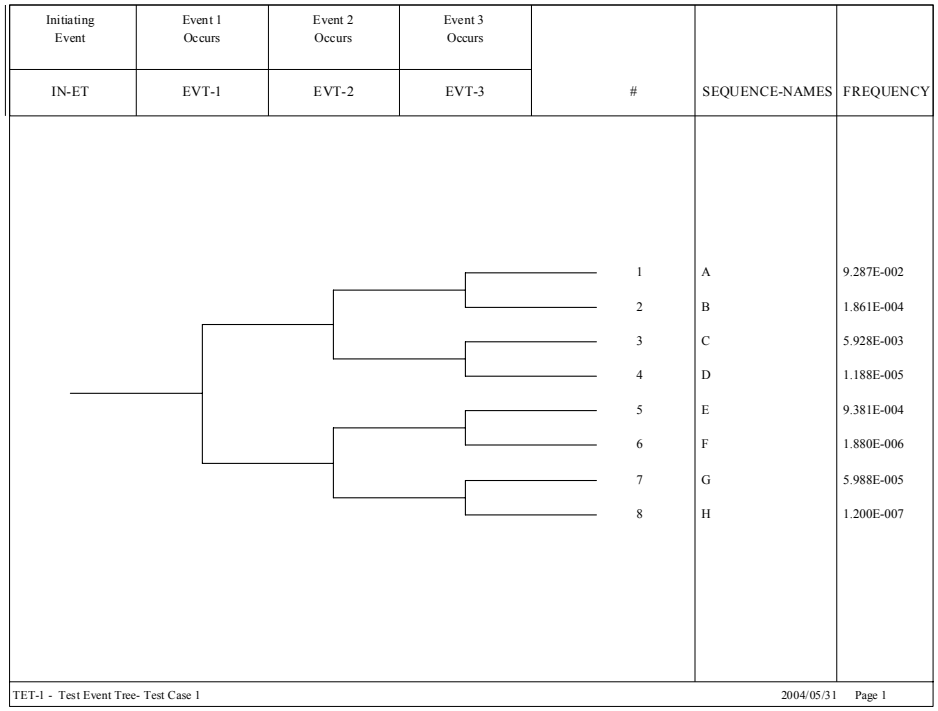


Sort/Slice Cut Set Report

Project-> TEST Fault Tree-> TFT4-A
Mincut Upper Bound -> 1.005E-004 This Partition -> 1.005E-004

Cut No.	% Total	% Cut Set	Frequency	Cut Sets
1	99.5	99.5	1.000E-004	EVENT-1
2	100.0	0.5	5.000E-007	EVENT-2, EVENT-5
3	100.0	0.0	1.600E-012	EVENT-3, EVENT-4, EVENT-5

SAPHIRE VERSION 6.80 EVENT TREE MODEL: TET–1



SEQUENCE SUMMARY REPORT

Project: TEST
Case:CURRENT
Analysis: RANDOM

Seq No	Event Tree Name Sequence Description	Sequence Name	MinCut Upper Bound
1	TET-1	1	9.287E-002
2	TET-1	2	1.861E-004
3	TET-1	3	5.928E-003
4	TET-1	4	1.188E-005
5	TET-1	5	9.381E-004
6	TET-1	6	1.880E-006
7	TET-1	7	5.988E-005
8	TET-1	8	1.200E-007
		TOTAL	1.000E-001