“Development Paper – Staggered Proof Testing Coefficients”

DISCLAIMER: Whilst every effort has been made to ensure the accuracy of the information contained in this document neither “The 61508 Association” nor its members will assume any liability for any use made thereof.
1 Contents

1 Contents ....................................................................................................................................... 2
2 Revision History .................................................................................................................................. 3
3 Introduction / Foreword .................................................................................................................. 4
4 Executive Summary ...................................................................................................................... 5
5 Terminology .................................................................................................................................. 6
6 Complex redundancy with staggered proof testing ................................................................. 8
7 Failure of 4 Items .......................................................................................................................... 8
8 Failure of N Items .......................................................................................................................... 10
9 Finding the products of singles, pairs, triples etc in a set ........................................................... 11
  9.1.1 Set of 3 ................................................................................................................................ 11
  9.1.2 Set of 4 ................................................................................................................................ 11
  9.1.3 Set of 5 ................................................................................................................................ 12
10 Finding sums of products by programming .............................................................................. 13
  10.1.1 The sum of pairs (selections of 2) ................................................................................ 13
  10.1.2 The sum of triples (selections of 3) ............................................................................... 14
  10.1.3 The sum of selections of N ........................................................................................... 14
11 Finding subsets of M from N .................................................................................................... 16
  11.1 Subsets of 4 in the range 1 to 7 ........................................................................................... 16
  11.2 Subsets of M in the range 1 to N .......................................................................................... 17
12 Conclusion .............................................................................................................................. 19
13 Existing and Emerging Standards ............................................................................................ 19
14 61508 Association Recommended Practices ........................................................................... 19
2 Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>28/01/2022</td>
<td>RM</td>
<td>Draft release for public comment.</td>
</tr>
</tbody>
</table>
3 Introduction / Foreword

This development paper is not yet complete or fully reviewed by members of The 61508 Association. This paper has been published to elicit further comment and input on the comments within the paper from both members of the Association and any other interested party from outside the Association. Please send all comments on this paper to the Association coordinator via the email: info@61508.org.

Message from the initial author:
This document is to support Effects of Proof Testing in the topic area of Staggered Proof Testing.

This is where we consider the failure of a subset with the understanding that the probability of failure depends on the relative positions in the testing cycle of the items of interest. The only way I have found is to consider all the possible cases and average them.

This results in a program which is run once to generate a set of coefficients for use in the Calculation Engine.

This document represents my notes on the algorithm development use before creating Excel VBA routines to generate the coefficients required by Staggered Proof Testing. It is nothing more than an aide memoire.

The final result is used in Fault Tolerant Systems which is the document which lies behind the so-called PFD Calculator which is an Excel based reliability assessment spreadsheet with a reliability calculator built in. The VBA used to develop the coefficients is contained within the same Excel file.

Ray Martin
4 Executive Summary

This paper is not yet complete; therefore, an executive summary is not yet required. This paper has been published in this form to elicit comments on the content.
5 Terminology

*f* General term for ‘fault tolerance’ – i.e. for simple redundancy, the number of failed devices a system can tolerate and still perform its function.

Note: *r* is the general term for the number of survivors required for a system to perform its function.

*F* Probability of failure (normally a function of time).

Note: this has the same meaning at *PFD* (probability of failure on demand).

*MT* Mission Time (for use with residual failures)

*MTBF* Mean time before failure. $MTBF = 1 / \lambda$ (for constant $\lambda$)

*MTTR* Mean time to restore.

*PFD* Probability of failure on demand.

Notes:
- this has the same meaning as *F* (probability of failure).
- This is sometimes used in the text as shorthand for *PFD*$_{AV}$.

*PFD$_{AV}$* Time average of *PFD*.

*PFD$_D$* PFD for diagnosed failures for single channel / device.

$$PFD_D = PFD_D^1 = ((1 - \beta_D) \lambda dd. MTTR)$$

*PFD$_R$* PFD for residual failures for single channel / device.

$$PFD_R = PFD_R^1 = \left(\frac{(1 - \beta_R) \lambda dd. MTTR}{2}\right)$$

*PFD$_U$* PFD for undiagnosed failures for single channel / device.

$$PFD_U = PFD_U^1 = \left(\frac{(1 - \beta_U) \lambda dd. MTTR}{2}\right)$$

*PFD$_D^K$* PFD for diagnosed failures for k channels / devices

$$PFD_D^K = (PFD_D)^k$$

*PFD$_R^K$* PFD for residual failures for k channels / devices

$$PFD_R^K \neq (PFD_R)^k \text{ due to test regime}$$

*PFD$_U^K$* PFD for undiagnosed failures for k channels / devices

$$PFD_U^K \neq (PFD_U)^k \text{ due to replacement regime}$$

*PFD$_N$* PFD rolled up for all failures for N channels / devices (including common causes)

*R* Probability of survival (normally a function of time).

*s* Used as a suffix to represent attributes of a system.

E.g. *F$_S$* is used to represent probability of system failure.

*T* Proof test interval.
\[ \beta \] Beta factor – general term for fraction of failures which affect all channels / devices.

\[ \beta_D \] Beta factor specific to diagnosed failures

\[ \beta_R \] Beta factor specific to residual failures

\[ \beta_U \] Beta factor specific to undiagnosed failures

\[ \lambda \] General term for underlying failure rate – a function of time that represents the failure rate ‘given that there is no current failure’. This paper assumes it is a constant in time.

Note: this is not the same as \( \dot{I}(t) \) (which is the failure rate not assuming current survival).

\[ \lambda_d \] General term for diagnosed failure rate – i.e. failure that is automatically revealed.

\[ \lambda_u \] General term for undiagnosed failure rate.

\[ \lambda_{dd} \] Dangerous diagnosed failure rate.

\[ \lambda_{dr} \] Dangerous residual failure rate – i.e. dangerous failure rate that is not automatically revealed or revealed by periodic proof test.

\[ \lambda_{du} \] Dangerous undiagnosed failure rate.
6 Complex redundancy with staggered proof testing

Here, we consider the case of complex redundancy where faults are undiagnosed (i.e. found only in proof testing) and where the system fails if M out of N items fail.

The purpose of this paper is to generate the algorithms that will generate a 10 by 10 table of numbers representing the test factors to be applied. The test factors are multiplying factors for M out of N failures where these factors are multiplied by \( \text{PFD}_{\text{1oo1}}^N \) to find the \( \text{PFD}_{\text{AV}} \).

The mathematics is dealt with in more detail in Staggered Proof Testing but the cases of 4 failures and M failures are repeated here for information.

7 Failure of 4 Items

Consider the following graph

The joint probability of failure is given by

\[
F(t) = \lambda t (\lambda t + (1 - a)\lambda T)(\lambda t + (1 - b)\lambda T)(\lambda t + (1 - c)\lambda T) \quad [0,aT]
\]

\[
F(t) = \lambda t (\lambda t + (-a)\lambda T)(\lambda t + (1 - b)\lambda T)(\lambda t + (1 - c)\lambda T) \quad [aT,bT]
\]

\[
F(t) = \lambda t (\lambda t + (-a)\lambda T)(\lambda t + (-b)\lambda T)(\lambda t + (1 - c)\lambda T) \quad [bT,cT]
\]

\[
F(t) = \lambda t (\lambda t + (-a)\lambda T)(\lambda t + (-b)\lambda T)(\lambda t + (-c)\lambda T) \quad [cT,T]
\]

We write this as:

\[
F(t) = x(x + A)(x + B)(x + C)
\]

Where

\[
x = \lambda t;
\]

\[
A = (1 - a)\lambda T \quad [t < aT]
\]

\[
A = (-a)\lambda T \quad [t \geq aT]
\]
B = (1 - b)λT [t < bT]
B = (-b)λT [t ≥ bT]
C = (1 - c)λT [t < cT]
C = (-c)λT [t ≥ cT]

Expanding the expression for F(t)

\[ F(t) = (x^2 + Ax)(x - B)(x - C) \]
\[ F(t) = (x^3 + (A + B)x^2)(x - C) \]
\[ F(t) = x^4 + (A + B + C)x^3 + (AB + AC + BC)x^2 + ABCx \]

Note: The coefficients for powers of x (other than the first) are:
- the sum of all the solos, then
- the sum of all the pairs, then
- the sum of all the triples
This pattern is repeated for greater powers.

So, for 4 failures with staggered proof testing:

\[ F(t) = λ^4 t^4 + (A + B + C)λ^3 t^3 + (AB + AC + BC)λ^2 t^2 + ABCλt \]

Where
\[ A = (1 - a)λT [t < aT] \]
\[ A = (-a)λT [t ≥ aT] \]
\[ B = (1 - b) \lambda T \ [t < bT] \]
\[ B = (-b) \lambda T \ [t \geq bT] \]

\[ C = (1 - c) \lambda T \ [t < cT] \]
\[ C = (-c) \lambda T \ [t \geq cT] \]

We could simplify this whole expression by replacing as follows:
\[ A' = A \lambda T \]
\[ B' = B \lambda T \]
\[ C' = C \lambda T \]

Then, for 4 failures with staggered proof testing:

\[
F(t) = \lambda^4 t^4 + (A + B + C) \lambda^3 t^3 + (AB + AC + BC) \lambda^2 t^2 + ABC \lambda t
\]

\[
F_{AV} = \lambda^4 T^4 \left[ \frac{x^5}{5} + (A' + B' + C') \frac{x^4}{4} + (A'B' + A'C + B'C') \frac{x^3}{3} + A'B'C' \frac{x^2}{2} \right]
\]

\[
+ \lambda^4 T^4 \left[ \frac{x^5}{5} + (A' + B' + C') \frac{x^4}{4} + (A'B' + A'C + B'C') \frac{x^3}{3} + A'B'C' \frac{x^2}{2} \right]
\]

\[
+ \lambda^4 T^4 \left[ \frac{x^5}{5} + (A' + B' + C') \frac{x^4}{4} + (A'B' + A'C + B'C') \frac{x^3}{3} + A'B'C' \frac{x^2}{2} \right]
\]

\[
+ \lambda^4 T^4 \left[ \frac{x^5}{5} + (A' + B' + C') \frac{x^4}{4} + (A'B' + A'C + B'C') \frac{x^3}{3} + A'B'C' \frac{x^2}{2} \right]
\]

Where
\[ A' = (1 - a) \ [x < a] \]
\[ A' = (-a) \ [x \geq a] \]
\[ B' = (1 - b) \ [x < b] \]
\[ B' = (-b) \ [x \geq b] \]
\[ C' = (1 - c) \ [x < c] \]
\[ C' = (-c) \ [x \geq c] \]

### 8 Failure of N Items

It is possible to expand and find the general case from the above.
9 Finding the products of singles, pairs, triples etc in a set

To be able to evaluate the above, we need to be able to find the sum of the products of pairs, triples, quads etc.

9.1.1 Set of 3

Assume we have a vector of 3 elements: \( x_1, x_2 \) and \( x_3 \)

The sum of the singles is: \( x_1 + x_2 + x_3 \)

The sum of pairs is: \( x_1x_2 + x_1x_3 + x_2x_3 \)

The sum of triples is: \( x_1x_2x_3 \)

9.1.2 Set of 4

The sum of the singles is: \( x_1 + x_2 + x_3 + x_4 \)

The sum of pair products is: \( x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \)

The sum of triple products is: \( x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \)

The sum of quad products is: \( x_1x_2x_3x_4 \)
The sum of the singles is: \( x_1 + x_2 + x_3 + x_4 \)
The sum of pairs is: \( x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \)
This is \( x_1 \) times (sum of singles in range \( x_2 \) to \( x_4 \)) + \( x_2 \) times (sum of singles in range \( x_3 \) to \( x_4 \)) + \( x_3 \) times (sum of singles in range \( x_4 \) to \( x_4 \))
The sum of triples is: \( x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \)
This is \( x_1 \) times (sum of pairs in range \( x_2 \) to \( x_4 \)) + \( x_2 \) times (sum of pairs in range \( x_3 \) to \( x_4 \))
The sum of quads is: \( x_1x_2x_3x_4 \)
This is \( x_1 \) times (sum of triples in range \( x_2 \) to \( x_4 \))

9.1.3 Set of 5
The sum of the singles is: \( x_1 + x_2 + x_3 + x_4 + x_5 \)

The sum of pair products is:
\( x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_3 + x_2x_4 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5 \)
This is \( x_1 \) times (sum of singles in range \( x_2 \) to \( x_5 \)) + \( x_2 \) times (sum of singles in range \( x_3 \) to \( x_5 \)) + \( x_3 \) times (sum of singles in range \( x_4 \) to \( x_5 \)) + \( x_4 \) times (sum of singles in range \( x_5 \) to \( x_5 \))

The sum of triple products is:
\( x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 \)
This is \( x_1 \) times (sum of pairs in range \( x_2 \) to \( x_5 \)) + \( x_2 \) times (sum of pairs in range \( x_3 \) to \( x_5 \)) + \( x_3 \) times (sum of pairs in range \( x_4 \) to \( x_5 \))

The sum of quad products is:
\( x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5 \)
This is \( x_1 \) times (sum of triples in range \( x_2 \) to \( x_5 \)) + \( x_2 \) times (sum of triples in range \( x_3 \) to \( x_5 \))

The sum of the quint is: \( x_1x_2x_3x_4x_5 \)
This is \( x_1 \) times (sum of quads in range \( x_2 \) to \( x_5 \))

Note, the above in blue text can all be found algorithmically.
The sum of the singles is: \( x_1 + x_2 + x_3 + x_4 + x_5 \)

The sum of pair products is:
\[
x_1(x_2 + x_3 + x_4 + x_5) + x_2(x_3 + x_4 + x_5) + x_3(x_4 + x_5) + x_4(x_5)
\]

The sum of triple products is:
\[
x_1x_2(x_3 + x_4 + x_5) + x_1x_3(x_4 + x_5) + x_1x_4(x_5) + x_2x_3(x_4 + x_5) + x_2x_4(x_5) + x_3x_4(x_5)
\]

The sum of quad products is:
\[
x_1x_2x_3(x_4 + x_5) + x_1x_2x_4(x_5) + x_2x_3x_4(x_5)
\]

The sum of quint products is: \( x_1x_2x_3x_4(x_5) \)

### 10 Finding sums of products by programming

#### 10.1.1 The sum of pairs (selections of 2)

To find sums of pairs (selections of 2) in range \( x_K \) to \( x_M \)

For example: \( x_4 \) to \( x_7 \)

\( K=4, M=7, N=2 \)

The sum is:
\[
x_4x_5 + x_4x_6 + x_4x_7 + x_5x_6 + x_5x_7 + x_6x_7
\]

Written another way, the sum is:
\[
x_4(x_5+x_6+x_7) + x_5(x_6+x_7) + x_6(x_7)
\]

i.e.:
\[ x_4 \text{ times (sum of singles in range } x_5 \text{ to } x_7) \]
\[ + x_5 \text{ times (sum of singles in range } x_6 \text{ to } x_7) \]
\[ + x_6 \text{ times (sum of singles in range } x_7 \text{ to } x_7) \]

i.e.
\[ x_K \text{ times (sum of singles in range } x_{K+1} \text{ to } x_M) \]
\[ + x_{K+1} \text{ times (sum of singles in range } x_{K+2} \text{ to } x_M) \]
\[ + \ldots \]
\[ + x_{M-1} \text{ times (sum of singles in range } x_M \text{ to } x_M) \]

To generate this in a program:

\[ \text{Sum} = 0 \]
\[ \text{For } i = K \text{ to } M-1 \]
\[ \quad \text{For } j = K+1 \text{ to } M \]
\[ \quad \text{Product} = x_i \cdot x_j \]
\[ \quad \text{Sum} = \text{Sum} + \text{Product} \]
\[ \quad \text{Next } j \]
\[ \text{Next } i \]

### 10.1.2 The sum of triples (selections of 3)

To find sums of triples (selections of 3) in range \(x_K \text{ to } x_M\)

For example: \(x_5 \text{ to } x_8\)
\(K=5, \ M=9, \ N=3\)

\[ \text{Sum is} \]
\[ x_5 \text{ times}(x_6 \cdot x_7 + x_6 \cdot x_8 + x_6 \cdot x_9 + x_7 \cdot x_8 + x_7 \cdot x_9 + x_8 \cdot x_9) + x_6 \text{ times}(x_7 \cdot x_8 + x_7 \cdot x_9 + x_8 \cdot x_9) + x_7 \text{ times}(x_8 \cdot x_9) \]
i.e.:
\[ x_5 \text{ times (sum of pairs in range } x_6 \text{ to } x_9) \]
\[ + x_6 \text{ times (sum of pairs in range } x_7 \text{ to } x_9) \]
\[ + x_7 \text{ times (sum of pairs in range } x_8 \text{ to } x_9) \]

### 10.1.3 The sum of selections of \(N\)
To generate this algorithmically:

For i1 = K To M + 1 - N
    product(1) = x(i1)
    If N > 1 Then
        For i2 = i1 + 1 To M + 2 - N
            product(2) = product(1) * x(i2)
            If N > 2 Then
                For i3 = i2 + 1 To M + 3 - N
                    product(3) = product(2) * x(i3)
                    sum = sum + product(3)
                Next i3
            Else
                sum = sum + product(2)
            End If
        Next i2
    Else
        sum = sum + product(1)
    End If
Next i1

Where the nesting would be continued to the maximum level required.
11 Finding subsets of M from N

In the following, we are finding all the possible subsets of M integers in the range 1 to N.

What seems an easy enough task when \( N = 3 \) and \( M = 2 \), soon becomes very difficult when these values are higher. A system is required.

11.1 Subsets of 4 in the range 1 to 7

Below is a systematic progression for 4 integers in the range 1 to 7.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
We started with 1234 and then indexed the 4th column until it reached the limit 7.
We then indexed the 3rd column (3 to 4) and started by resetting the 4th column to one higher.
We repeated the indexing of the 3rd column until it could go no higher, i.e. the 4th column started at 7.
We then indexed the second column (2 to 3) and started by resetting the 3rd and 4th columns in ascending values.
We repeated the above process until indexing the second column meant that it could go no higher because all columns to the right were packed and so on.

11.2 Subsets of M in the range 1 to N
It is possible to generalise the above process
Set the first column to 1 and then reset all columns to the right.
Note that resetting all columns to the right means make them ascend in turn from the current column's contents.
Increment the Nth column until the contents is equal to M
Index the (N-1)th column and ‘reset’ column to right
Index from the Nth column until the contents is equal to M.
Index the (N-1)th column again and repeat the above process until the contents is equal to M-1.
Then start indexing the (N-2)th column and repeat.
On the next page is a routine which generates these combinations.
For N = 1 To M
  count = 0
  K = N - 1
For i1 = 1 To M - K
  S(1) = i1
  If N > 1 Then
    For i2 = i1 + 1 To M - K + 1
      S(2) = i2
      If N > 2 Then
        For i3 = i2 + 1 To M - K + 2
          S(3) = i3
          If N > 3 Then
            For i4 = i3 + 1 To M - K + 3
              S(4) = i4
              If N > 4 Then
                Else
                  count = count + 1
                  Write (S)
              End If
            End For
          End If
        End For
      End If
    End For
  End If
End For
Next i3
Else
  count = count + 1
  Write (S)
End If
Next i2
Else
    count = count + 1
    Write (S)
End If
Next i1
Next N

12 Conclusion
This paper is not yet complete; therefore, a conclusion is not yet required. This paper has been published in this form to elicit comments on the content.

13 Existing and Emerging Standards

14 61508 Association Recommended Practices
This document sets out to describe current best practices in Reliability and Availability for functional safety systems, but does not seek to prescribe specific measures, since these will depend on the application, and any existing constraints of the installation.

This paper is not yet completed, it has been published solely to elicit comment and encourage input to further develop the technical content.

DISCLAIMER: Whilst every effort has been made to ensure the accuracy of the information contained in this document neither “The 61508 Association” nor its members will assume any liability for any use made thereof.