The CMOS ASIC Lifetime Model (CALM) Cockpit API and Specification

May 30, 2014

Brian W. Wood, Jon V. Osborn, John R. Scarpulla, Vishnu Jyothindran
Modeling and Simulation Department

AEROSPACE
Assuring Space Mission Success

Abstract

The Aerospace Corporation, and the broader high reliability electronics community, are developing a growing suite of CMOS ASIC physics-of-failure reliability models.. A capability is needed to execute these models both independently and as integrated components in a system-of-systems model for full-chip lifetime reliability modeling. Furthermore, third parties are also developing PoF models which are to be shared among the high reliability electronics community and must be interoperable in a plug-and-play environment.

This document describes a specification and Application Programmer's Interface (API) of a framework that allows developers to produce models that execute both standalone and in concert with other, independently developed models. It also describes the CMOS ASIC Lifetime Model (CALM) Cockpit, an Excel workbook which allows users to add/remove individual models from a study, coordinates their execution, and consolidates their output.

Contents

Figures

1. Introduction

The Aerospace Corporation, and the broader high reliability electronics community, are developing a growing suite of CMOS ASIC physics-of-failure reliability models.. A capability is needed to execute these models both independently and as integrated components in a system-of-systems model for full-chip lifetime reliability modeling. Furthermore, third parties are also developing PoF models which are to be shared among the high reliability electronics community and must be interoperable in a plug-and-play environment.

This document describes a specification and Application Programmer's Interface (API) of a framework that allows developers to produce models that execute both standalone and in concert with other, independently developed models. It also describes the CMOS ASIC Lifetime Model (CALM) Cockpit, an Excel workbook which allows users to add/remove individual models from a study, coordinates their execution, and consolidates their output.

1.1 Overview

When modeling CMOS full-chip reliability, we need to be able to represent individual failure mechanisms in fine detail as well as t an entire system of all failure mechanisms. That is, we must be able to execute and develop individual reliability subsystem models independently, in addition to executing them in concert.

The CALM Cockpit provides a specification for building individual subsystem reliability models that can also be coupled and executed as a collection of models to analyze overall chip lifetime reliability.

In this document we refer to individual subsystem models as federates. A collection of federates being run in concert is a federation. The execution of a federation is initiated, managed, and controlled by the CALM Cockpit.

1.2 Modeling the Subsystems – Federates and Federations

A full-chip reliability model requires the modeling of several subsystems (federates). Each federate has its own behavior and failure modes and requires specific skillsets from the model developer.

Each federate must be able to be developed and used independently. Each federate will have its own specific algorithms to implement and output requirements. Each federate has significant value as a standalone model.

But each federate must also be able to perform in a federation. As such, it must also be capable of:

- Executing when instructed externally (by the CALM Cockpit)
- Receiving its inputs from the CALM Cockpit
- Sending high-level output to the CALM Cockpit

In order to support both standalone and federated execution, this document describes a specification and API that supports building federates that are unrestricted when executed in standalone mode, but also provide the outputs required for studying overall reliability of a federation.

Figure 1 below illustrates how a federate executes as a standalone model:

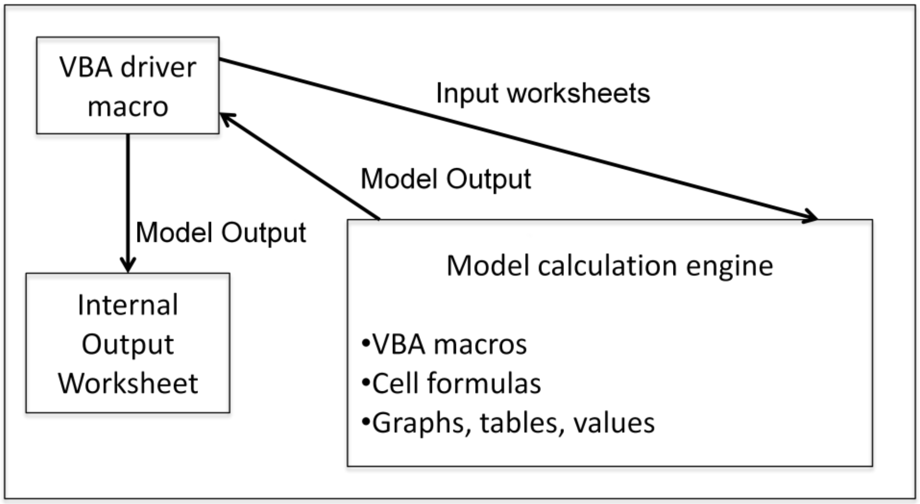


Figure 1 Federate as a Standalone Model

Figure 2 below illustrates a federate running under control of the CALM Cockpit:

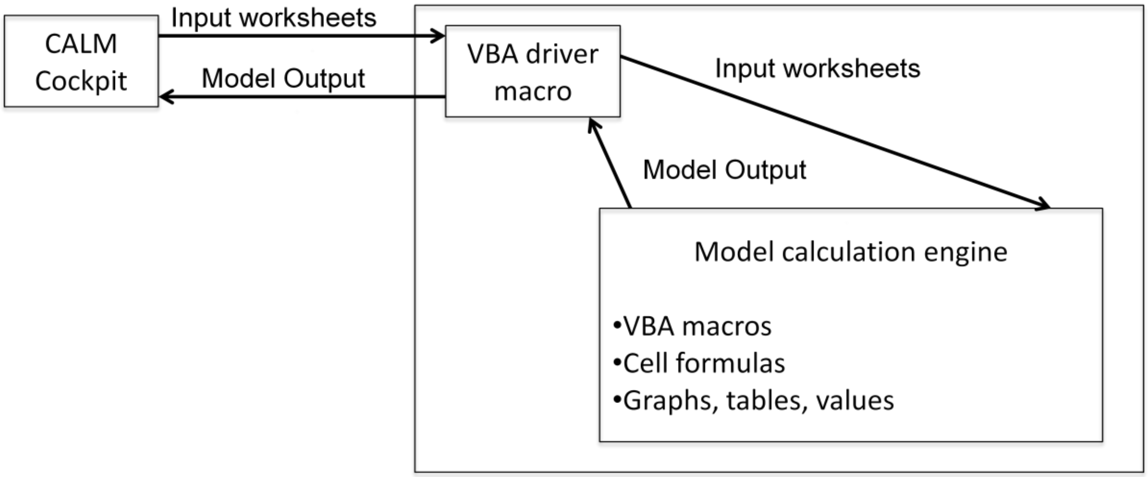


Figure 2: Federate Controlled by the CALM Cockpit

Finally, Figure 3 below shows the CALM Cockpit controlling a federation:

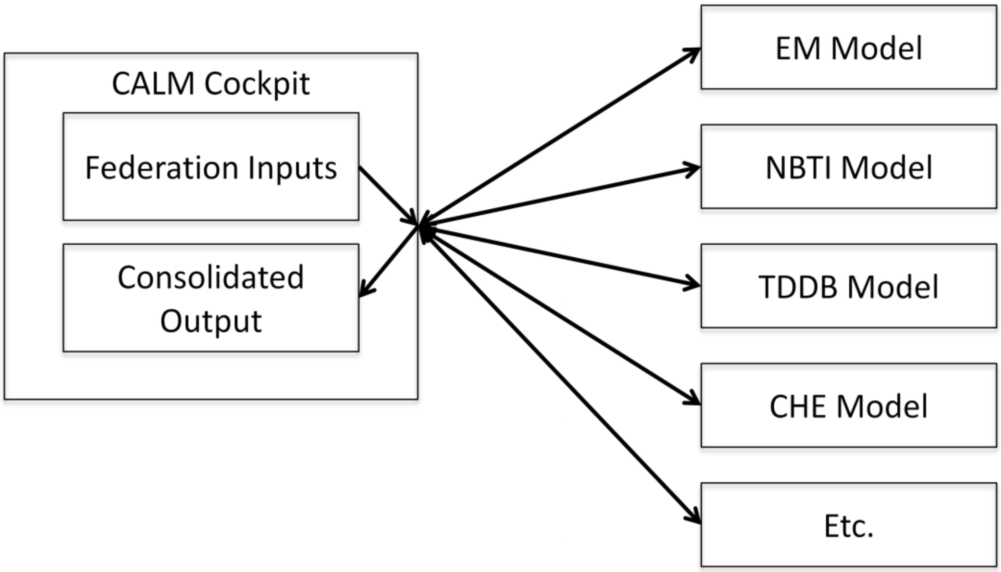


Figure 3: The CALM Cockpit Controlling a Federation

1.3 Controlling the Federation – The CALM Cockpit

This document also describes the plan for building the CALM Cockpit. The CALM Cockpit is an Excel workbook that controls a federation of subsystem models. The CALM Cockpit provides the following services of the analyst:

- Define a federation as a collection of federates (subsystem models)
- Provide inputs for each federate in the federation
- Provide inputs shared by all federates in the federation (i.e., timesteps, temperature data, on/off status of the chip, etc.)
- Execute each federate in the federation
- Collect time-stepped reliability output from each federate
- Consolidate federate outputs into overall reliability metrics, generate graphs and charts, etc.

2. Federate Specification

The CALM federate specification is simple: your federate must receive its inputs in the form of Excel worksheets, it must provide an externally callable VBA macro for federate execution, and it must return output consisting of three columns of data.

One of the input worksheets contains global inputs. Global inputs are data that are shared by all federates in the federation, such as time step values. Other inputs worksheets may contain inputs specific to the federate.

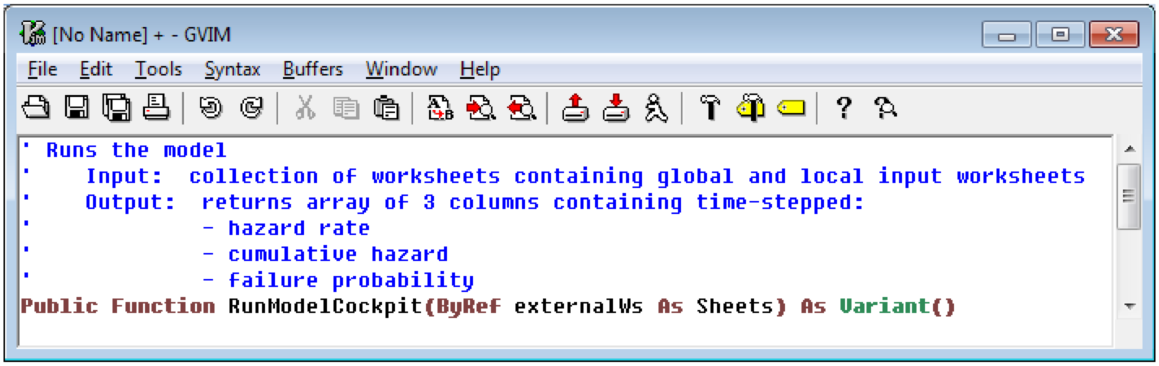
Output consists of time-stepped reliability measures, including the Hazard Rate (FIT) Cumulative Hazard, and Failure Probability.

The specification spells out how your federate must be designed to execute its model whether these inputs come from worksheets in its workbook (standalone mode) or from worksheets passed in from the CALM Cockpit workbook (federated mode).

2.1 The Federate API in a Nutshell

There are 3 essential elements to the CALM Cockpit federate API:

1. All user-editable inputs specific to the federate are contained in named ranges in Excel worksheets.
2. All time-stepped data shared by federates in a federation are contained in named ranges in a single Excel worksheet.
3. The federate must provide the following VBA macro to be called by the CALM Cockpit:



```
' Runs the model
'     Input:  collection of worksheets containing global and local input worksheets
'     Output:  returns array of 3 columns containing time-stepped:
'               - hazard rate
'               - cumulative hazard
'               - failure probability
Public Function RunModelCockpit(ByRef externalWs As Sheets) As Variant()
```

Figure 4: Federate API in a Nutshell

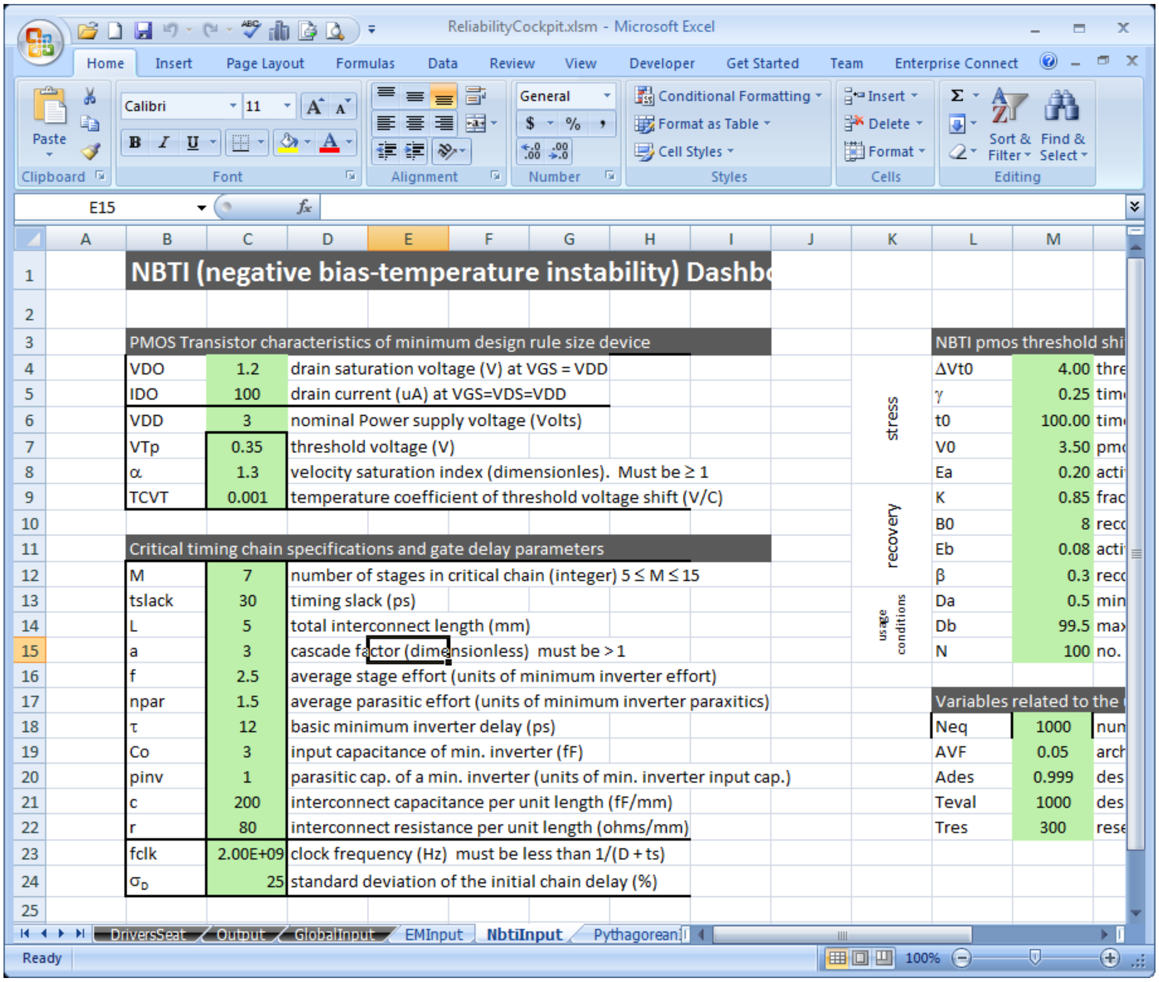
This document describes the details of this API, as well as a specification for implementing models that support the API.

2.2 Using Named Ranges for Input Values

Since the federate's input consists of worksheets, the federate must reference Excel cell values by range rather than cell address. A named range is one or more cells that have been named within Excel. Named ranges are far more robust than cell addresses because they can withstand being cut/copied/pasted without losing referential integrity. Also, it frees the user to arrange input sheets as they see fit – they know the value for, say, impedance is in a range of a single cell named ‘Impedance’, and they do not care at which row/column address that cell resides.

2.3 Local Inputs

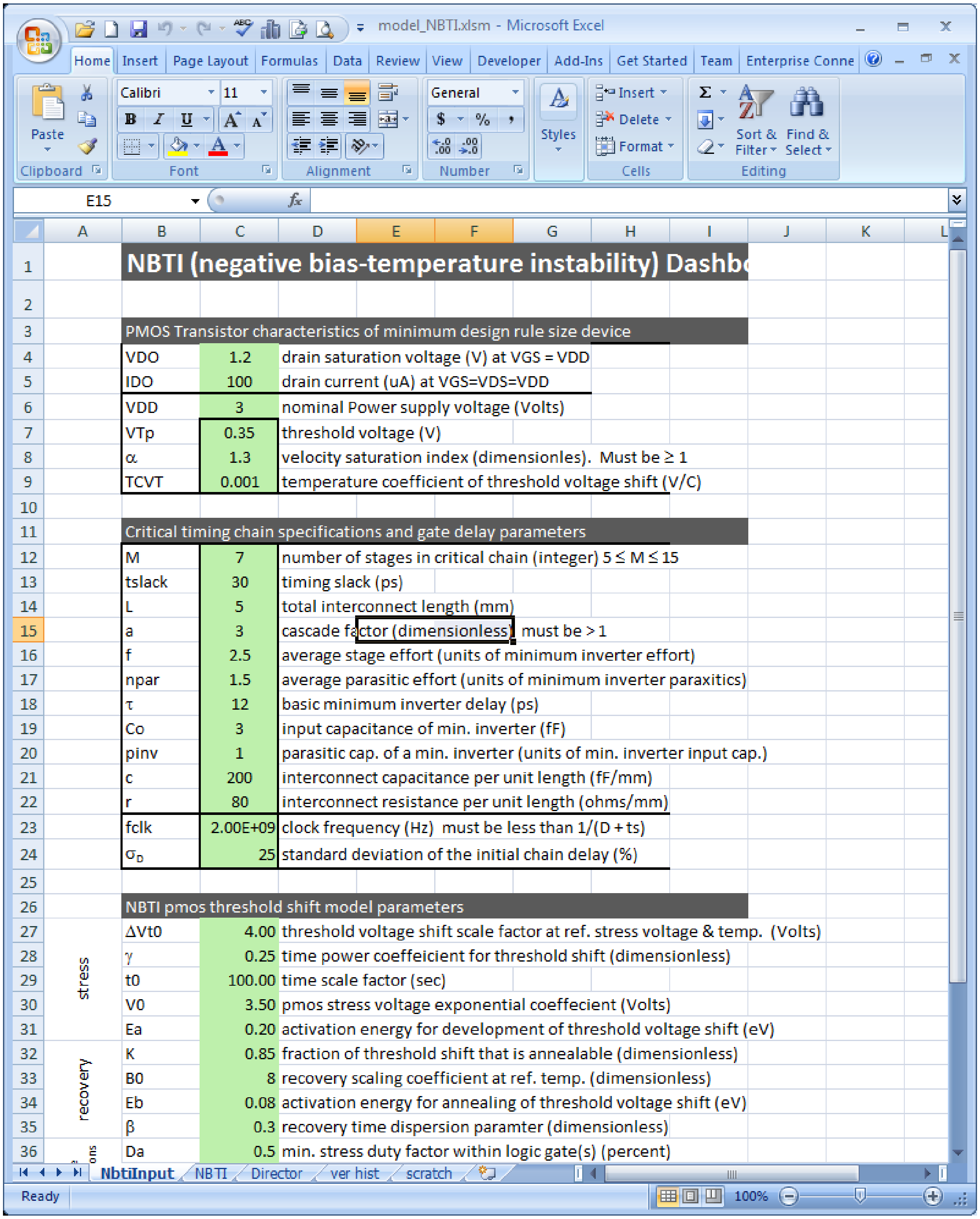
Each federate may require one (or more) worksheets with particular names. For example, the NBTI workbook requires a worksheet named ‘NbtiInput’. This worksheet contains all of the input values that someone running a federation is able to change. The NBTI input worksheet in the CALM Cockpit looks like:



	B	C	D	K	L	M	
1	NBTI (negative bias-temperature instability) Dashbo						
2							
3	PMOS Transistor characteristics of minimum design rule size device				NBTI pmos threshold shi		
4	VDO	1.2	drain saturation voltage (V) at VGS = VDD	stress	ΔVt0	4.00	thre
5	IDO	100	drain current (uA) at VGS=VDS=VDD		γ	0.25	tim
6	VDD	3	nominal Power supply voltage (Volts)		t0	100.00	tim
7	VTp	0.35	threshold voltage (V)		V0	3.50	pm
8	α	1.3	velocity saturation index (dimensionles). Must be ≥ 1		Ea	0.20	acti
9	TCVT	0.001	temperature coefficient of threshold voltage shift (V/C)	recovery	K	0.85	frac
10					B0	8	rec
11	Critical timing chain specifications and gate delay parameters				Eb	0.08	acti
12	M	7	number of stages in critical chain (integer) 5 ≤ M ≤ 15		β	0.3	rec
13	tslack	30	timing slack (ps)	usage conditions	Da	0.5	min
14	L	5	total interconnect length (mm)		Db	99.5	max
15	a	3	cascade factor (dimensionless) must be > 1		N	100	no.
16	f	2.5	average stage effort (units of minimum inverter effort)				
17	npar	1.5	average parasitic effort (units of minimum inverter paraxitics)		Variables related to the		
18	τ	12	basic minimum inverter delay (ps)		Neq	1000	nun
19	Co	3	input capacitance of min. inverter (fF)		AVF	0.05	arch
20	pinv	1	parasitic cap. of a min. inverter (units of min. inverter input cap.)		Ades	0.999	des
21	c	200	interconnect capacitance per unit length (fF/mm)		Teval	1000	des
22	r	80	interconnect resistance per unit length (ohms/mm)		Tres	300	rese
23	fclk	2.00E+09	clock frequency (Hz) must be less than 1/(D + ts)				
24	σ_D	25	standard deviation of the initial chain delay (%)				
25							

Figure 5: NBTI Inputs -- CALM Cockpit

Each value in the above worksheet resides in a cell that is assigned to a named range. The figure below shows the NBTI input sheet for the standalone workbook:



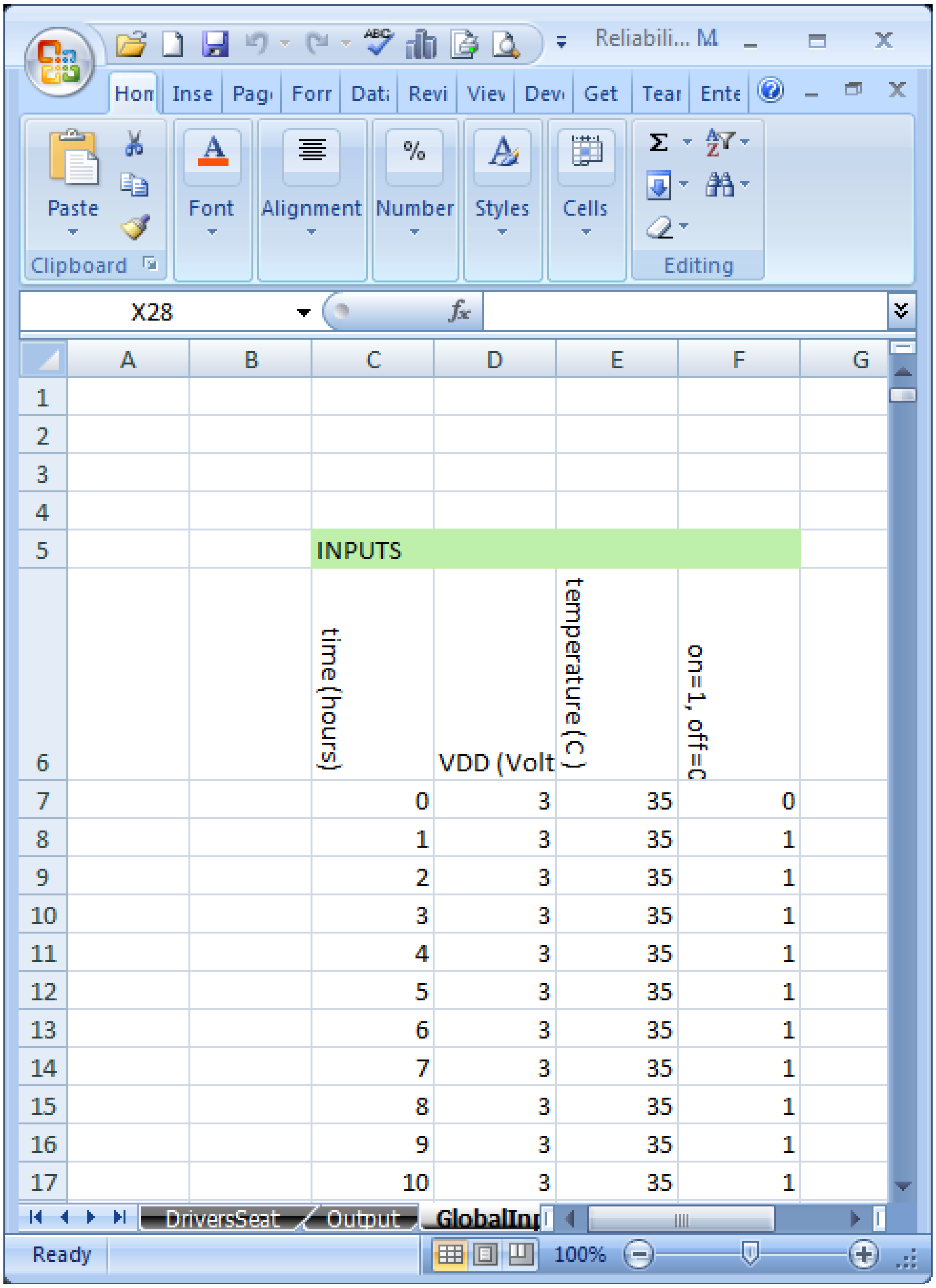
	A	B	C	D
1		NBTI (negative bias-temperature instability) Dashb		
2				
3		PMOS Transistor characteristics of minimum design rule size device		
4		VDO	1.2	drain saturation voltage (V) at VGS = VDD
5		IDO	100	drain current (uA) at VGS=VDS=VDD
6		VDD	3	nominal Power supply voltage (Volts)
7		VTp	0.35	threshold voltage (V)
8		α	1.3	velocity saturation index (dimensionles). Must be ≥ 1
9		TCVT	0.001	temperature coefficient of threshold voltage shift (V/C)
10				
11		Critical timing chain specifications and gate delay parameters		
12		M	7	number of stages in critical chain (integer) 5 ≤ M ≤ 15
13		tslack	30	timing slack (ps)
14		L	5	total interconnect length (mm)
15		a	3	cascade factor (dimensionless) must be > 1
16		f	2.5	average stage effort (units of minimum inverter effort)
17		npar	1.5	average parasitic effort (units of minimum inverter paraxitics)
18		τ	12	basic minimum inverter delay (ps)
19		Co	3	input capacitance of min. inverter (fF)
20		pinv	1	parasitic cap. of a min. inverter (units of min. inverter input cap.)
21		c	200	interconnect capacitance per unit length (fF/mm)
22		r	80	interconnect resistance per unit length (ohms/mm)
23		fclk	2.00E+09	clock frequency (Hz) must be less than 1/(D + ts)
24		σ_D	25	standard deviation of the initial chain delay (%)
25				
26		NBTI pmos threshold shift model parameters		
27	stress	ΔVt0	4.00	threshold voltage shift scale factor at ref. stress voltage & temp. (Volts)
28		γ	0.25	time power coeffeicient for threshold shift (dimensionless)
29		t0	100.00	time scale factor (sec)
30		V0	3.50	pmos stress voltage exponential coeffecient (Volts)
31		Ea	0.20	activation energy for development of threshold voltage shift (eV)
32	recovery	K	0.85	fraction of threshold shift that is annealable (dimensionless)
33		B0	8	recovery scaling coefficient at ref. temp. (dimensionless)
34		Eb	0.08	activation energy for annealing of threshold voltage shift (eV)
35		β	0.3	recovery time dispersion paramter (dimensionless)
36	ons	Da	0.5	min. stress duty factor within logic gate(s) (percent)

Figure 6: NBTI Inputs -- CALM Cockpit – Standalone

Note that these worksheets are laid out differently. The positions of the values are not important because each value is referenced by its named range. For example, the value for t0 is at address M6 in the CALM Cockpit, but in the standalone worksheet it is in cell C29. In both workbooks, these cells have been named “t0”.

2.4 Global Inputs

Each CALM-compatible federate takes global input from a single worksheet named ‘GlobalInput’. Currently, this worksheet contains:



	A	B	C	D	E	F	G
1							
2							
3							
4							
5			INPUTS				
6			time (hours)	VDD (Volt	temperature (C)	on=1, off=C	
7			0	3	35	0	
8			1	3	35	1	
9			2	3	35	1	
10			3	3	35	1	
11			4	3	35	1	
12			5	3	35	1	
13			6	3	35	1	
14			7	3	35	1	
15			8	3	35	1	
16			9	3	35	1	
17			10	3	35	1	

Figure 7: Global Inputs

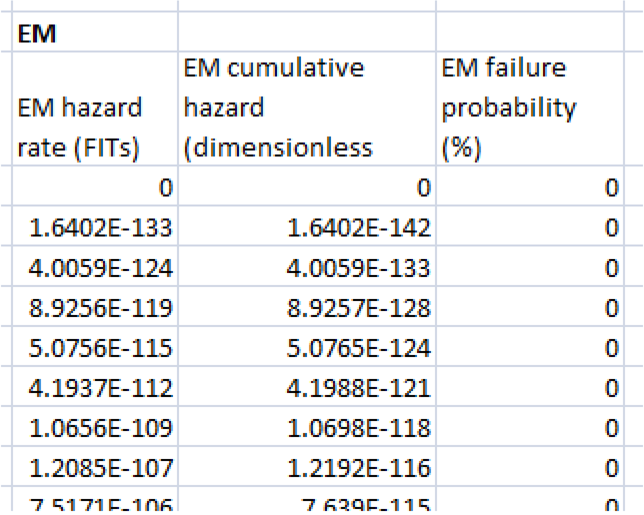
This worksheet contains four named ranges:

- Cell C6 is named “time” – it heads the column containing timesteps for the model
- Cell D6 is named “vdd” - it heads the column containing supply voltage for the mode
- Cell E6 is named “temperature” – it heads the column containing the temperature for each timestep
- Cell F6 is named “onOff” – it heads the column containing the on/off state of the chip for each timestep

The figure above shows the CALM Cockpit’s GlobalInput worksheet. When running standalone, the model will contain identically-named columns somewhere in the workbook.

2.5 Output

Federate output consists of an array of time-stepped results that are returned from the main execution function. As an example for the electromigration federate the model outputs are shown below:



EM		
EM hazard rate (FITs)	EM cumulative hazard (dimensionless	EM failure probability (%)
0	0	0
1.6402E-133	1.6402E-142	0
4.0059E-124	4.0059E-133	0
8.9256E-119	8.9257E-128	0
5.0756E-115	5.0765E-124	0
4.1937E-112	4.1988E-121	0
1.0656E-109	1.0698E-118	0
1.2085E-107	1.2192E-116	0
7.5171E-106	7.639E-115	0

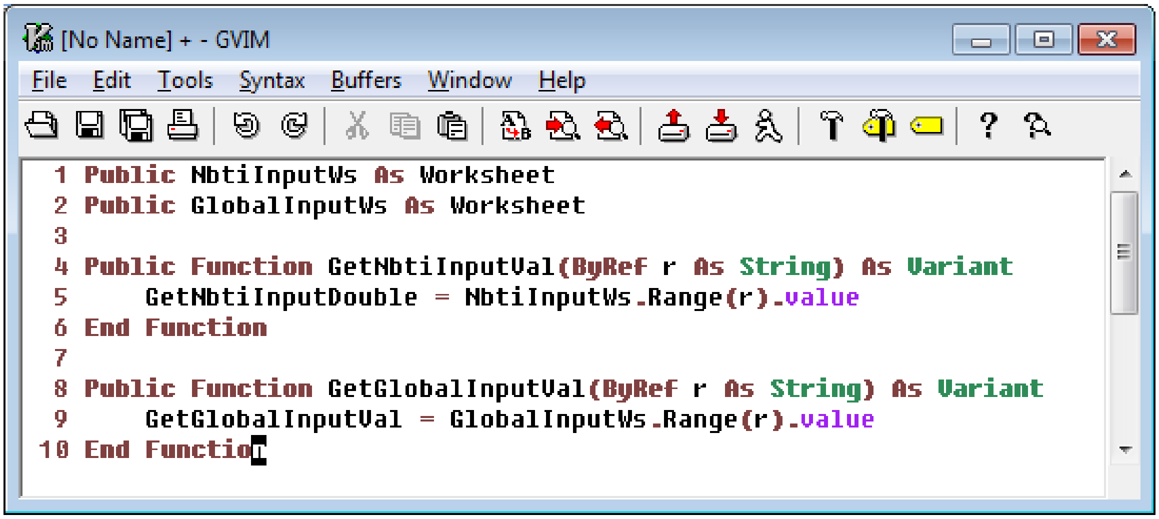
Figure 8: Federate Output

The array must have 2 dimensions, and the first row contains column headers. Note that the headers should include the federate’s name, as these headers are used in labeling the CALM Cockpit’s graphs.

Note that a federate may also produce all manner of internal output in the form of values, graphs, etc. to support detailed analysis when running in standalone mode. See section 2.8 for a discussion of internal output.

2.6 Setting VBA Variables

The federate workbook defines worksheets as global variables to be used to set input values throughout the workbook. For example, the NBTI model defines two worksheets as global variables:



```
 1 Public NbtiInputWs As Worksheet
 2 Public GlobalInputWs As Worksheet
 3
 4 Public Function GetNbtiInputVal(ByRef r As String) As Variant
 5     GetNbtiInputDouble = NbtiInputWs.Range(r).value
 6 End Function
 7
 8 Public Function GetGlobalInputVal(ByRef r As String) As Variant
 9     GetGlobalInputVal = GlobalInputWs.Range(r).value
10 End Function
```

Figure 9: NBTI Glogal Variables

- Line 1: *NbtiInputWs* is a worksheet containing all user-editable input values that are specific to the NBTI model
- Line 2: *GlobalInputWs* is a worksheet containing global inputs such as timestep values.
- Lines 4-6: *GetNbtiInputVal()* is a function that may be called from worksheet cells to retrieve input values. For example, the formula *'=GetNbtiInputVal("t0")'* retrieves the value of *t0* for use in cell calculations. VBA code may also use this function, or directly reference the value with the built-in VBA function *'NbtiInputWs.Range("t0").Value'*
- Lines 8-10: *GetGlobalInputVal()* retrieves global input values

NOTE: The worksheets containing these inputs may be local to the model workbook, or passed in from another workbook (the CALM Cockpit). The model's computations and algorithms will not know or care where these input worksheets came from.

2.7 Model Control and Execution with Excel Events

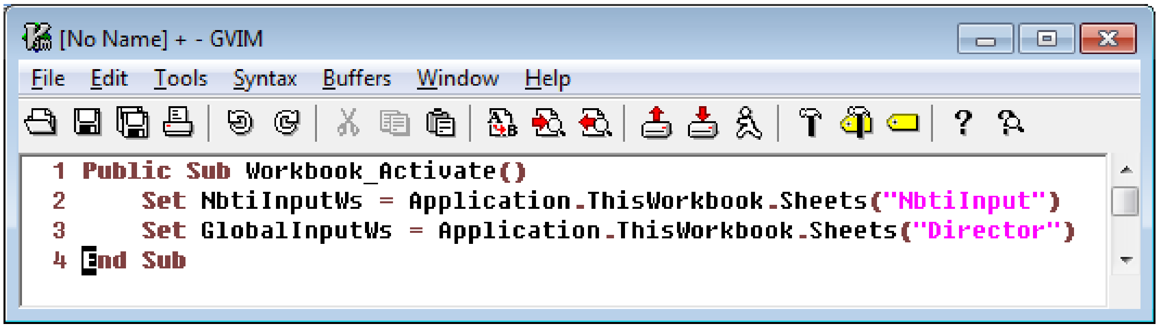
An Excel event is an action that causes VBA functions to be executed. For example, the workbook developer may create an event by mapping a button click to a VBA function call.

Excel offers a limited set of built-in events. These built-in events allow the developer to insert VBA code when particular Excel events occur, such as when the workbook is opened, when the user switches worksheets, etc.

In this API, two events are used to control the initialization and execution of the federate; Workbook Open Event and Run Model Event, both of which as discussed below.

2.7.1 Workbook Open Event

When the workbook is opened, the built-in *Workbook_Activate()* macro is called automatically by Excel. This event sets the input sheet variables to worksheets contained in the workbook. This sets the workbook's default mode to standalone:



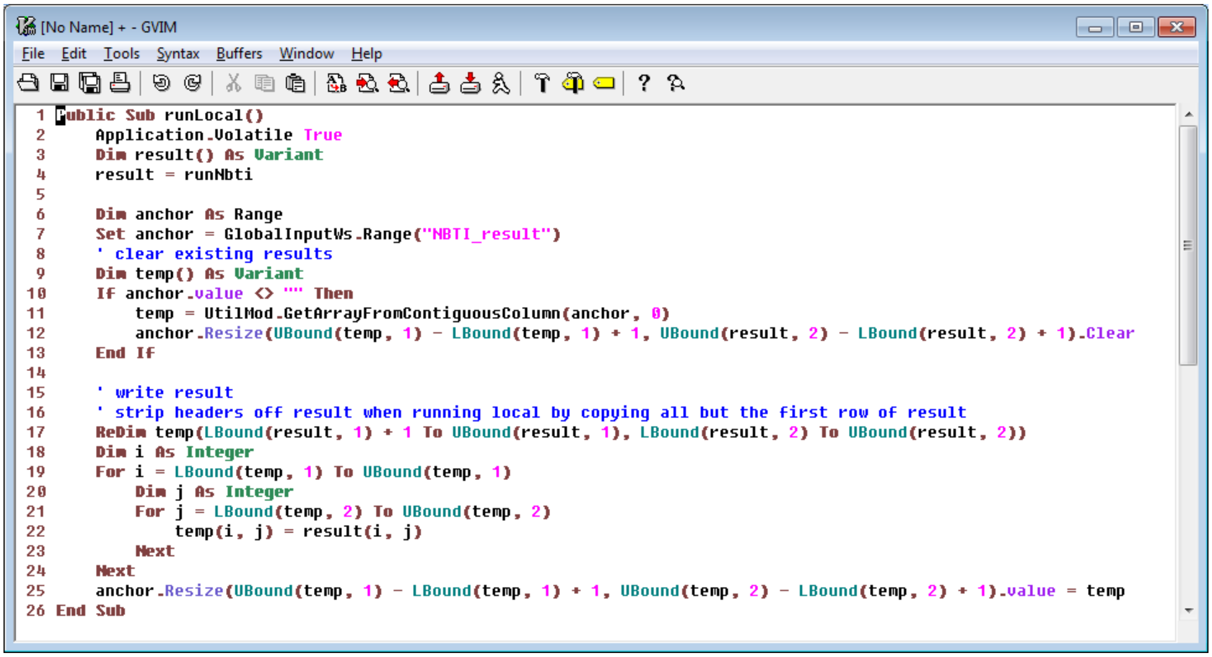
```
1 Public Sub Workbook_Activate()
2     Set NbtiInputWs = Application.ThisWorkbook.Sheets("NbtiInput")
3     Set GlobalInputWs = Application.ThisWorkbook.Sheets("Director")
4 End Sub
```

Figure 10: Open Workbook Event

Lines 2-3 above set the local and global input worksheets to sheets that are contained in the workbook. This guarantees that all formulas and VBA code that reference the inputs are able to execute since as soon as the workbook is opened.

2.7.2 Run Model Event – Standalone

Executing the model is caused by a user action, typically a button click or menu item click. Excel allows you to associate a button (or menu item, hotkey, etc.) with a VBA macro. The screenshot below shows NBTI's model execution function, *runLocal(),* for executing the model in standalone mode:



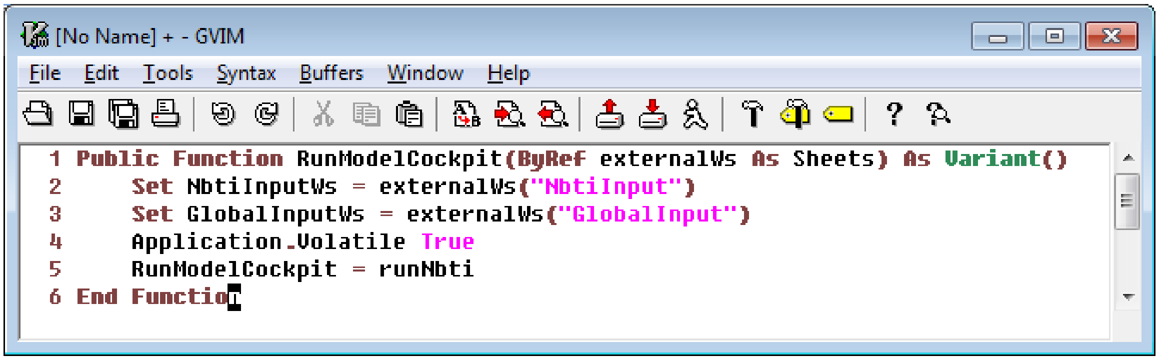
```
 1 Public Sub runLocal()
 2     Application.Volatile True
 3     Dim result() As Variant
 4     result = runNbti
 5 
 6     Dim anchor As Range
 7     Set anchor = GlobalInputWs.Range("NBTI_result")
 8     ' clear existing results
 9     Dim temp() As Variant
10     If anchor.value <> "" Then
11         temp = UtilMod.GetArrayFromContiguousColumn(anchor, 0)
12         anchor.Resize(UBound(temp, 1) - LBound(temp, 1) + 1, UBound(result, 2) - LBound(result, 2) + 1).Clear
13     End If
14 
15     ' write result
16     ' strip headers off result when running local by copying all but the first row of result
17     ReDim temp(LBound(result, 1) + 1 To UBound(result, 1), LBound(result, 2) To UBound(result, 2))
18     Dim i As Integer
19     For i = LBound(temp, 1) To UBound(temp, 1)
20         Dim j As Integer
21         For j = LBound(temp, 2) To UBound(temp, 2)
22             temp(i, j) = result(i, j)
23         Next
24     Next
25     anchor.Resize(UBound(temp, 1) - LBound(temp, 1) + 1, UBound(temp, 2) - LBound(temp, 2) + 1).value = temp
26 End Sub
```

Figure 11: Model Execution Standalone Example Macro

- Line 2 – *Application.Volative = True* tells Excel to execute all cell formulas when this function is called.
- Line 3 – calls the model execution function, *runNbti*, and captures the result in an array. Note that the array is type *Variant*. Though results are nominally all doubles, variant is used to accommodate column headers, NaN, etc. This *result* array is shown in Figure 8: Federate Output.
- Lines 6-13: In this case since the model is running in standalone mode, we want to write the output to the workbook. These lines clear existing output.
- Lines 15-25: Write the output to an internal worksheet.

2.7.3 Run Model Event – Federate

The code below runs the NBTI model as a federate. This function is called externally by the CALM Cockpit workbook, passing in the CALM Cockpit's worksheets to be used as input worksheets. **This function must have the signature shown in line 1 below and reside in a module called RunModel**:



```
1 Public Function RunModelCockpit(ByRef externalWs As Sheets) As Variant()
2     Set NbtiInputWs = externalWs("NbtiInput")
3     Set GlobalInputWs = externalWs("GlobalInput")
4     Application.Volatile True
5     RunModelCockpit = runNbti
6 End Function
```

Figure 12: Model Execution Federate Example Macro

This function is somewhat simpler because we don't have to do anything with the output except return it to the CALM Cockpit.

- Lines 2-3: sets the input worksheet variables to those passed in by the CALM Cockpit.
- Line 4: update all cell formulas
- Line 5: calls the model execution macro, *runNbti*, returning the result to the CALM Cockpit.

2.8 Intermediate Calculations, Charts, etc. in Standalone Mode

Of course most models will produce far more output than the few columns of reliability values returned to the CALM Cockpit. Also, intermediate calculations and results produced by the model's algorithm may have value to the analyst, even though they may not be of interest when executing a federation from the CALM Cockpit. It is up to the model developer to decide how to produce and represent this intermediate output. However, there are a couple of considerations to keep in mind when developing your model:

- Try to be efficient. Excessive computation in cells can be extremely slow compared to computation in VBA. Storing intermediate tables and values in cells might seem expedient, but try to avoid it unless the values benefit the analyst. Consider also adding an internal flag indicating whether your model is running in standalone or federated mode – you may be able to avoid costly computation that is not relevant to the federation when running from the CALM Cockpit.
- When you open a workbook in standalone mode, be mindful that the output being displayed may be stale – it may have last been run from the CALM Cockpit with a different set of inputs than currently resides locally in the workbook. For safety, consider adding a 'clear results' mechanism each time the workbook is opened, particularly for models that may be executed quickly.

2.9 Testing Your Federate

Given identical inputs, a federate should produce identical results when running standalone or in a federation. Verify this by:

1) Copy/paste your federate's local input sheet to the CALM Cockpit
2) Copy the time-stepped global input values from the federate to the CALM Cockpit's GlobalInput worksheet.
3) Execute the federate from the CALM Cockpit.
4) Open the federate workbook and execute it standalone.
5) Compare the 3-column output in the federate to the output in the CALM Cockpit.

See section 3.5 for a description of a tool for comparing data in two workbooks.

3. The CALM Cockpit

The CALM Cockpit is an Excel workbook that assembles and executes a federation of CMOS reliability models. This section provides a brief overview of how to use the CALM Cockpit.

The CALM Cockpit contains the following worksheets:

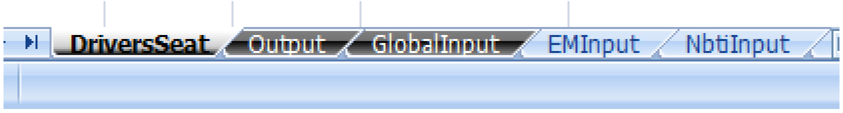


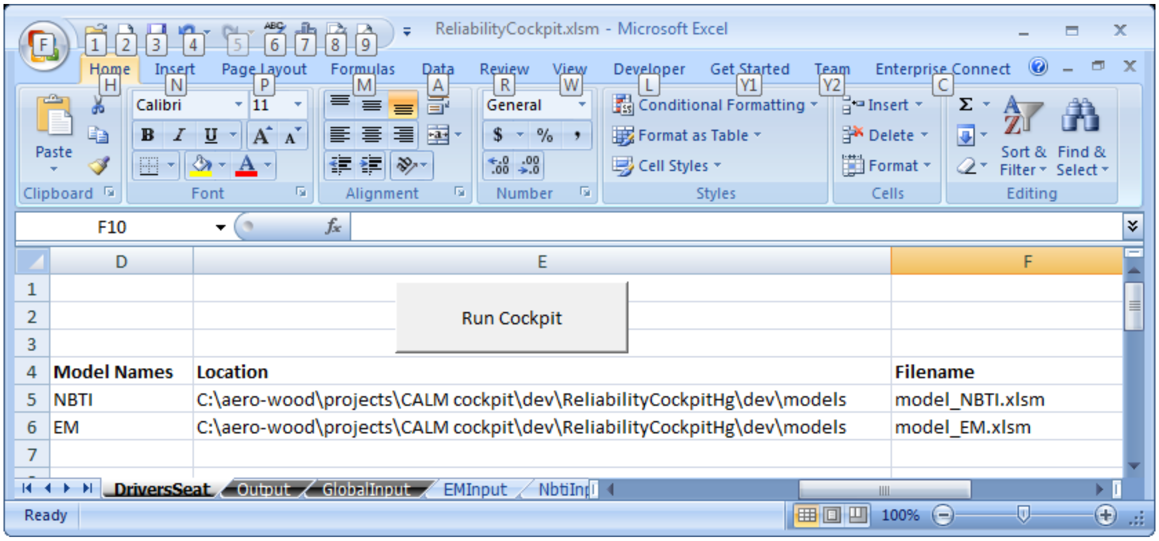
Figure 13: CALM Cockpit Worksheets

The black tabs are standard worksheets, and the blue tabs contain inputs that are particular to models in the federation. The worksheets shown above are as follows:

- DriversSeat: allows the analyst to:
  - define which federates comprise the federation
  - define the order in which they will be executed
  - execute the federation
- Output: collects output from each of the federates
- GlobalInput: contains time-stepped inputs shared by the federates
- EMInput: inputs specific to the Electromigration model
- NbtiInput: inputs specific to the NBTI model
- …and additional tabs for other contributing failure models

3.1 DriversSeat Worksheet -- Creating and Executing a Federation

The DriversSeat worksheet is shown below:



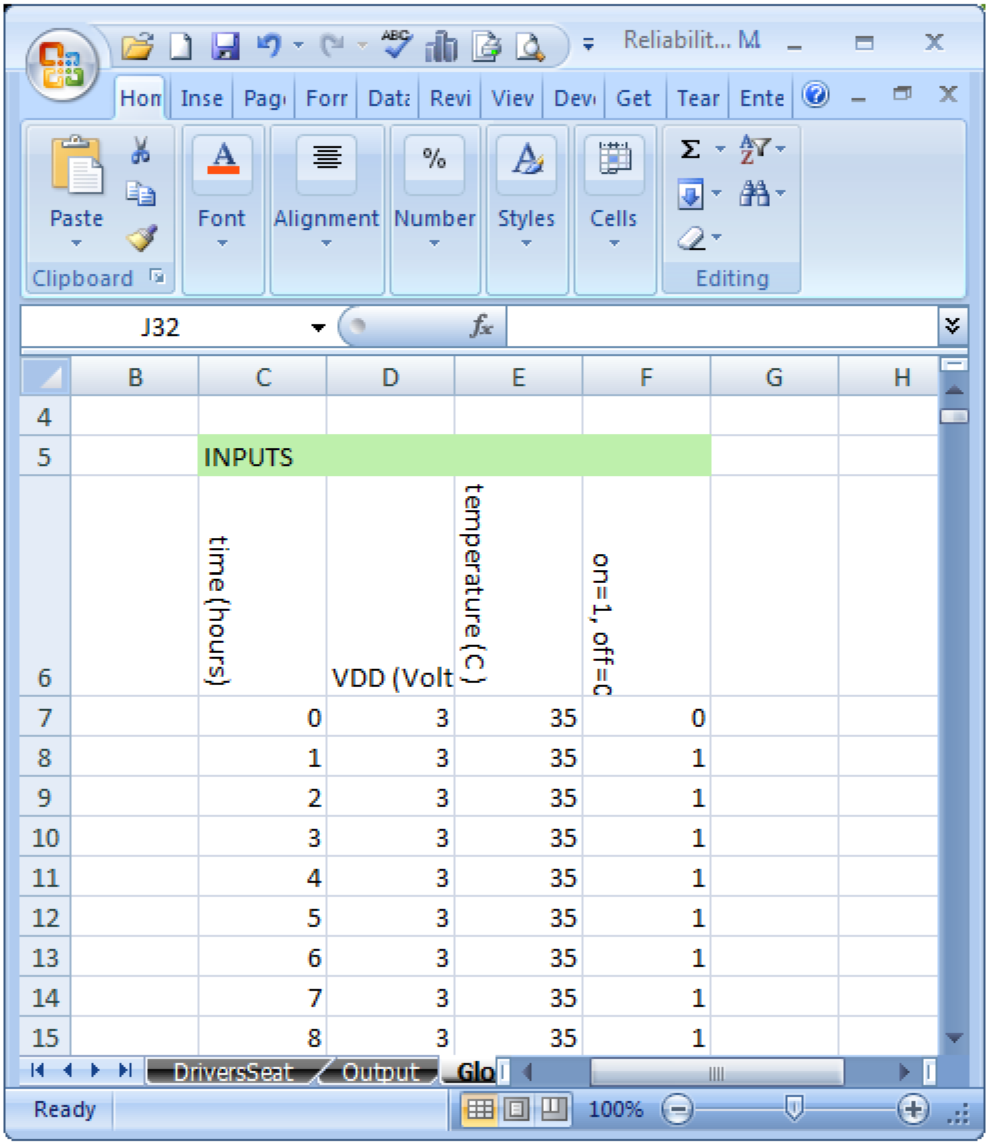
	D	E	F
1			
2			
3			
4	**Model Names**	**Location**	**Filename**
5	NBTI	C:\aero-wood\projects\CALM cockpit\dev\ReliabilityCockpitHg\dev\models	model_NBTI.xlsm
6	EM	C:\aero-wood\projects\CALM cockpit\dev\ReliabilityCockpitHg\dev\models	model_EM.xlsm
7			

Figure 14: DriversSeat Worksheet

The analyst uses this sheet to list the federates in the federation in the order they are to be executed. Click the "Run Cockpit" button to execute the federation.

3.2 GlobalInput Worksheet – Shared Inputs

The GlobalInput worksheet is shown below:



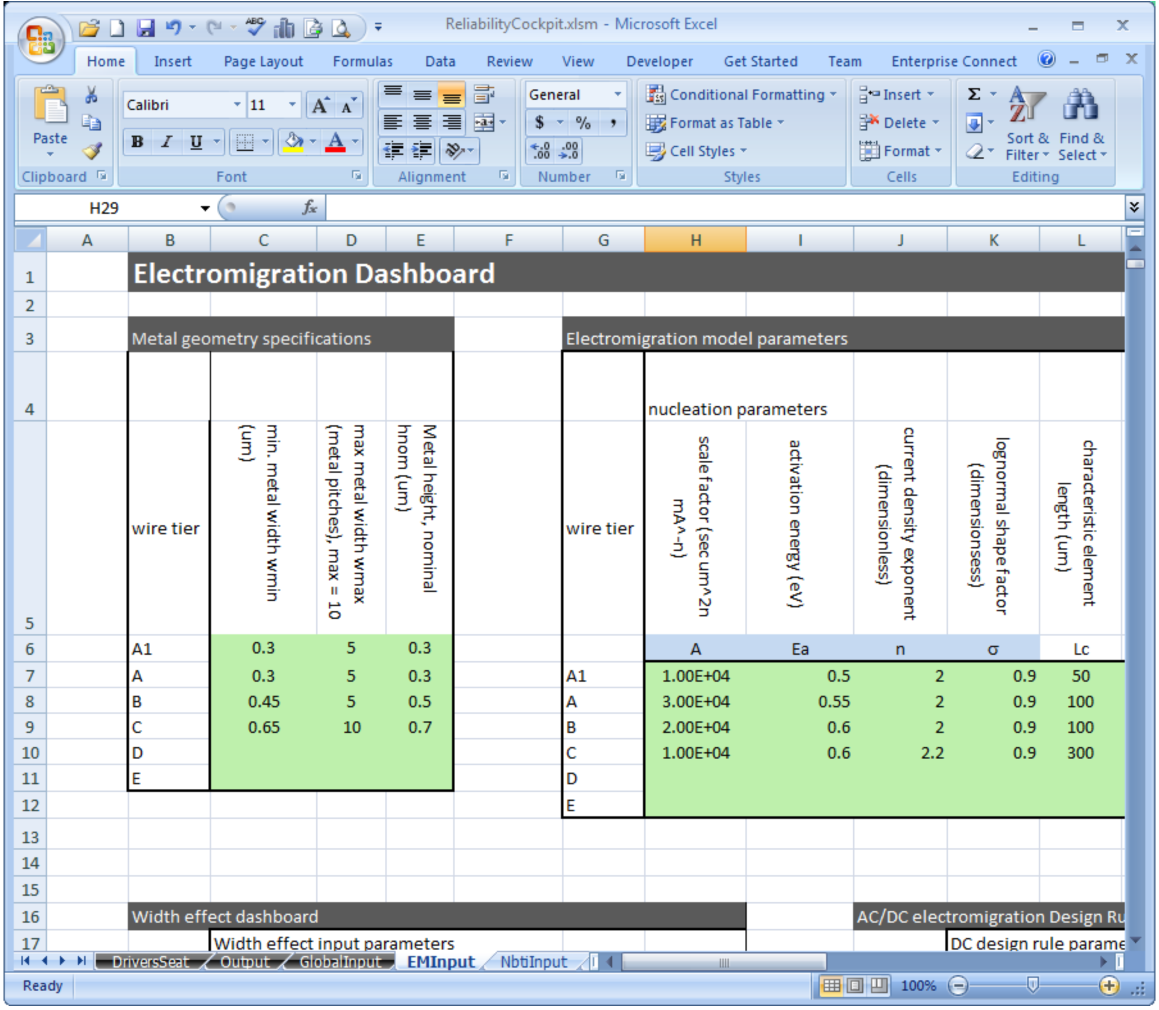
	B	C	D	E	F	G	H
4							
5		INPUTS					
6		time (hours)	VDD (Volt	temperature (C)	on=1, off=C		
7		0	3	35	0		
8		1	3	35	1		
9		2	3	35	1		
10		3	3	35	1		
11		4	3	35	1		
12		5	3	35	1		
13		6	3	35	1		
14		7	3	35	1		
15		8	3	35	1		

Figure 15: GlobalInput Worksheet

This data is shared by all federates. There may be any number of values in this table, but they must be sorted by increasing time (first column) and each row must be complete (no empty cells). The time step increment need not be consistent.

3.3 Local Input Worksheets

This example federation is comprised of two federates, with each having a single worksheet containing its particular inputs. The EMInput sheet below contains inputs specific to the Electromigration model:



Electromigration Dashboard

Metal geometry specifications

wire tier	min. metal width wmin (um)	max metal width wmax (metal pitches), max = 10	Metal height, nominal hnom (um)
A1	0.3	5	0.3
A	0.3	5	0.3
B	0.45	5	0.5
C	0.65	10	0.7
D			
E			

Electromigration model parameters

	nucleation parameters				
wire tier	scale factor (sec um^2n mA^-n)	activation energy (eV)	current density exponent (dimensionless)	lognormal shape factor (dimensionsess)	characteristic element length (um)
	A	Ea	n	σ	Lc
A1	1.00E+04	0.5	2	0.9	50
A	3.00E+04	0.55	2	0.9	100
B	2.00E+04	0.6	2	0.9	100
C	1.00E+04	0.6	2.2	0.9	300
D					
E					

Width effect dashboard

Width effect input parameters

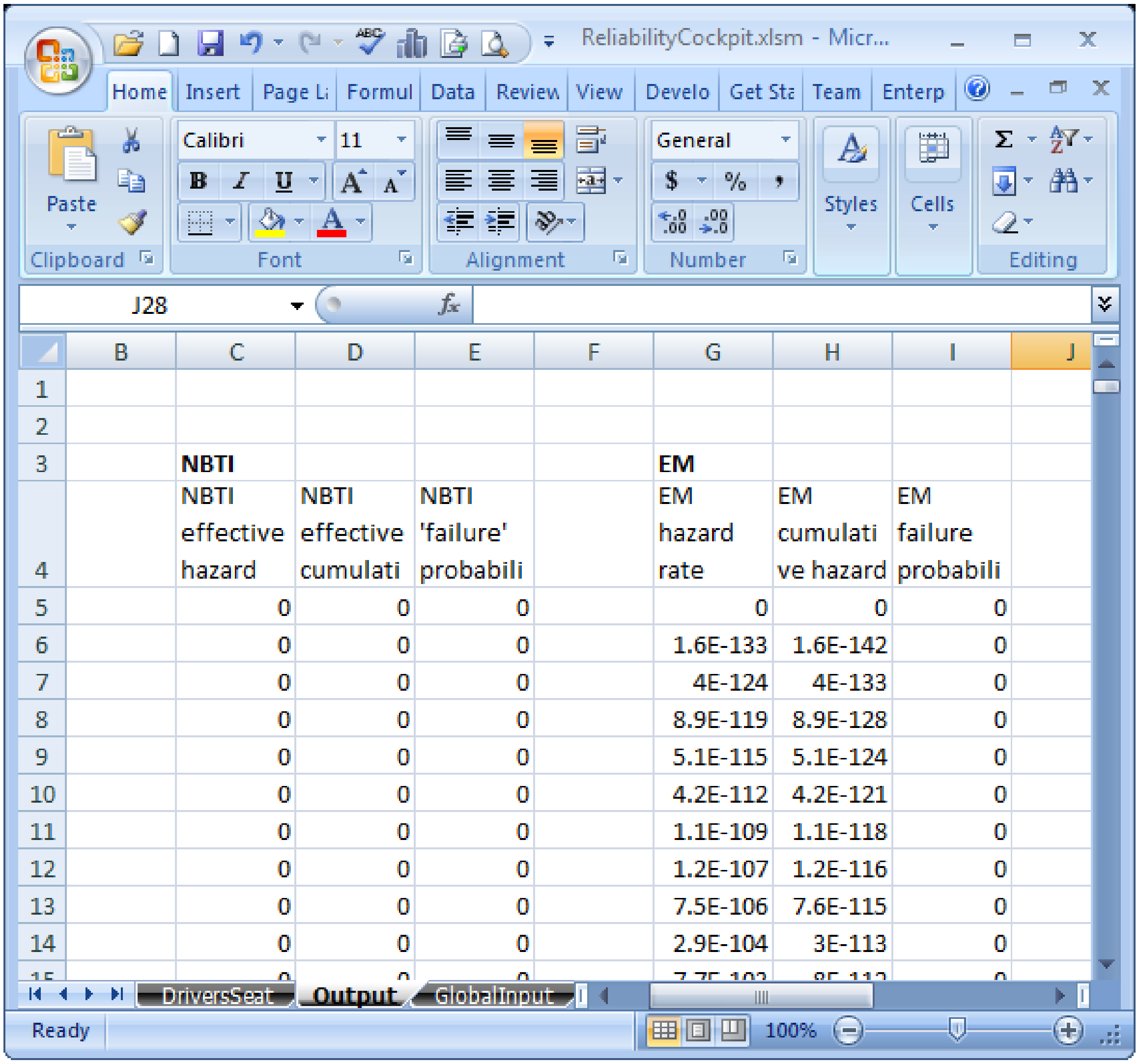
AC/DC electromigration Design Ru

DC design rule parame

Figure 16: Local Input Worksheet (EM Example)

3.4 Output Worksheet

The Output worksheet consolidates output from each of the federates. Since each federate generates a hazard rate as an output, the CALM Cockpit uses these individual Hazards, to produce an aggregate Cumulative Hazard and numerically integrated Cumulative Failure Probability. Shown below are individual federate tabular outputs and aggregated failure probabilities in graphical form.



	B	C	D	E	F	G	H	I	J
1									
2									
3		**NBTI**				**EM**			
4		NBTI effective hazard	NBTI effective cumulati	NBTI 'failure' probabili		EM hazard rate	EM cumulati ve hazard	EM failure probabili	
5		0	0	0		0	0	0	
6		0	0	0		1.6E-133	1.6E-142	0	
7		0	0	0		4E-124	4E-133	0	
8		0	0	0		8.9E-119	8.9E-128	0	
9		0	0	0		5.1E-115	5.1E-124	0	
10		0	0	0		4.2E-112	4.2E-121	0	
11		0	0	0		1.1E-109	1.1E-118	0	
12		0	0	0		1.2E-107	1.2E-116	0	
13		0	0	0		7.5E-106	7.6E-115	0	
14		0	0	0		2.9E-104	3E-113	0	
15		0	0	0		7.7E-103	8E-112	0	

Figure 17: Output Worksheet

The CALM Cockpit produces standard graphs of the columnar output for each federate. It also produces a graph showing the overall reliability of the federation, as shown below in Figure 18.

The analyst also has the complete Excel toolset at their disposal for analyzing this output.

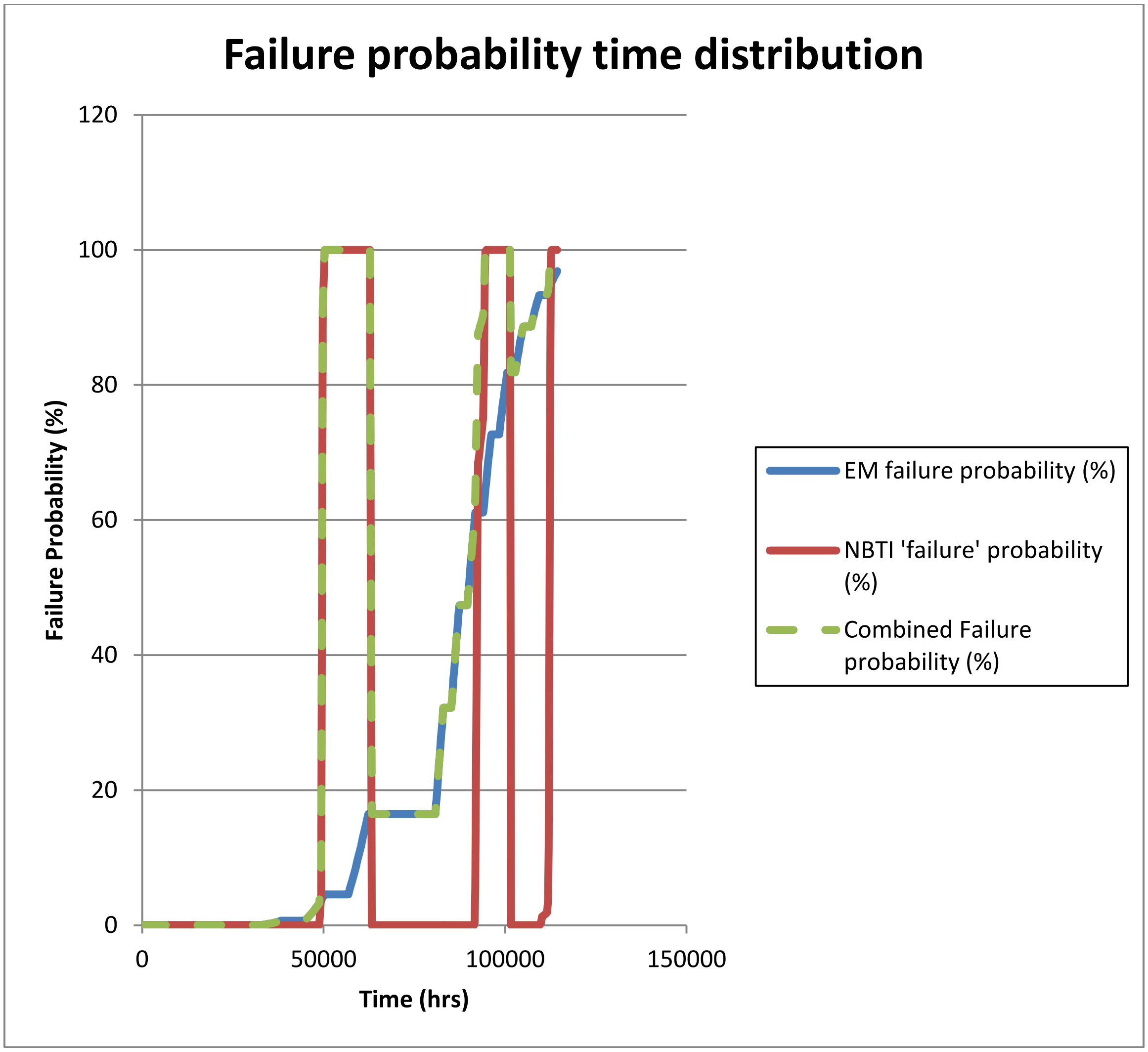


Figure 18: Combined Federation Reliability Plot

3.5 Comparison Tool

The CALM Cockpit also provides a comparison tool to aid testing in its ‘CompareTool’ worksheet. The tool allows you to compare two ranges of cells in Excel workbooks. The compare tool may be used for comparing federate results with CALM Cockpit results.

4. Trusted Locations

The CALM Cockpit and Federate workbooks rely on macros to operate correctly. On some systems, Excel macros are disabled by default.

All CALM workbooks should be placed in a 'trusted location' for seamless enabling of macros. Trusted locations are directories on your file system where Microsoft Office macros may be run without being checked by Excel's Trust Center security feature.

Create a trusted location by following these steps (copied from the website below, and subject to change with future releases of Microsoft Excel):

1) Click the Microsoft Office Button Button image, and then click Excel Options.
2) Click Trust Center, click Trust Center Settings, and then click Trusted Locations.
3) If you want to create a trusted location that is not local to your computer, select the Allow trusted locations on my network (not recommended) check box.
4) Click Add new location.
5) IMPORTANT   We recommended that you don't make your entire Documents or My Documents folder a trusted location. Doing so creates a larger target for a hacker to potentially exploit and increases your security risk. Create a subfolder within Documents or My Documents, and make only that folder a trusted location.
6) In the Path box, type the name of the folder that you want to use as a trusted location, or click Browse to locate the folder.
7) If you want to include subfolders as trusted locations, select the Subfolders of this location are also trusted check box.
8) In the Description box, type what you want to describe the purpose of the trusted location.
9) Click OK.

Please see the Microsoft documentation at the following link for further details:

http://office.microsoft.com/en-us/word-help/create-remove-or-change-a-trusted-location-for-your-files-HA010031999.aspx

5. Summary

This API was created to enable Reliability Systems Engineers to use low cost, readily available Microsoft Excel-based spreadsheet tools to develop physics-of-failure lifetime reliability models and to aid in sharing those individual failure mechanism models in the broader high reliability electronics community. The API provides a framework and cockpit to run these individual models stand-alone, to understand an individual failure mechanisms reliability impact on a full chip, and to enable aggregation of individual failure rates to a combined full-chip failure rate when modeled with all contributing failure models.

All trademarks, service marks, and trade names are the property of their respective owners.