BERT – CIRCUIT ELECTROMIGRATION SIMULATOR

by

B. K. Liew, P. Fang, N. W. Cheung, and C. Hu

Memorandum No. UCB/ERL M90/3

8 January 1990

BERT – CIRCUIT ELECTROMIGRATION SIMULATOR

by

B. K. Liew, P. Fang, N. W. Cheung, and C. Hu

Memorandum No. UCB/ERL M90/3

8 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

BERT - CIRCUIT ELECTROMIGRATION SIMULATOR

B. K. Liew, P. Fang, N. W. Cheung, and C. Hu
Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, California 94720

Abstract

Models for predicting interconnect and intermetallic contact reliability under arbitrary current waveforms have been developed [1-3]. These models are incorporated in the Circuit Electromigration Simulator module in BErkeley Reliability Tool (BERT). The simulator can (1) generate layout advisory for width and length of each interconnect, the safety factor of each contact and via in a circuit to meet user-specified reliability requirements and (2) estimate the overall circuit electromigration failure rate and/or cumulative failure percent of a layout design.

This work was partially supported by ISTO/SDIO administered through ONR under Contract N00019-85-K-0603.

Table of Contents

I. Overview of the Simulator

1.1 Introduction

In the effort to assure reliable design in VLSI system, it is imperative that the reliability guidelines and potential reliability hazards are known during the design stage of a circuit. In this respect, circuit reliability simulator becomes an indispensable tool for circuit designers. The BErkerley Reliability Tool (BERT) is developed with the aim to assist circuit designer in implementing design for reliability. Presently, BERT has three modules: Circuit Aging Simulator (CAS) [4] for hot-electron degradation, Circuit Oxide Reliability Simulator (CORS) [5] for MOS oxide breakdown and Circuit Electromigration Simulator for interconnect and contact electromigration failures. This report describes the Circuit Electromigration Simulator.

1.2 Overview of the Simulator

To use the EM simulator, the circuit designer prepares the SPICE input deck the usual way or uses the layout extractor to produce the SPICE input deck. But before invoking SPICE to perform circuit analysis, the EM simulator pre-processor is called. The pre-processor adds a dummy voltage source in the original SPICE input deck for every terminal of the circuit element and requests SPICE to printout the current in all the dummy voltage sources (this is necessary because SPICE can only print current flowing in voltage sources). In addition, a database of the dummy voltage sources and their associated circuit elements are generated by the pre-processor. This information will be passed automatically to the post-processor. The modified input deck can now be sent to SPICE simulator.

Output from SPICE simulation is filtered by the post-processor which reads the current waveforms and performs the required calculations. At the same time, the post-processor will remove from the SPICE output the voltage sources and printout lines added by the pre-processor. In this manner, the operation of the simulator is transparent to the user.

There are two modes of operation for the simulator, either one or both can be selected. In the first mode (Fig. 1(a)), the simulator is used as as a layout advisor. The user inputs the desired reliability specification in failure rate after a specified device operating hours. The simulator will generate layout guidelines for width and length of each interconnect (up to three layers of interconnect are supported), and safety factor for each contact or via in a circuit to meet user-specified reliability requirement.

In the second mode of operation (Fig. 1(b)), the simulator calculates the failure rate and/or the cumulative percent failure of a circuit layout. The user can supply the layout geometry of all the interconnects, contacts and vias for the layout by hand or more conveniently use the layout extractor which is available with our simulator. The layout extractor reads in the Caltech Intermediate Format (CIF) layout description file, extracts the circuit elements, and generates the SPICE input deck for the circuit. At the same time, it also fractures the interconnect patterns into segments and produces a geometry description file containing the length and width of each segment. Area of metal-to-metal vias and metal-to-silicon contacts are also identified. The geometry information is added to the database produced by the pre-processor and will be used by the post-processor to calculate overall circuit electromigration reliability.

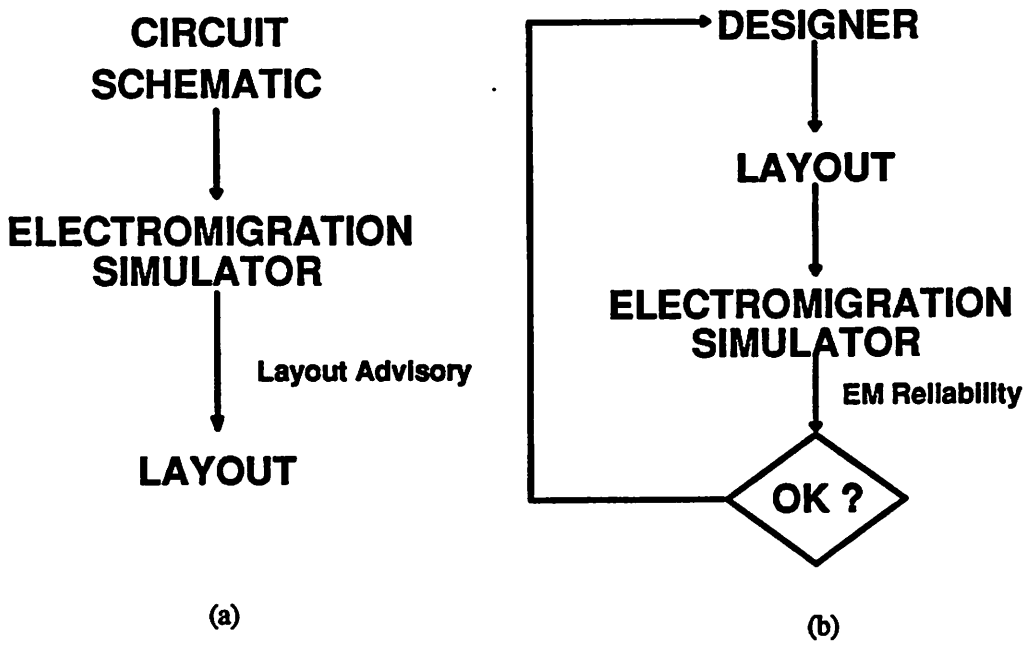

Fig.1. The electromigration simulator can be used in two ways: (a) to generate layout advisory, (b) to compute circuit failure rate and cumulative percent failure.

II. Electromigration Reliability Model

2.1 Introduction

This section briefly describes the physical and statistical models implemented in the simulator.

2.2 Dependence on current density

The electromigration time-to-failure (TTF) under arbitrary current waveforms (which is valid for frequencies of waveforms greater than 1kHz) is given by [2]:

$$\mathrm{TTF} = \frac{A_{DC}(T)}{\overline{|J|^{m-1}}\,\bar{J}\left[1+\frac{A_{DC}}{A_{AC}}\frac{(\overline{|J|}-\bar{J})}{\bar{J}}\right]} \tag{1}$$

where $\bar{J}$ is the average current density, $\overline{|J|}$ is the average of the absolute current density. m, A_{DC} and A_{AC} are experimentally determined constants which need to be supplied to the simulator by the user. $A_{AC}(T)$ and $A_{DC}(T)$ have Arrhenius dependence on temperature with the same activation energy E_a.

2.3 Dependence on geometry

The simulator uses the independent element analysis for length dependence described in Ref[6]. The user is requested to input one time-to-failure result for long interconnect (longer than the longest interconnect in the circuit). Failure statistics are calculated for shorter lines by the following assumption: long interconnect is modeled by a series of shorter segments (see Appendix A) and its TTF is determined by the weakest segment. Therefore, the failure rate of a long metal line is the sum of the failure rates of shorter segments. For example, if the user inputs a MTF of 7.5 hours for 4.5 cm long interconnect. From this data, the simulator can construct the failure distribution for a line half the length of the original test line (with failure probability of $F(t)$). Because the 4.5 cm long line will fail if either or both of the two segments fail, the failure probability of a 2.25 cm line $G_2(t)$ is:

$$[1-G_2(t)]^2 = [1-F(t)]$$

$$G_2(t) = 1-[1-F(t)]^{\frac{1}{2}}$$

The first equation states that the probability that the long line will not fail is the product of the probability of the two shorter segments are good. Thus, at time $t = 7.5$, the failure probability for the 2.25 cm long line is 0.29. In general, for a line that is $1/x$ of the long test line, $G_x(t)$ is:

$$G_x(t) = 1-[1-F(t)]^{\frac{1}{x}} \tag{2}$$

The width dependence of TTF is obtained from empirical fit of TTF versus linewidth data using piecewise fit of two second order polynomial functions (see Appendix B), which would allow an increase in TTF as linewidth is decreased below the average grain size [7]. The width dependence is specified by four parameters, A_w, B_w, C_w and D_w which are defined by the following equations:

for $W \geq B_w$:

$$\mathrm{TTF}(W) = A_w \times (W-B_w)^2 + D_w \tag{3a}$$

for $W < B_w$:

$$\mathrm{TTF}(W) = C_w \times (W-B_w)^2 + D_w \tag{3b}$$

B_w is the linewidth at the minimum of TTF. It is approximately equal to the average grain size in the interconnect.

The failure probability for one contact/via opening is similarly calculated from the test data of a chain of contacts/vias.

2.4 Dependence on statistical distribution

The user is given the choice of using either lognormal or Weibull distribution function to calculate failure rate and cumulative percent failure. The user is advised to choose the appropriate distribution function that best describes experimental time-to-failure data. Ref[8] has an excellent discussion on the statistical failure distributions.

Although lognormal distribution function is commonly used to represent the experimental time-to-failure data, there is no physical foundation to expect that the electromigration failures should be lognormally distributed. If lognormal distribution is chosen, the length dependence model adopted in the simulator will results in distributions for shorter lines that are not lognormal. This inconsistency is illustrated in Appendix A. If Weibull distribution is chosen, the inconsistency is removed, i.e. failure distributions for all lines are Weibull.

III. Operation of the Simulator

3.1 Installing the simulator

The **makefile** in the BERT distribution tape will automatically compile all modules and create the BERT executable codes **prebert** and **postbert**. Simply type **make** in UNIX. In addition, **make** will create the executable codes for the layout extractor **mextra**, **sim2spice** and also the code for a cif plotting program **cif2ps**. A "standalone" version of the electromigration simulator can be created (stripped off hot electron circuit aging and time-dependent oxide breakdown modules) by typing **make emonly** in the **BERT/EM** directory.

3.2 Notational conventions

The following notations will be used to indicate how the command should be entered:

- Keywords shown in **bold** must be entered exactly as shown. However, they can consist of any combination of uppercase and lowercase letters.

- The items in lowercase *italic* must be supplied. For example, the name of the file should be entered when *filename* is shown in the format.

- Items enclosed in square brackets ([]) are optional. If optional information is to be included, the information is entered without the brackets.

- Items enclosed in braces ({}) are to appear together. Information is not complete if one of the items is not specified.

- Either one of the two items (or groups of items in braces) separated by a bar (|) can be specified but not both together.

- An ellipsis (...) indicates that an item can be repeated as many times as needed but the item(s) must stay in one line.

3.3 Simulator commands in SPICE input deck

The following commands are inserted in the SPICE input deck to request EM analysis. Only one command can appear in each line.

(1) .EMMODEL *filename*

Example:

.EMMODEL emrulefile

This command tells the simulator that reliability parameters are in emrulefile.

(2) .EMSTAT *filename hour* [*hour*...]

Example:

.EMSTAT bicmos.int 100.0 1.0e3 2e4

This command specifies the failure rates are to be calculated at times: 100 hours, 1000 hours and 20000 hours. The geometry file which contains the length and width of interconnects, number of contact and via openings that the user set up (by hand) is *bicmos.int*.

(3) .EMSTATX *filename hour* [*hour*...]

Example:

.EMSTATX fulladder.geo 200.0

This command specifies the failure rates are to be calculated at times: 200 hours. The geometry file extracted from the layout by the extractor is *fulladder.geo*.

Either one of **.EMSTAT** or **.EMSTATX**, but not both is needed if user requests failure rate calculation for a circuit layout.

3.4 Setting up reliability parameters

The simulator needs as input the necessary reliability parameters to perform electromigration analysis. The parameters are given in a file (which will be referred to as the EM design rule file). There are two types of parameters: numerical and logical. The parameter name must be typed in full. However, it can contain a combination of lowercase and uppercase letters. There is no requirement on the order the parameters are entered in the file. User is allowed to enter more than one parameters on a single line. Default parameters are given in Appendix C.

- Logical parameters are entered without assignment. It is set true as it appears.

Example:

PRINTCURRENT

- Numerical parameters are assigned values by following the name of the parameter immediately by an equal sign, and the value. The value must also immediately follow the equal sign. There can be no blank space in between.

Example:

LENGTH=1.2e4

- Comments can be entered by entering * in the first column of the comment line. Entries in that line will not be read by the simulator.

Example:

*This is a comment line

3.5 Logical parameters in the rule file

The list of logical parameters are:

(1) **PRINTCURRENT**
Request simulator to print out the current in each connection.

(2) **SKIPLAYOUTCUR**
Normally, the simulator generates the layout advisory table containing the width and length of interconnect, the safety factor of contacts and vias in a design for a number of specified current values to meet the reliability specification. This option requests the simulator not to generate this table.

(3) **SKIPLAYOUTGEO**
Normally, the simulator generates the layout advisory table containing width and length of each interconnect in the circuit, the safety factor of contacts and vias in every connection of the circuit to meet the reliability specification. This option requests the simulator not to generate this table.

(4) **SKIPFAILRATE**

(5) **SKIPFAILPERCENT**
The simulator will not print out the cumulative percent failure table (if **SKIPFAILPERCENT** is specified) or the failure rate table (if **SKIPFAILRATE** is specified). The default is both tables will be printed if either **.EMSTAT** or **.EMSTATX** card is found in the SPICE input deck.

(4) **METAL1**
The reliability parameters that follow are for metal-one.

(5) **METAL2**
The reliability parameters that follow are for metal-two.

(6) **METAL3**
The reliability parameters that follow are for metal-three.

(7) **CONTACT**
The reliability parameters that follow are for metal-one contact to diffusion.

(8) **VIA**
The reliability parameters that follow are for metal-to-metal vias.

3.6 Numerical parameters in the rule file

The list of numerical parameters are:

(1) **NCURRENT=***ncurrent current1 current2 ... currentn*
Example:
NCURRENT=4 1.0e-4 2.0e-4 1.0e-3 2.0e-3
ncurrent is the number of current values the simulator uses to generate the current layout advisory table. *current1..currentn* are given in Amperes. There must be exactly *ncurrent* fields following *ncurrent* and they must all be in one line.

(2) **AC_DEFINE=***ac_define*
Example:
AC_DEFINE=0.1
ac_define is used as a criterion for pure AC waveforms. The simulator will treat a current waveform as pure AC if the average current is less than or equal to *ac_define* × the average of the absolute current. This parameter is important because quite often the transient analysis in SPICE does not result in pure AC waveform for charging and discharging of a capacitor node. This is because the time duration requested in the .TRAN card might not cover one period under steady state operation. Usually a value of 0.1-0.2 is reasonable. To obtain steady state waveforms, user is advised to set up SPICE to run for more than one cycle of the waveform.

(3) **MINJCURRENT=***minJcurrent*
Example:
MINJCURRENT=1.0e3

In order to save computation time, when the current density in the interconnect, contact or via is below **minJcurrent** (in A/cm^2), the simulator will skip failure calculation and prints **J <MinJ** in the output.

(4) **WORSTLIST=***WorstList*
Example: WORSTLIST=0.5

The simulator will print out worst *WorstList* fraction of the connections in the failure rate/cumulative percent failure tables. In the example, half (50%) of the worst connections will be printed.

(5) **NWIDTH=*nwidth width1 width2 ... widthn***
Example:
NWIDTH=3 0.8 1.2 2.0

nwidth is the number of widths the simulator will use to generate the layout advisory table for interconnect. *width1..widthn* are given in μm. There must be exactly *nwidth* fields following *nwidth* and they must all be in one line.

(6) **NCV=*ncv N1 N2 ... Nn***
Example:
NCV=4 1.0 2.0 3.0 4.0

ncv tells the the simulator that user wishes a layout advisory table for contacts and vias containing *ncv* columns for *N1..Nn* openings. In each column a safety factor is generated for the number of contact or via openings (in parallel) at that connection. There must be exactly *ncv* fields following *ncv* and they must all be in one line. The example shown above will request simulator to calculate safety factor for connections having: one contact/via, two contact/via, three contact/via and four contact/via openings.

(7) **SPEC_TIME=*spec_time***
Example:
SPEC_TIME=1.0e6
spec_time is the device operating hours when failure rate is *spec_failrate* in the user's reliability specification. *spec_time* is in hours. The simulator will generate layout advisory to meet this spec.

(8) **SPEC_FAILRATE=*spec_failrate***
Example:
SPEC_FAILRATE=1.0e-3
spec_failrate (in 1/hour) is the failure rate at *spec_time* (hours) in the user's reliability specification. Note that 1 FIT = 0.1% percent (10^{-3}) failures per million (10^{6}) device hours is equal to a **SPEC_FAILRATE** of 10^{-9}. The simulator will generate layout advisory to meet this spec.

(9) **TOP=T_{op}**
Example:
TOP=25
The circuit operates at temperature T_{op} (in °C). The accelerated testing data entered by the user will be extrapolated to this temperature.

The following parameters apply to metal-one, metal-two, metal-three, metal-one to diffusion contact and metal-to-metal via. To associate the parameters to one of the connection types, one of the logical parameters: **METAL1, METAL2, METAL3, CONTACT** and **VIA** must be set before any of the following appears. Items (1) to (14) must be given to complete the parameter set for each connection type. The simulator will skip any connection type with incomplete parameter set.

(1) **ADC=A_{DC}**

(2) **AAC=A_{AC}**

(3) **M=*m***

(4) **TDATA=T_{data}**

(5) **EA=E_a**
Above are the parameters in Eq.(1). Note that, the user will have to enter the experimental determined values of A_{DC} and A_{AC} (both are in units of hours × $(A/cm^2)^2$). These values are obtained from experiment at temperature T_{data} (in °C). E_a is the activation energy for A_{DC}

and A_{AC}.

(6) THICK=*thickness*

(7) WIDTH=*width*

(8) LENGTH=*length*
These specify the *thickness*, *width* and *length* (all in μm) of interconnect used in electromigration lifetime experiment to extract A_{DC}, A_{AC} and m.

(9) AREA=*area*

(10) NCHAIN=*nchain*
This specifies the *area* (in μm^2) of each contact or via in the chain test structure used in lifetime experiment to extract A_{DC}, A_{AC} and m. *nchain* is the number of contacts or vias in series in the test structure.

(11) LOGMEDIAN=*MTF*

(12) LOGSIGMA=σ
Select lognormal distribution with median *MTF* (in hours) and the lognormal standard deviation σ. The MTF is the experimental median-time-to-failure at temperature T_{data}.

(13) WEIBULL_A=α

(14) WEIBULL_B=β
Select Weibull distribution and specify the parameters α and β. The failure data are collected at temperature T_{data}. The Weibull cumulative distribution function is described by:

$$F(t) = 1 - \exp(-\frac{1}{\alpha} t^{\beta}) \tag{4}$$

where t is in hours.

(15) WIDTH_A=A_w

(16) WIDTH_B=B_w

(17) WIDTH_C=C_w
Specify the width dependence of time-to-failure for interconnect (see Appendix B for an example). The parameters A_w, B_w, C_w are defined by Eqs.3(a) and 3(b) which are repeated below:
for $W \geq B_w$:

$$\mathrm{TTF}(W) = A_w \times (W - B_w)^2 + D_w \tag{3a}$$

for $W < B_w$:

$$\mathrm{TTF}(W) = C_w \times (W - B_w)^2 + D_w \tag{3b}$$

D_w is found using the previously entered experimental time-to-failure and *width*. W is in μm. Note that by setting, $A_w = C_w = 0$, TTF will be a constant independent of width.

3.7 Parameters needed by the simulator

One set of physical and statistical parameters are needed for each layer of interconnect, contact and via. The parameters are entered by the user in the EM design rule file which will be read by the post-processor.

The user is required to enter the parameters in Eq.(1) for long interconnect: A_{DC}, A_{AC}, m and E_a. These parameters can be extracted from the TTF versus current density plot and the Arrhenius plot of TTF versus temperature. In addition, the length, width and thickness of the test

structures are input to the simulator. The user also selects either lognormal or Weibull distribution to represent the TTF data. If TTF is required to be a function of width (the simulator defaults to constant width dependence), the width parameters (Eqs.(3a) and (3b)) have to be extracted from additional experiments using the same long line but with a number of widths.

The parameter sets for contact and via are set up similarly. The test data for contact or via electromigration are obtained from contact/via chain test structures. User enters the A_{DC}, A_{AC}, m, E_a as well as the area and the total number of contacts/vias in the chain.

3.8 Example of EM design rule file:

The following is an example of the rule file:

```
* Reliability parameters for electromigration simulator
PrintCurrent
* print the current in each connection in the circuit
ncurrent=3 0.50e-3 1.00e-3 2.00e-3
* requests simulator to generate current layout guidelines for 3 current values
SkipFailpercent
* Do not print cumulative percent failure table, only failure rate table will
* be printed
WorstList=0.5
* print half (50%) of the worst connections in failure rate/cumulative percent
* failure table.
AC_define=0.2
* Definition of Pure AC current
MinJcurrent=1e2
* ignore anything with current density less than 1e2 A/cm2
spec_time=1.0e+4 spec_failrate=1.0e-9
* this is the reliability specifications
nwidth=2 1.0 2.0
* Generate layout guidelines for two widths of interconnect
ncv=3 1.0 2.0 4.0
* Generate layout guidelines for three contacts/vias openings: 1, 2 and 4
* at a connection
* The following set for metal-one
metal1 length=4.5e+4 width=1.0 thick=0.1
Tdata=25.0 Ea=0.5
width_a=0.0 width_b=1.0
m=2.0 Adc=1.0e+16 Aac=1.0e+20
Lognormal logsigma=1.0 logmedian=7.52
* The following set for metal-two
metal2 length=4.5e+4 width=1.0 thick=0.1
Tdata=25.0 Ea=0.5
width_a=0.0 width_b=1.0
m=2.0 Adc=1.0e+15 Aac=1.0e+19
Lognormal logsigma=1.0 logmedian=7.52
* The following set for via
via area=1.0 nchain=10.0
* The test structure is a chain of 10 vias, each is 1μm^2
Tdata=25.0 Ea=0.5
area_a=0.0 area_b=1.0
m=2.0 Adc=1.0e+16 Aac=1.0e+20
Lognormal logsigma=1.0 logmedian=7.52
```

* Because set for contact and metal-three are not given
* Simulator will ignore them

3.9 Invoking the simulator

The simulator can be invoked using the following command line:

prebert [-*x*] *deckfile* | *spice* | **postbert** [> *outfile*]

x is the version of *spice*, **2** for SPICE2, **3** for SPICE3B1 and **4** for SPICE3C1. The default (if no option is specified) is SPICE3C1. *spice* is the SPICE command used.
Example:

vivante>prebert bicmos | spice3c1 | postbert > em.out

The command to invoke the "standalone" version of the pre-processor is:

preem [-G *geometryfile*] *deckfile* | *spice* > *spiceoutfile*

The pre-processor will generate the geometry file in *deckfile*.geo (or optionally named *geometryfile* with -G). If the SPICE input deck is not generated from the layout extractor, the pre-processor will add dummy voltages in the SPICE input deck. The dummy voltages have prefix VEM and the added node number will start from 5000. To prevent error in the simulator, the original SPICE input deck must not have voltage sources with prefix VEM and node number equal to or exceeding 5000. User can change the starting number of dummy nodes by defining new STARTNODE in the pre-processor header file **empredef.h**
The command to invoke the "standalone" version of the post-processor is:

postem [-S *x*] -R*rulefile* -G*geometryfile spiceoutfile* [> *outfile*]

x is the version of *spice*, **2** for SPICE2, **3** for SPICE3B1 and **4** for SPICE3C1. The default (if no option is specified) is SPICE3C1. *rulefile* is the EM design rule file described previously and *geometryfile* is the geometry file from the pre-processor. *spiceoutfile* is the output file from SPICE simulation.

By using the "standalone" version of the pre- and post-processor, the user can save the SPICE output *spiceoutfile* (which can be quite a large file). This file can be post-processed again to generate layout guidelines and failure statistics if any of the parameters in the design rule file is changed. Thus user does not need to run SPICE analysis again.

IV. Operation of the Layout Extractor

4.1 Introduction

This section describes the procedures to extract SPICE input deck from CIF layout file. First, **mextra** is used to extract the connectivity of transistor, interconnect, contact and via. The extracted information is written to a file in **sim** format. **sim2spice** is called next to construct the SPICE input deck and also produce the layout geometry database from the **sim** file. A CIF to PostScript plotting program **cif2ps** is provided to plot the CIF file. **cif2ps** can also be used to locate input and power supply nodes in the SPICE deck and also to view any reliability hazards in interconnects, vias and contacts.

4.2 Manhattan circuit extractor for VLSI simulation

The operation of the layout extractor: **mextra** is described in the accompanying manual page (see Appendix G). The technologies known to **mextra** are: nMOS ("nmos"), MOSIS P well CMOS/Bulk, also known as CBPM ("cmos-pw"), MOSIS Scalable CMOS/Bulk N-well, also known as SCN ("cmos-nw"), MOSIS Scalable CMOS/Bulk P-well, also known as SCP ("cmos-s"), and MOSIS Scalable CMOS/Bulk Generic, also known as SCG ("cmos-g"). The mask layer names for each technology are listed in Appendix D. If the CIF layers have different names than the ones listed in Appendix B, user can set up his/her own CIF layer names and use the **-L** option of **mextra** (explained in Appendix E). An example of the command line is:

vivante>mextra -t scmos circ.cif
which extracts the layout from *circ.cif* file. The technology is *scmos*.

4.3 Modifications in mextra

Detailed description of the original version of **mextra** can be found in Ref[9]. Modifications have been made to the original version to extract interconnect width and length, metal-to-metal via and metal-to-silicon contact area. The additional layout information is appended to the *basename*.**sim** file. The line describing the via or contact geometry has the following format:

type node1 node2 area xloc yloc;

where *type* is either **CONT** or **VIA** (for metal-to-silicon contact or metal-to-metal via), *node1* and *node2* are the connection nodes, *area* is in square centi-microns (=0.0001 μm^2). The location of the via or contact is given by *xloc* and *yloc* in cif coordinates.
The line describing the metal-one or metal-two interconnect has the following format:

type node1 node2 width length xloc yloc;

where *type* is either **M1** or **M2**, *node1* and *node2* are the connection nodes, *width* and *length* are in centi-micron (=0.01 μ). The location of the interconnect is given by *xloc* and *yloc* in cif coordinates.

4.4 Limitations of the layout extractor

The layout extractor has been tested for a number of circuit designs. Although it works well in most cases, the user has to be aware of its limitations:

(1) **mextra** can only handle Manhattan type structure. Non-manhattan polygon will be ignored by the extractor.

(2) **mextra** can only extract two layers of interconnects.

(3) The extracted interconnect length and width are accurate for long metal lines. **mextra** tends to err at irregular corners and where many metal lines join together.

(4) **mextra**, by default only recognizes the technologies listed in Appendix D. The user is advised to check the layer names in his/her CIF file with the layers listed in the table to

prevent unknow layers error. If necessary, user can set up alternate CIF layer names (see Appendix E).

4.5 Converting .sim format to SPICE input format

The program **sim2spice** converts the *basename*.**sim** file produced by **mextra** to SPICE input deck and produces the layout geometry database in file *basename*.**geo**. Two files are needed for successful conversion: *basename*.**sim** which contains connectivity information and *basename*.**nodes** which contains node names and numbers. Both files are generated by **mextra**. **sim2spice** produces the spice input deck in *basename*.**spice** and the geometry information of metal interconnects, contacts and vias in *basename*.**geo**. In the SPICE deck, **sim2spice** will add for each interconnect and contact structure a dummy voltage source. For metal-to-silicon contact and metal-to-metal via a small series resistance (0.001 Ω) will also be added in the SPICE deck. The name of the dummy voltage source has prefix **VEM** and the series resistor has prefix **REM**. The resistor is needed to prevent SPICE error when there is a closed loop consisting of voltage sources only.

The SPICE deck from **sim2spice** is incomplete. It does not have the input and power sources and it does not request any spice analysis. User is required to insert the voltages, set up the transient analysis card. The user will also have to insert the following two cards to request EM analysis (explained in Sec 3.3):

.EMMODEL *rulefile*
.EMSTATX *basename*.geo *1.0e4 1.0e5*

where *rulefile* is the EM design rule file and *basename*.**geo** is the extracted geometry file from **sim2spice**.
The user can locate the input, power and output nodes of the circuit by using the **cif2ps** program. This is described in Section 4.5.

The extracted SPICE deck may have a number of floating nodes where there is only one connection to the node. This will cause SPICE2G6 to abort simulation. Therefore, the user is advised to use SPICE3C1 or SPICE3B1 for the extracted SPICE input deck.

The geometry information of the layout in *basename*.geo has the following format:

node1 node2 VEMxxx type {*width length*} | *area xloc yloc;*

where *node1* and *node2* are the nodes in SPICE deck, *VEMxxx* is the dummy voltage source in SPICE deck, type is one of **M1** (for metal-one), **M2** (for metal-two), **CO** (for contact) or **VI** (for via), *width* and *length* are in μm (for interconnects) and *area* (for contacts and vias) is in μm^2. The location of the connection is given in CIF coordinates *xloc* and *yloc*.

sim2spice also produces a file *basename*.**spcnode** containing node numbers in the SPICE deck and their locations in the layout. The list is written in CIF format. It can be read by **cif2ps** to make a plot showing the circuit with the named nodes superimposed.

An example of the command line is:

```
vivante>sim2spice circ.sim
```

which will read **circ.sim**, **circ.nodes** and produce **circ.spice**, **circ.spcnode** and **circ.geo**.

4.6 Plotting CIF file

The program **cif2ps** converts CIF layout file to PostScript file which can be sent to a PostScript printer for printout. The operation of **cif2ps** is described in the accompanying manual page (see Appendix G). The following command line:

```
vivante>cif2ps -t scmos circ.cif | lpr
```

will plot **circ.cif** (which contains CIF layer names of scmos technology). **cif2ps** is technology

dependent: the layers known to cif2ps are listed in Appendix B. Appendix C describes the setting up of different CIF layer names for **mextra** and **cif2ps**.

The **-m** option of **cif2ps** can be used to superimpose CIF labels on CIF layout. This is useful in locating the SPICE node numbers assigned by **sim2spice**. Recall that **sim2spice** writes the locations of nodes in CIF format to *basename*.**spcnode** file. This file can be printed together with the original CIF layout file *basename*.**cif** using **cif2ps -m**. For example:

```
vivante>cif2ps -m -t scmos circ.cif circ.spcnode | lpr
```

will print the SPICE node numbers on the layout *circ.cif*.

The EM simulator produces a listing of metal structures that pose reliability hazards in CIF format. This listing can be printed on the CIF layout using the same procedure:

```
vivante>cif2ps -m -t scmos circ.cif circ.worst | lpr
```

circ.worst contains the listing of the worst metal structures.

V. Setting up Geometry File by Hand

5.1 Introduction

The user can input the layout geometry by hand as an alternative to using the layout extractor. The use of the layout extractor might be cumbersome because it generates a lot of data and will consume considerable computation time in the SPICE simulation. For layout which uses a large number of standard cells, such as gate arrays and memory chips, user is advised to set up the geometry file by hand. Because of the repetition of standard cells and only the current waveforms in one cell are needed, user can set up subcircuit elements (using the **SUBCKT** card in SPICE) for all but one of the standard cells in the circuit. This implicitly instructs the EM simulator not to perform failure calculation for the metal connections in the subcircuit element. As a result, this will reduce the amount of cpu time used in the EM and SPICE analysis. The user can do this by the following procedures:

(1) User enters the layout geometry of one cell by hand (or just have the layout extractor extract one cell).

(2) User then sets up the subcircuit definition using the **SUBCKT** card in SPICE. All but one cell is replaced by subcircuit call. (using the **X***zzz* element in SPICE).

(3) Layout geometry information is entered into the geometry file for that cell only. User is required to identify any "stacked" connections in the circuit. These are the connections that simultaneously feed current to a number of cells (for example: power, ground and clock lines). An example of "stacked" connection is explained in Sec.5.3.

5.2 The format of user-entered geometry file

The pre-processor of the EM simulator reads the user-entered geometry file *geometryfile* that appears on the **.EMSTAT** card in the SPICE input deck. This file has a different format from the *geometryfile* produced by the extractor program. The format is such that user can easily enter and read the layout information in this file.
The format for capacitance, inductance, resistance, voltage and current (and other two-node) elements is:

element [*type* {*width length*} | *openings*] ...

type is one of MF, MS, MT, VI, CO which represents metal-one, metal-two, metal-three, via or contact. For interconnect, the user enters the width and length in μm. The number of contact or via openings is entered for contact and via (the area of the opening is taken to be the same as the area of contact/via opening entered in the **Area=** card for the test structure, see Sec. 3.6). There is no requirement on the order of appearance for the connection type (i.e. whether MF should be first, followed by VI and MS, etc.). But all information for one element must appear in one line only. For example, a capacitance element (labeled C10 in the SPICE deck) which has a first level metal connection 4μm wide 100μm long connected to second level metal (2μm wide 20μm long) through via with 2 via openings can be specified as:

C10 MS 2.0 20.0 MF 4.0 100.0 VI 2.0

A single line with no type specified is treated by the simulator as a blank line, i.e. no failure statistics calculation will be performed for connection to this element. For example, simulator will ignore this line:

C10

The format for transistor element and for subcircuit elements is:

element [**node** *nodenumber* [*type* {*width length*} | *openings*] ...]

type is one of MF, MS, MT, VI, CO which represents metal-one, metal-two, metal-three, via or contact. For interconnect, the user enters the width and length in μm. The number of contact or via openings is entered for contact and via (the area of the opening is taken to be the same as the

area of contact/via opening entered in the Area= card for the test structure, see Sec. 3.6). There is no requirement on the order of appearance for the connection type (i.e. whether MF should be first, followed by VI and MS, etc.). But all information for one element must appear in one line only. For example, a bipolar transistor (three terminal device) (labeled Q1 in the SPICE input deck) with the following metal connections:

(1) Emitter (node number 1) contacting polysilicon,

(2) Base (node number 2) has 1 contact opening to first level metal connection 2μm wide 5μm long.

(3) Collector (node number 3) has 3 contact openings to first level metal connection 2.5μm wide 15μm long. The first level metal later connects to second level metal interconnect 5.0μm wide 100μm long through a via with 2 via openings

can entered in the geometry file as follows:

Q1 node 3 CO 3.0 MF 2.5 15.0 VI 2.0 MS 5.0 100.0 node 2 CO 1.0 MF 2.0 5.0 node 1

A node number not followed by the type, width and length or number of openings will be ignored by the pre-processor, as in last entry of the above example. The last node entry is redundant (but can be kept for better readability) and can be removed:

Q1 node 3 CO 3.0 MF 2.5 15.0 VI 2.0 MS 5.0 100.0 node 2 CO 1.0 MF 2.0 5.0

Comments can be inserted in the file but typing * in the first column. For example:

* This is a comment line

5.3 "Stacked" metal connections

A "stacked" metal connection occurs when a power line (or signal line) successively connects to a number of identical cells (for example, see the 21-stage BiCMOS inverter chain in Fig. 3). The metal segments between two adjacent cells all have the same length and width. As a result, the average current density flowing in the metal segments "stacks" from the farthest cell (from the power or signal source) to the nearest cell. As noted earlier (in Sec.5.1), the user can set up subcircuit definition for the cell and repeat the subcircuits in the SPICE deck. But to account for the "stacked" connection, user have to set up a dummy resistor between two adjacent cells to attach geometry information. This can be quite troublesome if the number of cells is very large.

The simulator can be requested to treat "stacked" metal connection. The user can attach a suffix S after *type* followed by a "stacking" parameter *nSTACK* when entering the geometry information by hand. For example:

M1 node 2 MFS 21 10.0 30.0

is a "stacked" metal-one connection of 21 segments at node 2 of element M1. Each segment is 10 μm wide 30.0 μm long. In calculating the failure statistics, the simulator will sum the failure rate of first segment carrying current density J (from SPICE simulation), second segment carrying current density $2 \times J$, and so on, up to and including the *nStack*-th segment carrying current density $nStack \times J$.

VI. Output from the Simulator

6.1 Introduction

The EM simulator can function in two modes: to provide layout guidelines or to predict failure rate or cumulative percent failure of a circuit. Both modes of operation can be requested simultaneously. In the first mode, it generates a layout advisory table listing maximum length allowed for each width of interconnect, the safety factor for number of contact and via openings given the reliability requirement. In the second mode, it calculates the electromigration failure rate and/or cumulative percent failure for each element in the circuit.

6.2 Current Table

Normally the current table is not printed, it is in the output if user request **PrintCurrent** in the rule file. It is printed in the first part of the output. The table lists average current and average absolute current (both in units of A) for each connection in the circuit. For some connections, where the average current < AC_DEFINE × average absolute current, the simulator performs calculation assuming the current waveform is pure AC (by setting the internal average current to zero). Those connections are indicated by **AC** in the current table. It is a good idea to check this table to make sure that the assumptions for pure AC waveform are valid. This is particularly important for signal lines in MOS circuit where the gate current is supposed to be pure AC waveform. If a particular connection to a purely capacitive node is not indicated by **AC** in the output, the layout guideline and the failure rate calculated for this connection will be too pessimistic (assuming **Aac** > **Adc** in the rule file).

6.3 Layout advisory tables

The second part of the output has two layout advisory tables. The first one is generated unless **SKIPLAYOUTCUR** is specified in the rule file. This table lists:

(1) the maximum interconnect length L (in μm) for metal-one, metal-two or metal-three allowed for a given DC current (in A).

(2) the safety factor S for contacts or vias (of different number of openings) for a given DC current (in A). If a number of less than 1.0 is printed, the number of contact/via openings carrying the given total DC current is not sufficient to guarantee the reliability spec and more contact/via openings are needed.

This table, although generated for a given DC current, can be used as a guideline for pulse DC current (if **m** in Eq.(1) is equal to 2). This is because for **m**=2 and under pulse DC current, the only current density dependence in Eq.(1) is J. Therefore, the designer can just read off the average current from the top of the table. For the case of pure AC current, if **m** is equal to 2, Eq.(1) will simplify to:

$$\mathrm{TTF} = \frac{A_{AC}}{\overline{|J|}^2}$$

or

$$= \frac{A_{DC}}{\left[\sqrt{\frac{A_{DC}}{A_{AC}}}\,\overline{|J|}\right]^2}$$

To use the table for pure AC current waveforms, the current at the top of the table will have to be multiplied by $\sqrt{\frac{A_{AC}}{A_{DC}}}$. For example, **Aac**=1.0e20 and **Adc**=1.0e16, the table result for DC current of 0.1mA will be the same for AC current (the average of the absolute current) of 10.0mA.

The second layout advisory table gives a list of maximum interconnect length L (in μm) allowed and the safety factor S for contacts or vias (given the number of openings) in the circuit. The guideline is given for a number of specified widths of interconnects and number of openings of contacts and vias. The user can specified (by using **NWIDTH** and **NCV** in the rule file) widths of interconnects or the number of contact/via openings the simulator should consider.

The safety factor S for contacts and vias should be interpreted as follows: it is the maximum number of contacts/vias in a chain that is allowed along the path of the current flow. For example, if 3 is printed for 2 via openings, the number metal-to-metal vias in series (2 via openings in parallel at each terminal) along the current flow path can not exceed 3. A number of less than 1.0 for a particular number of contact/via openings indicates that the number of openings is not sufficient to guarantee the reliability spec.

If user enters the "stacked" parameter for a connection (See Sec.7.4), the guidelines generated are to be applied to one connection. (For example, the L printed for 21 "stacked" metal line segments should be the maximum length for one segment).

This table will not be printed if **SKIPLAYOUTGEO** is specified in the rule file. In this table, if the current density in a connection is smaller than *MinJCurrent* (specified using **MINJCURRENT=** in the rule file), the table entry is $J < MinJ$.

6.4 Failure rate statistics

The final part of the output contains the projected electromigration failure statistics of the design. This part is requested by **EMSTAT** (if user set up the geometry file by hand) or **EMSTATX** (if the geometry file comes from the layout extractor). In this table, the failure rate and/or cumulative percent failure for each connection are calculated at times specified in the **EMSTAT** or **EMSTATX** card. **SKIPFAILPERCENT** and/or **SKIPFAILRATE** can be specified in the rule file to instruct the simulator to skip either or both tables. The user can select to print out some of the worst connections by using the **WorstList** card in the rule file, otherwise all connections will be printed. The first column of the table is the name given by the simulator for the connection (useful for identifying trouble spots in CIF file) followed by the node connection in the SPICE input deck. At the bottom of the table the total failure rate/cumulative percent failure for the design is given. If **EMSTATX** is specified, the simulator will print out the location of of connections that pose electromigration hazards in CIF format. The listing consists of the connection names in descending order of failure rates and the ranking is attached to the name separated by a dash. User can then cut out the CIF format output and superimpose the names on the original CIF layout file for printout.

VII. Examples of Simulation

7.1 Introduction

In this section, we will demonstrate the operation of the EM simulator in four examples:

(1) To design a CMOS EPROM sense amplifier circuit using the layout guideline from the simulator and make an estimation of the failure rate for the 512K × 8 EPROM.

(2) To assess the failure rate of a 21-stage BiCMOS inverter chain. The layout information in entered by hand. The use of subcircuits to represent each inverter cell and "stacked" connection for the power supply line is demonstrated.

(3) To project the reliability of a CMOS logic circuit designed using arrays of inverter, NOR and NAND gates. The SPICE input deck is extracted from the CIF layout file.

(4) To project the reliability of a CMOS fulladder circuit. The SPICE input deck is extracted from the CIF layout file.

The SPICE input decks and CIF layout files are found in **BERT/EM/examples** in the distribution tape.

7.2 The design rule and reliability parameters

The EM design rule file for all the examples described below is **emrule**. The line:

.EMMODEL emrule

is inserted in the SPICE input deck. The listing of *emrule* is given below:

```
* Reliability parameters for electromigration simulator
PrintCurrent
ncurrent=3 1.00e-4 1.00e-3 5.00e-3
SkipFailpercent
WorstList=0.5
AC_define=0.2
MinJcurrent=1e2
* this is the reliability specifications
nwidth=4 1.0 2.0 4.0 10.0
ncv=3 1.0 2.0 4.0
* The following set for metal-one interconnect
metal1 length=4.5e+4 width=1.0 thick=0.5
Tdata=200.0 Ea=0.5
width_a=6.25 width_b=1.0 width_c=40.0
m=2.0 Adc=7.52e+12 Aac=7.52e+16
Lognormal logsigma=1.0 logmedian=7.52
* The following set for metal-one to silicon contact
contact area=1.0 Nchain=10.0
Tdata=200.0 Ea=0.5
m=2.0 Adc=1.0e+13 Aac=9.0e+14
Lognormal logsigma=1.0 logmedian=7.52
spec_time=1.0e+4 spec_failrate=1.0e-9
```

In the example, a listing of current in each connection is requested. For the layout advisory part, the simulator is instructed to generate guidelines for four interconnect widths: 1μm, 2μm, 4μm and 10μm, and three contact and via openings: 1, 2 and 4. It is also requested that layout guidelines are generated for three current values: 0.1 mA, 1.0 mA and 5.0 mA. Simulator will ignore any connections with current density less than 1e2 A/cm^2. Pure AC waveform is defined if the

average current is less than 20% of the average absolute current. The width dependence is extracted from the example in Appendix B. The metal-one line used in electromigration lifetime experiment is 4.5 cm long, 2 μm wide and 0.5 μm thick. A_{DC}, A_{AC} and m are obtained from accelerated testing at 200°C. Lognormal failure distribution is to be used. For example, $\sigma = 1.0$ and MTF = 7.52 hours (using $A_{DC}=7.52\times10^{12}$, the DC stress current is 10^6 A/cm^2 at 200°C in the experiment). The contact chain structure used in the experiment has ten 1μm^2 contact openings in series. Finally, the reliability requirement for the circuit is failure rate of 1 FIT (10^{-9} failure / device hour) at 10^4 hours (1.1 year).

The parameters for metal-two, metal-three and metal-to-metal vias are not given. The simulator will not perform analysis for these structures.

7.3 CMOS EPROM sense amplifier

In this example, the designer wishes to generate a layout guideline for a CMOS EPROM sense amplifier circuit (see Fig.2) and later estimate the electromigration reliability of a 512K×8 EPROM design.

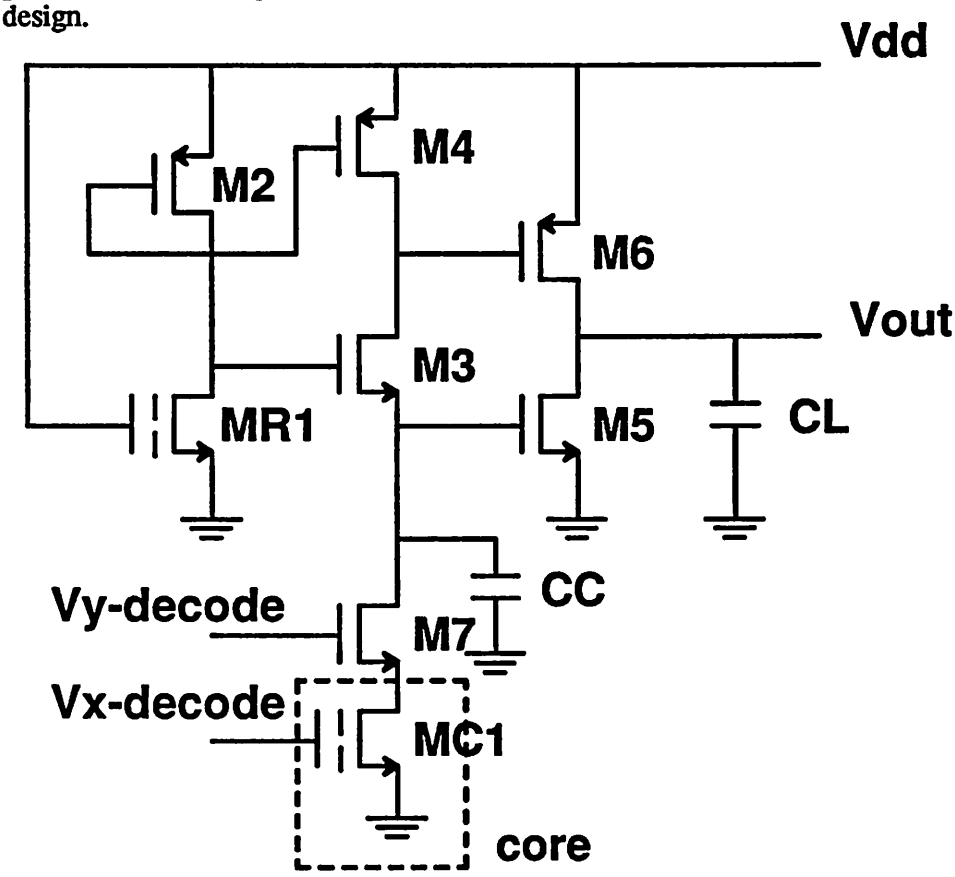

Figure 2. EPROM sense amplifier

The SPICE input deck (in file eprom) is set up as follows:

```
CMOS EPROM SENSE AMPLIFIER

VDD 5 0 DC 5
VX 6 0 DC 0 PWL(0 0 5NS 5 55NS 5 60NS 0 150NS 0 155NS 5 205NS 5
+ 210NS 0 300NS 0)
```

```
VY 7 0 DC 0 PWL(0 0 5NS 5 55NS 5 60NS 0 150NS 0 155NS 5 205NS 5
+ 210NS 0 300NS 0)


MR1 2 5 0 0 EPROM L=2U W=3U
+ AD=20P AS=20P PD=14U PS=14U NRD=1.25 NRS=1.25
M2 2 2 5 5 MODP L=2U W=11U
+ AD=60P AD=60P PD=22U PS=22U NRD=0.42 NRS=0.42
M3 3 2 1 0 MODN L=2U W=39U
+ AD=200P AD=200P PD=50U PS=50U NRD=0.13 NRS=0.13
M4 3 2 5 5 MODP L=2U W=7U
+ AD=40P AD=40P PD=18U PS=18U NRD=0.63 NRS=0.63
M5 4 1 0 0 MODN L=2U W=49U
+ AD=250P AD=250P PD=60U PS=60U NRD=0.1 NRS=0.1
M6 4 3 5 5 MODP L=2U W=11U
+ AD=60P AD=60P PD=22U PS=22U NRD=0.42 NRS=0.42
M7 1 7 8 0 MODN L=2U W=49U
+ AD=250P AD=250P PD=60U PS=60U NRD=0.1 NRS=0.1
MC1 8 6 0 0 EPROM L=2U W=3U
+ AD=20P AS=20P PD=14U PS=14U NRD=1.25 NRS=1.25

CC 1 0 2P
CL 4 0 0.3P

.MODEL MODN NMOS (LEVEL=2 VTO=0.9 KP=36U GAMMA=0.16 PHI=0.58 LAMBDA=0.04
+  CGSO=2.3E-10 CGDO=2.3E-10 CGBO=1.0E-10 RSH=30 CJ=1E-4 CJSW=3E-10
+  JS=3E-9 TOX=300E-10 NSUB=1E15 NFS=2E10 XJ=0.3E-6 LD=0.2E-6 UO=310
+  VMAX=6E4)

.MODEL MODP PMOS (LEVEL=2 VTO=-0.9 KP=17U GAMMA=0.27 PHI=0.63 LAMBDA=0.06
+  CGSO=3.4E-10 CGDO=3.4E-10 CGBO=1.0E-10 RSH=120 CJ=1.7E-4 CJSW=6.7E-10
+  JS=1E-9 TOX=300E-10 NSUB=3E15 NFS=2E10 XJ=0.4E-6 LD=0.4E-6 UO=150
+  VMAX=8E4)

.MODEL EPROM NMOS (LEVEL=2 VTO=2.0 KP=14U GAMMA=0.16 PHI=0.58 LAMBDA=0.04
+  CGSO=2.3E-10 CGDO=2.3E-10 CGBO=1.0E-10 RSH=30 CJ=1E-4 CJSW=3E-10
+  JS=3E-9 TOX=300E-10 NSUB=1E15 NFS=2E10 XJ=0.3E-6 LD=0.2E-6 UO=310
+  VMAX=6E4)

.EMModel emrule
.WIDTH OUT=80
.TRAN 1NS 300NS 150NS
.OPTIONS NODE METHOD=GEAR VNTOL=0.001 ABSTOL=1.0E-8
.PLOT TRAN V(4) V(1) V(3) V(6) (0,5)
.END
```

We have set up the input voltages and transient analysis card such that SPICE will simulate the circuit for two read cycle (each read cycle is 150 ns). Keep in mind that in the actual read cycle of the EPROM, only one out of many cells will be accessed. Since only layout guidelines are desired, we omit the .EMSTAT card in the SPICE deck. The command to call the simulator (using SPICE2G6) is:

```
vivante>prebert -2 eprom | spice2g6 | postbert > eprom.em
```

or using the "standalone" version of the EM simulator which saves the SPICE output:

```
vivante>preem eprom | spice2g6 > eprom.spcout
vivante>postem -S2 -G eprom.geo -R emrule eprom.spcout > eprom.em
```

A partial listing of the current table in **eprom.em** is given below:

<Current Table>

Connection	Avg Current(A)	Avg Abs Cur(A)	
VDD to 5	2.56e-04	2.56e-04	
MR1 to 2	1.12e-04	1.12e-04	
MR1 to 5	1.49e-10	2.22e-09	AC
MR1 to 0	1.12e-04	1.12e-04	
M2 to 2	1.11e-04	1.11e-04	
M2 to 2	2.98e-09	4.47e-08	AC
M2 to 5	1.11e-04	1.11e-04	
M5 to 4	1.09e-04	1.09e-04	
M5 to 1	4.80e-07	2.29e-06	
M5 to 0	1.09e-04	1.10e-04	
M6 to 4	1.07e-04	1.07e-04	
M6 to 3	3.48e-07	1.12e-06	
M6 to 5	1.06e-04	1.06e-04	
M7 to 1	2.98e-05	3.64e-05	
M7 to 7	2.19e-06	2.38e-05	AC
M7 to 8	2.98e-05	3.03e-05	
MC1 to 8	2.98e-05	3.03e-05	
MC1 to 6	9.04e-08	1.72e-06	AC
MC1 to 0	2.98e-05	3.02e-05	

The simulator indicates that the current in gate of MR1, gate of M2, gate of M7 and gate of MC1 is treated as pure AC current. But, note that this is not the case for the current in gate of M5 and gate of M6. SPICE simulation covering more cycles might thus be needed to obtain steady state current waveforms or **AC_Define** can be set at a higher value.

The first layout advisory table gives the maximum allowable length for interconnect, and the safety factor for contacts and vias as a function of current:

<Layout Advisory Table for Current>

Maximum Interconnect Length (in micron) or Safety Factor for Contacts or Vias to guarantee 1.00e-09(/h) failure rate at time 1.00e+04(h) for given Current(A)

DC Current(mA)	1.0e-04	1.0e-03	5.0e-03
M1 W = 1.0	2.2e+13	2.0e+02	5.6e-01
M1 W = 2.0	9.4e+20	8.9e+05	8.0e+00
M1 W = 4.0	1.9e+35	2.5e+14	1.9e+05
M1 W = 10.0	1.9e+60	3.8e+31	9.6e+16
No.of CONT=1.0	9.5e+15	3.9e+01	9.7e-04
No.of CONT=2.0	1.3e+22	9.0e+04	2.7e-02
No.of CONT=4.0	1.2e+29	1.4e+09	4.9e+00

For example, the maximum length allowed for 2μm wide metal-one interconnect carrying 5 mA current is 8μm long. And the safety factor for 4 contact openings is 4.9 if 5 mA of current flows through the contact (each contact opening carries 1.25 mA).

The second table gives layout guideline for every connection in the circuit. The following is a partial listing (the least constraining guidelines have been omitted):

<Layout Advisory Table for all connections>
Maximum Interconnect Length (in micron) or Safety Factor for Contacts or Vias to guarantee 1.00e-09(/h) failure rate at time 1.00e+04(h) in each connection

Metal: (Width)	Type	1.0e+00	2.0e+00	4.0e+00	1.0e+01
Cont/Via:(#)	Type	1.0e+00	2.0e+00	4.0e+00	
VDD to 5	MF	5.4e+07	5.4e+13	4.5e+25	4.5e+30
VDD to 5	CO	1.0e+09	1.0e+14	7.0e+19	
MR1 to 2	MF	3.9e+12	1.1e+20	4.5e+30	4.5e+30
MR1 to 2	CO	1.2e+15	1.2e+21	1.0e+27	
MR1 to 5	MF	J <MinJ	J <MinJ	J <MinJ	J <MinJ
MR1 to 5	CO	J <MinJ	J <MinJ	J <MinJ	
MR1 to 0	MF	3.9e+12	1.1e+20	4.5e+30	4.5e+30
MR1 to 0	CO	1.2e+15	1.2e+21	1.0e+27	
M2 to 2	MF	4.2e+12	1.1e+20	4.5e+30	4.5e+30
M2 to 2	CO	1.2e+15	1.3e+21	1.0e+27	
M2 to 2	MF	J <MinJ	J <MinJ	J <MinJ	J <MinJ
M2 to 2	CO	J <MinJ	J <MinJ	J <MinJ	
M2 to 5	MF	4.2e+12	1.1e+20	4.5e+30	4.5e+30
M2 to 5	CO	1.2e+15	1.3e+21	1.0e+27	
M5 to 4	MF	6.1e+12	1.9e+20	4.5e+30	4.5e+30
M5 to 4	CO	2.0e+15	2.2e+21	1.0e+27	
M5 to 1	MF	4.5e+30	4.5e+30	4.5e+30	J <MinJ
M5 to 1	CO	1.0e+27	1.0e+27	J <MinJ	
M5 to 0	MF	5.4e+12	1.6e+20	4.5e+30	4.5e+30
M5 to 0	CO	1.7e+15	1.8e+21	1.0e+27	
M6 to 4	MF	8.2e+12	2.7e+20	4.5e+30	4.5e+30
M6 to 4	CO	2.8e+15	3.2e+21	1.0e+27	
M6 to 3	MF	4.5e+30	4.5e+30	J <MinJ	J <MinJ
M6 to 3	CO	1.0e+27	J <MinJ	J <MinJ	
M6 to 5	MF	8.6e+12	2.8e+20	4.5e+30	4.5e+30
M6 to 5	CO	3.0e+15	3.4e+21	1.0e+27	
M7 to 1	MF	9.1e+21	4.5e+30	4.5e+30	4.5e+30
M7 to 1	CO	1.6e+26	1.0e+27	1.0e+27	
M7 to 7	MF	4.5e+30	4.5e+30	4.5e+30	4.5e+30
M7 to 7	CO	1.0e+27	1.0e+27	1.0e+27	
M7 to 8	MF	5.8e+22	4.5e+30	4.5e+30	4.5e+30
M7 to 8	CO	1.0e+27	1.0e+27	1.0e+27	
MC1 to 8	MF	5.8e+22	4.5e+30	4.5e+30	4.5e+30
MC1 to 8	CO	1.0e+27	1.0e+27	1.0e+27	
MC1 to 6	MF	4.5e+30	4.5e+30	J <MinJ	J <MinJ
MC1 to 6	CO	1.0e+27	J <MinJ	J <MinJ	
MC1 to 0	MF	6.2e+22	4.5e+30	4.5e+30	4.5e+30

MC1 to 0	CO	1.0e+27	1.0e+27	1.0e+27

As pointed out earlier, because the gate current in M5 and M6 from SPICE output is not pure AC, the guideline calculated for the gate of M5 (M5 to 1) and gate of M6 (M6 to 3) is too pessimistic.

From the table above, the designer can layout the circuit and also estimate the failure rate of the layout using the table. The interconnect failure rate for the design at 10^4 hours is:

$$\text{Designed FIT} = \frac{\text{Layout Length}}{L} \times 1 \text{ FIT}$$

where L is maximum length allowed from the layout advisory table. 1 FIT is used on the R.H.S. of the equation because the layout guidelines are generated for the spec of 1 FIT at 10^4 hours. The contact/via failure rate for the design is:

$$\text{Designed FIT} = \frac{1}{S} \times 1 \text{ FIT}$$

where S is the the safety factor from the layout advisory table.

We will calculate the failure rate at 10^4 hours of a 512K× 8 EPROM assuming the worst case scenario: the same 8 EPROM bits located furthest away from the sense amplifier are continuously accessed in each read cycle. The geometry of interconnects and contacts in the chip are:

(1) All the transistors in the sense amplifier circuit (total of 8 sense amplifier circuits) have 20μm and 1μm wide metal-one lines connected to their source and drain.
Failure rate for these:

$$F_a = 8 \times 20 \times \left[\frac{1}{3.9e12} + \frac{1}{3.9e12} + \frac{1}{4.2e12} + \frac{1}{4.2e12} + \frac{1}{6.1e12} + \frac{1}{5.4e12} + \frac{1}{8.2e12} + \frac{1}{8.6e12} \right]$$

$$= 2.5 \times 10^{-10} = 0.25 \text{ FIT}$$

(2) Vdd line for each (total of 8) sense amplifier is 200 μm long 2μm wide.

$$F_b = 8 \times \frac{200}{5.4e13}$$

$$= 3.0 \times 10^{-11} = 0.03 \text{ FIT}$$

(3) The bitline (M7 to 8) (the same 8 connected to the EPROM bits are activated continuously at all time) is 1 μm wide 1 cm long.

$$F_c = 8 \times \frac{10^4}{5.8e22}$$

$$= 1.4 \times 10^{-18} = 0.0 \text{ FIT}$$

(4) In the core cell of the EPROM (the same 8 bits are accessed all the time) the gate, source and drain connections are each 3μm long 1μm wide.

$$F_d = 8 \times 3 \times \left[\frac{1}{5.8e22} + \frac{1}{6.2e22} \right]$$

$$= 8.0 \times 10^{-22} = 0.0 \text{ FIT}$$

(5) The source and drain of each transistor has one contact opening (of area $1\mu m^2$).

$$F_e = 8 \times \left[\frac{1}{1.2e15} + \frac{1}{1.2e15} + \frac{1}{1.2e15} + \frac{1}{1.2e15} + \frac{1}{2.0e15} + \frac{1}{1.7e15} + \frac{1}{2.8e15} + \frac{1}{3.0e15} + \cdots \right]$$

$$= 4.0 \times 10^{-14} = 0.0 \text{ FIT}$$

Thus, the FIT of the design is:

$$\text{Designed FIT} = F_a + F_b + F_c + F_d + F_e = 0.28 \text{ FIT@ } 10^4 \text{ hours}$$

We can obtain a better estimation by assuming a certain probability that the bit is accessed and the associated bit line is activated, and run the SPICE simulation over a duration that covers both the active cycles and the idle cycles. This will produce a more accurate average current flowing in each connection. For example, if we know on the average each EPROM cell will be accessed once every 100 read cycles and each bit line is activated once every 10 read cycles. A SPICE deck can be set up consisting three EPROM bits (bit A, bit B and bit C). Bit A and bit B are on the same bitline, bit C is on a different bitline (but connects to the same sense amplifier). Bit A is accessed once and left idle for 99 cycles, bit B is accessed for 9 cycles and idle for 91 cycles and bit C is accessed for 90 cycles and left idle in the 10 cycles when A and B are activated. The three bits are set up to continuously provide signal to the sense amplifier and to activate bit A's bitline once every 10 cycles. The transient duration requested in the SPICE input deck will be 100 cycles. The total failure rate can be calculated by summing the failure rate of all the bits (i.e. multiply the failure rate of bit A by 512K × 8), the failure rate of the bit lines (i.e. multiply the failure rate of bit A's bitline by the total number of bit lines) and failure rate of 8 sense amplifiers.

7.4 21-Stage BiCMOS inverter chain

In this example, the simulator is used to generate layout guidelines and to calculate the failure rate of a 21-stage BiCMOS inverter chain (Fig.3). The SPICE deck is found in file **bicmos**:

```
21-stage BiCMOS output inverter
vdd 98 0 dc 5.5
vin 4 0 dc 0.0 pulse (0.0 5.0 1n 10p 10p 7n 15n)
rdummy1 98 1 0.001
rdummy2 99 0 0.001
q1 1 2 5 0 npn1
q2 5 3 99 0 npn1
m1 1 4 2 1 pmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
m2 1 1 2 1 pmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
m3 2 4 6 0 nmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
m4 6 1 99 0 nmos w=15u l=1.2u ad=30p as=60p pd=4u ps=23u
m5 5 4 7 99 nmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
m6 7 1 3 0 nmos w=15u l=1.2u ad=30p as=60p pd=4u ps=23u
m7 3 5 99 0 nmos w=15u l=1.2u ad=60p as=60p pd=23u ps=23u
cl 5 0 0.5p

x1 5 8 1 0 inverter
x2 8 9 1 0 inverter
x3 9 10 1 0 inverter
```

```
x4 10 11 1 0 inverter
x5 11 12 1 0 inverter
x6 12 13 1 0 inverter
x7 13 14 1 0 inverter
x8 14 15 1 0 inverter
x9 15 16 1 0 inverter
x10 16 17 1 0 inverter
x11 17 18 1 0 inverter
x12 18 19 1 0 inverter
x13 19 20 1 0 inverter
x14 20 21 1 0 inverter
x15 21 22 1 0 inverter
x16 22 23 1 0 inverter
x17 23 24 1 0 inverter
x18 24 25 1 0 inverter
x19 25 26 1 0 inverter
x20 26 27 1 0 inverter

.subckt inverter 4 5 1 99
* 4 is input 5 is output 1 is +supply 99 is -supply
qs1 1 2 5 0 npn1
qs2 5 3 99 0 npn1
ms1 1 4 2 1 pmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
ms2 1 1 2 1 pmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
ms3 2 4 6 0 nmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
ms4 6 1 99 0 nmos w=15u l=1.2u ad=30p as=60p pd=4u ps=23u
ms5 5 4 7 99 nmos w=15u l=1.2u ad=60p as=30p pd=23u ps=4u
ms6 7 1 3 0 nmos w=15u l=1.2u ad=30p as=60p pd=4u ps=23u
ms7 3 5 99 0 nmos w=15u l=1.2u ad=60p as=60p pd=23u ps=23u
csl 5 0 0.5p
.ends
.nodeset v(5)=4.796 v(8)=0.531 v(9)=4.796 v(10)=0.531 v(11)=4.796 v(12)=0.531
+ v(13)=4.796 v(14)=0.531 v(15)=4.796 v(16)=0.531 v(17)=4.796 v(18)=0.531
+ v(19)=4.796 v(20)=0.531 v(21)=4.796 v(22)=0.531 v(23)=4.796 v(24)=0.531
+ v(25)=4.796 v(26)=0.531 v(27)=4.796
.width out=80

.model npn1 npn (is=5.0e-18 bf=86 br=10 cje=11f cjc=15f cjs=135f
+ tf=5p vaf=40 rb=50 re=25 rc=180 ikf=0.01)
.model nmos nmos (level=2 vto=0.58 kp=80u gamma=0.16 phi=0.58 lambda=0.04
+  cgso=2.3e-10 cgdo=2.3e-10 cgbo=1.0e-10 rsh=30 cj=1E-4 cjsw=3e-10
+  js=3e-9 tox=200e-10 nsub=1e15 nfs=2e10 xj=0.3e-6 ld=0.0e-6 uo=310
+  vmax=6e4)
.model pmos pmos (level=2 vto=-0.52 kp=27u gamma=0.27 phi=0.63 lambda=0.06
+  cgso=3.4e-10 cgdo=3.4e-10 cgbo=1.0e-10 rsh=120 cj=1.7e-4 cjsw=6.7e-10
+  js=1e-9 tox=200e-10 nsub=3e15 nfs=2e10 xj=0.4e-6 ld=0.0e-6 uo=150
+  vmax=8e4)
.emmodel emrule
.emstat bicmos.int 1.0e2 5.0e2 1.0e3 2.0e3

.tran 100p 20n
.plot tran v(27)
.end
```

Because the circuit has 21 identical inverter cells, we set up the subcircuit for the inverter stage. All but one of the 21 inverter cells are represented by the subcircuit elements. To account for the "stacked" Vcc power line, we inserted a dummy resistance rdummy1 in the first cell to represent the connection. In the geometry file, the connection of rdummy1 is indicated to be metal-one 7.5μm wide and 30μm long and with a "stack" parameter of 21. The same procedure is applied to the ground line of the first cell. The geometry file set up by the user in **bicmos.int** is:

```
rdummy1 MFS 21 7.5 30.0
rdummy2 MFS 21 7.5 30.0
q1 node 1 CO 2.0 node 2 CO 2.0 MF 2.0 10.0 node 5 CO 2.0 MF 2.0 10.0
q2 node 5 CO 2.0 MF 2.0 10.0 node 3 CO 2.0 MF 2.0 10.0 node 99 CO 2.0
m1 node 1 CO 2.0 MF 2.0 10.0 node 4 CO 2.0 MF 2.0 15.0 node 2 CO 2.0
m2 node 1 CO 2.0 MF 2.0 10.0 node 1 CO 2.0 node 2 CO 2.0
m3 node 2 CO 2.0 MF 2.0 10.0 node 4 CO 2.0 MF 2.0 10.0 node 6
m4 node 6 node 1 CO 2.0 node 99 CO 2.0 MF 2.0 10.0
m5 node 5 CO 2.0 MF 2.0 10.0 node 4 CO 2.0 MF 2.0 10.0 node 7
m6 node 7 node 1 CO 2.0 node 3 MF 4.0 25.0 CO 2.0
m7 node 3 CO 2.0 MF 2.0 10.0 node 5 CO 2.0 MF 2.0 10.0 node 99 CO 2.0
c1 MF 2.0 10.0
```

The command line to invoke the simulator is:

```
vivante>prebert bicmos | spice3c1 | postbert > bicmos.em
```

or using the "standalone" version of the simulator:

```
vivante>preem bicmos | spice3c1 > bicmos.spcout
vivante>postem -G bicmos.geo -R emrule bicmos.spcout > bicmos.em
```

The failure rate table calculated by the simulator in **bicmos.em**:

<Failure Rate (/hour) Table>

No	Conn. at time	1.0e+02	1.0e+03	1.0e+04	1.0e+05
MFS0	rdummy1 to 98	2.1e-11	2.3e-08	2.4e-07	1.6e-07
MFS1	rdummy2 to 99	1.4e-25	3.4e-18	4.5e-13	3.6e-10
C5	q1 to 5	2.2e-33	6.4e-24	9.1e-17	6.4e-12
MF6	q1 to 5	6.6e-31	1.6e-22	2.0e-16	1.2e-12
MF15	m1 to 4	5.3e-36	1.6e-26	2.5e-19	1.9e-14
C16	m1 to 2	2.8e-38	6.9e-28	8.4e-20	5.1e-14
MF27	m4 to 99	2.4e-36	8.0e-27	1.3e-19	1.1e-14
MF29	m5 to 5	1.8e-37	1.0e-27	2.7e-20	3.7e-15
Total Failure Rate(/h):		2.1e-11	2.3e-08	2.4e-07	1.6e-07

The worst electromigration hazard in the design is the Vcc and ground lines. Since we have only simulated one of the 21 stages. The failure rate at 10^4 hours (1.1 year) for the circuit is the sum of the failure rate for the supply and ground lines and 21× of the total failure rate for each cell:

$$\text{Total Failure Rate} = \Sigma_{cell} F_i + F_{vdd} + F_{ground}$$
$$= 21 \times 2.5 \times 10^{-16} + 2.4 \times 10^{-7} + 4.5 \times 10^{-13}$$
$$= 2.4 \times 10^{-7} = 2.4 \times 10^{2} \text{ FIT @ } 10^{4} \text{ hours}$$

Note that the failure rate of the Vcc line decreases after 10^4 hours. This is because failure rate for a lognormal distribution can first increase, reach a maximum and decrease. Therefore, the failure statistics beyond 10^4 hours might be incorrect to use.

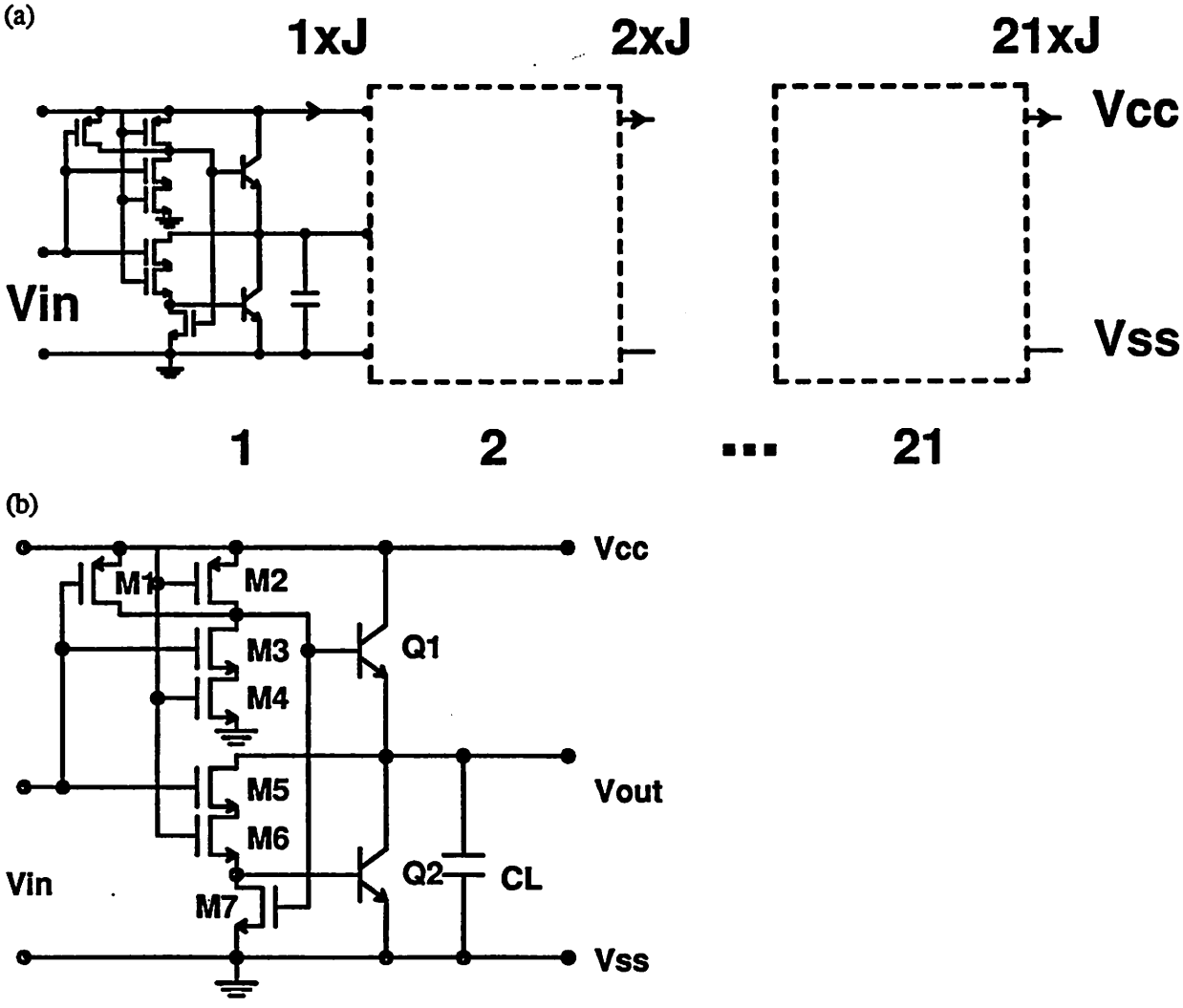

Figure 3. (a) 21-stage BiCMOS inverter chain. An example of "stacked" connection is shown. The first segment of the power line in the first cell on the left carries current density J, and the current density increases towards the Vcc source. The last segment of the power line carries 21× current density. (b) the circuit of one inverter stage.

7.5 CMOS logic circuit using inverters, NOR and NAND gates.

The circuit shown in Figure 4 is designed using a scmos process. The CIF file for the layout has been generated (see Fig. 5) and is in file circ.cif. The reliability of the layout design is to be assessed using the EM simulator. First, the SPICE deck is extracted from the CIF layout. The following commands are used to invoke the extractor which also produces the layout geometry

information:

```
vivante>mextra -t scmos circ.cif
vivante>sim2spice circ.sim
```

Input voltages, power supply voltages and ground nodes have to be added to complete the SPICE deck (in **circ.spice**). The completed SPICE deck is in file **circ.ospice**. In order to trace the input nodes and power buses, the extracted SPICE nodes (which is written to **circ.spcnode** by **sim2spice**) is printed on top the original CIF layout file using **cif2ps** with the option **-m**:

```
vivante>cif2ps -m -t scmos circ.cif circ.spcnode | lpr
```

The output is sent to a PostScript printer and the printout is shown in Figure 5. Polysilicon, diffusion and well layers have one node number assigned to each electrically connected area. The number is usually printed at the center lower edge of the layer. Node numbers on metal layers are found at the corners, intersections or contacts where the extractor fractures the interconnect. At contacts and vias, two node numbers are assigned, the node number for layer on top of the contact is printed at the upper right corner of the contact or via. The node number for the lower layer is usually at the lower right.

The commands for the simulator are:

```
vivante>prebert circ.ospice | spice3c1 | postbert > circ.em
```

or using the "standalone" version of the EM simulator:

```
vivante>preem circ.ospice | spice3c1 | postem -G circ.geo -R emrule > circ.em
```

The simulator prints a listing of the worst reliability hazards (the number of connections printed is determined by the **WorstList=** card in the rule file) after the failure rate or cumulative percent failure table. In the listing, the name of the connection (beginning with **C,V, MF** or **MS** for contact, via, metal-one, metal-two respectively) are given with its location in CIF coordinates.

The following is the list of worst 10% (**WorstList=** 0.1) of the connections from **circ.em**. The connection name is followed by dash and ranking number. The lines starting with **DS** and ending with **E** can be saved to a file and printed out on top of the original CIF file using **cif2ps**. The failure rate (/hour) of the connection is enclosed in brackets in the following line:

```
DS 94 1 1;
9 WORST_STAT;
94 C63-1 -1650 23700;
( 5.70e-39 )
94 C42-2 -1650 15750;
( 5.40e-39 )
94 C21-3 -1650 7800;
( 5.30e-39 )
94 C0-4 -1650 -175;
( 5.10e-39 )
94 C64-5 -450 23700;
( 1.40e-40 )
94 C43-6 -450 15750;
( 1.30e-40 )
94 C22-7 -450 7800;
( 1.30e-40 )
94 C1-8 -450 -175;
```

```
( 1.20e-40 )
94 MF215-9 187 24900;
( 8.90e-42 )
94 MF173-10 187 16950;
( 8.40e-42 )
94 MF133-11 187 9000;
( 8.10e-42 )
94 MF93-12 187 1050;
( 8.10e-42 )
94 C70-13 1050 24675;
( 1.10e-42 )
94 C49-14 1050 16725;
( 9.90e-43 )
94 C28-15 1050 8775;
( 9.60e-43 )
94 C7-16 1050 825;
( 9.50e-43 )
94 MF202-17 2550 22237;
( 5.60e-44 )
94 MF154-18 2550 14287;
( 5.30e-44 )
94 MF120-19 2550 6337;
( 5.20e-44 )
94 MF80-20 2550 -1612;
( 5.00e-44 )
94 MF124-21 16050 6337;
( 3.60e-44 )
94 MF206-22 16050 22237;
( 3.40e-44 )
94 MF158-23 16050 14287;
( 3.40e-44 )
DF;
C 94;
E
```

This listing is cut to file **circ4.worst** and printed together with the CIF file (see Fig.6) by:

```
vivante>cif2ps -m -t scmos circ.cif circ.worst | lpr
```

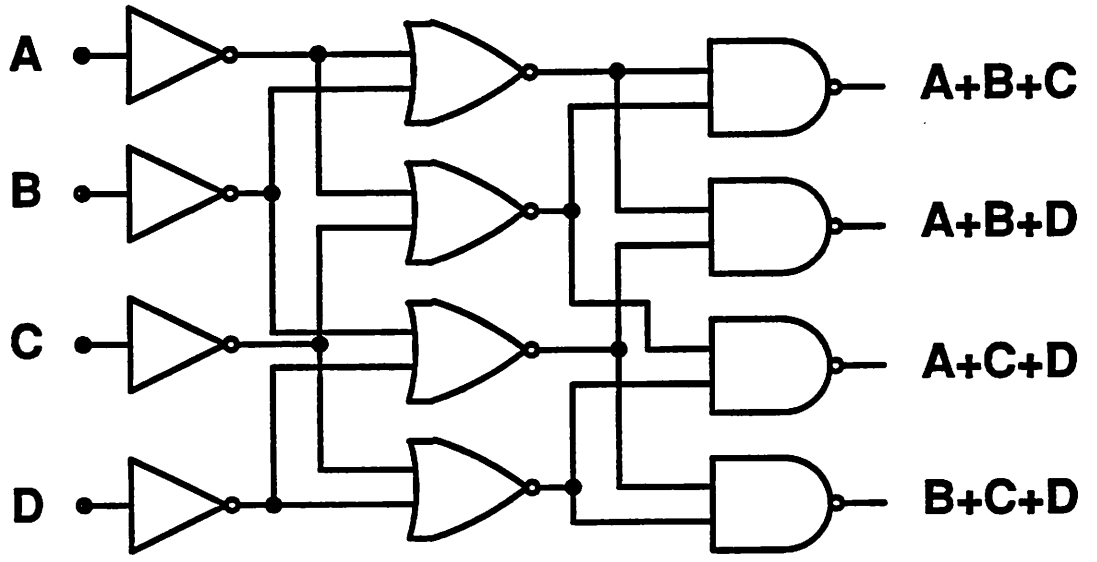

Figure 4. CMOS logic circuit using inverters, NOR and NAND gates

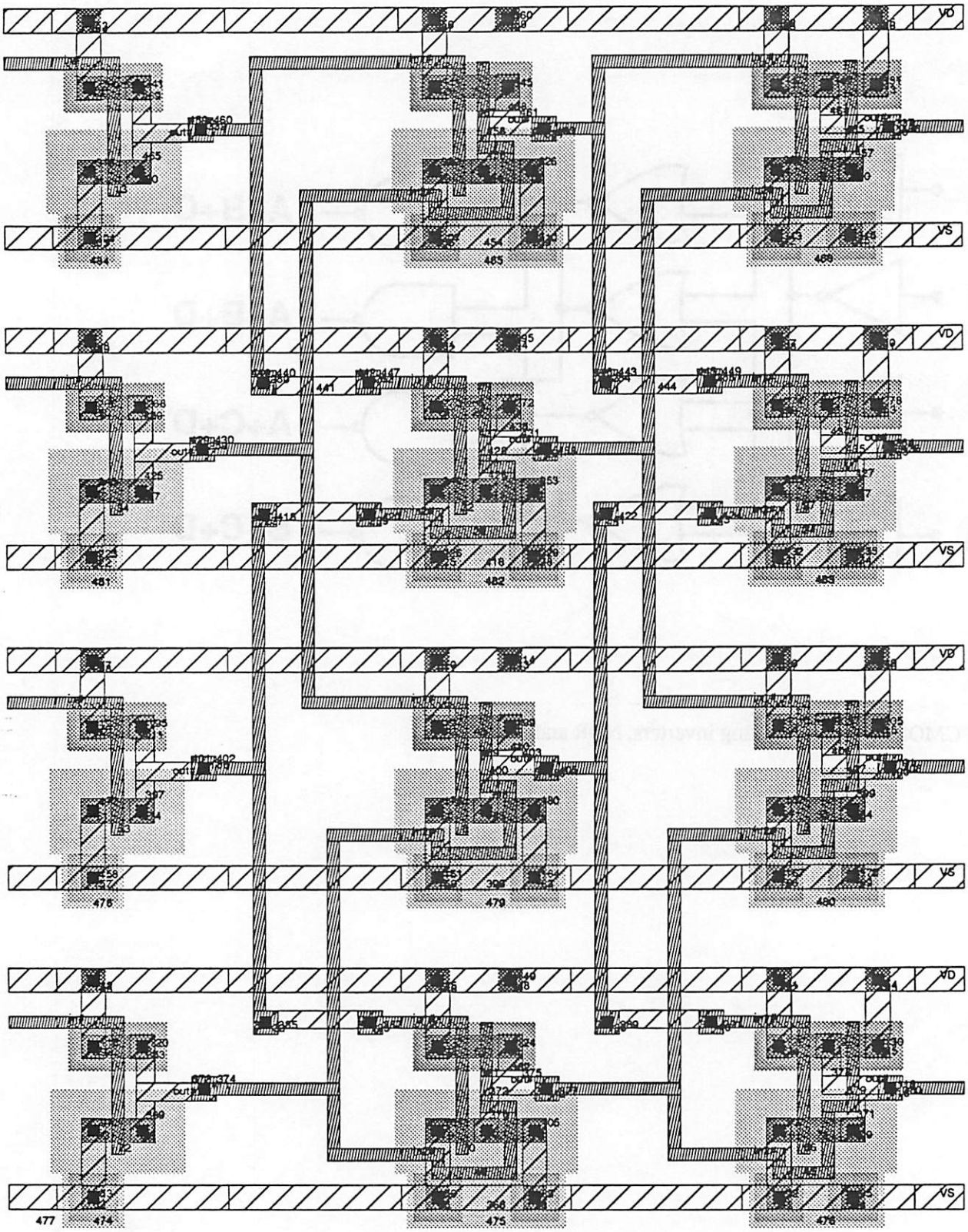

Figure 5. SPICE node numbers are plotted on top the **circ.cif** file using **cif2ps**

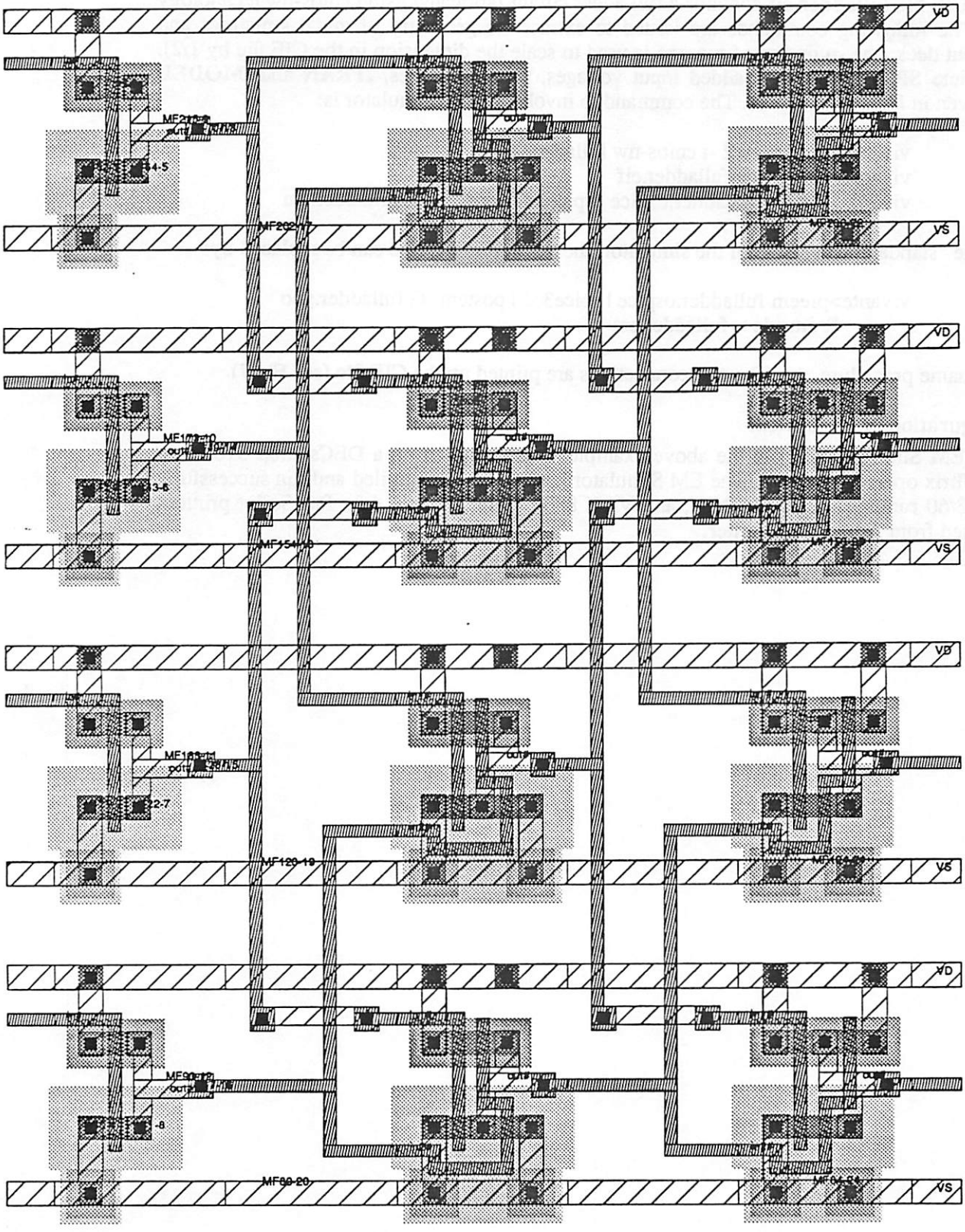

Figure 6. The worst 10% of the connections from EM Simulator in **circ.cif**.

7.6 CMOS fulladder circuit

The CIF file **fulladder.cif** contains a full adder circuit fabricated using cmos-nw technology (Fig. 7). The following commands are issued to extract the geometry of metal structures and SPICE input deck (the **-u** option of **mextra** is used to scale the dimension in the CIF file by 1/2). The complete SPICE deck with added input voltages, power supplies, **.TRAN** and **.MODEL** cards is given in **fulladder.ospice**. The command to invoke the EM simulator is:

```
vivante>mextra -u 2 -t cmos-nw fulladder.cif
vivante>sim2spice fulladder.cif
vivante>prebert fulladder.ospice | spice3c1 | postbert > fulladder.em
```

or using the "standalone" version of the simulator, the last command line can be replaced by:

```
vivante>preem fulladder.ospice | spice3c1 | postem -G fulladder.geo
      -R emrule > fulladder.em
```

Using the same procedure, the 10 worst connections are printed on the CIF file (see Fig.7)

7.7 Configuration of our system

The EM Simulator used in the above examples was compiled on a DECstation 3100 running the Ultrix operating system. The EM Simulator has also been compiled and run successfully on a Sun 3/60 running 4.2 BSD UNIX, and VAX 8800 running Ultrix 4.3. PostScript printout was obtained from Apple LaserWriter.

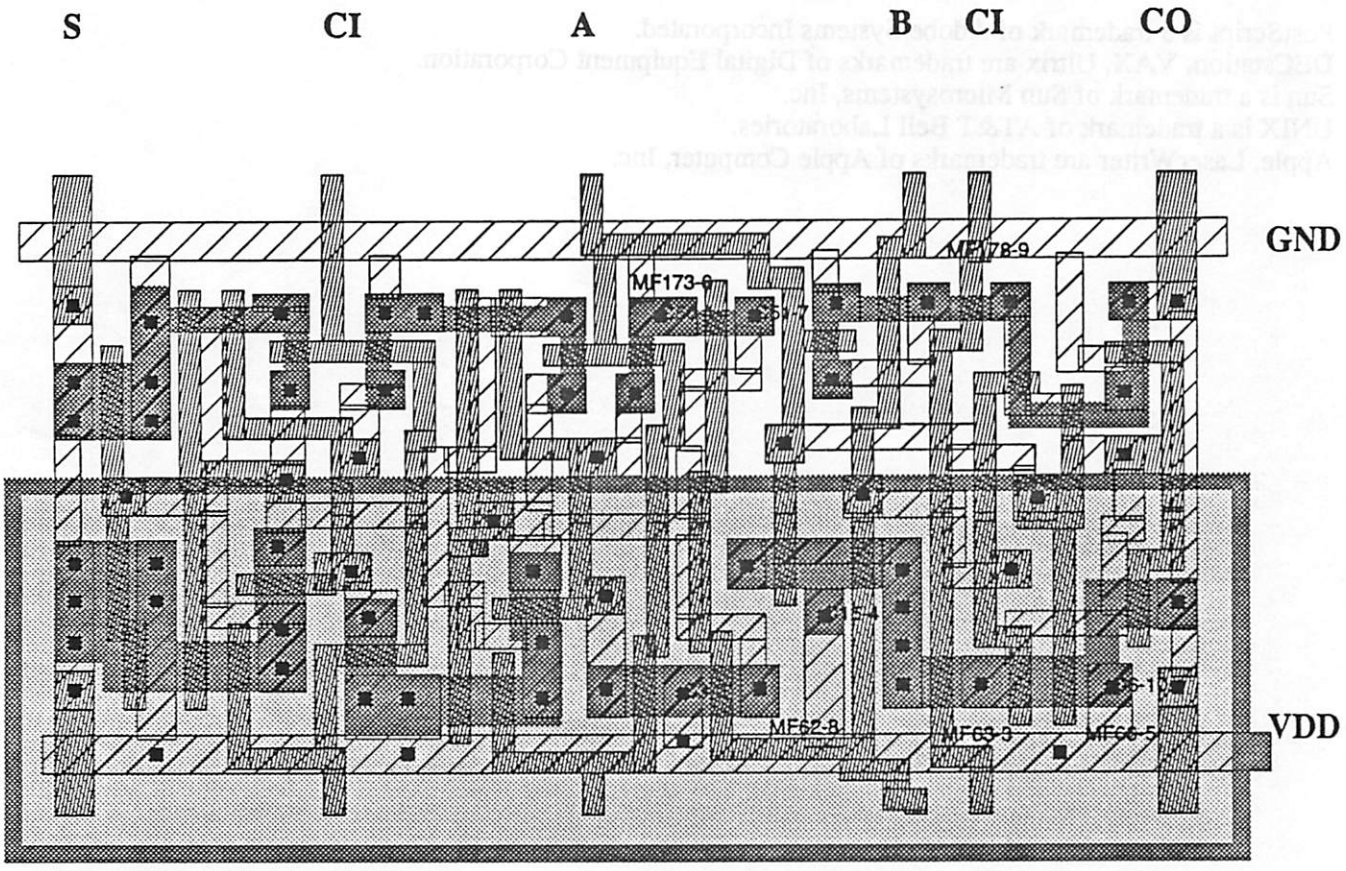

Figure 7. Full adder circuit with 10 worst connections from EM Simulator.

VIII. Credits and Acknowledgement

The original layout extractor **mextra** was the work of D. T. Fitzpatrick (UCB) and modifications to support metal-two were added by W. E. Winder (Northwest LIS, University of Washington). **sim2spice** was originally written by D. T. Fitzpatrick and fixed by N. Soiffer (UCB). The first version **cif2ps** was obtained from A. Simoneau (Aerospace Corporation, California) and M. Lesure (Arizona State University).

IX. References

[1] B. K. Liew, N. W. Cheung, and C. Hu, "Electromigration interconnect lifetime under AC and pulse DC stress," *Proc. 29th Int. Reliab. Phys. Symp. IEEE*, p. 215, 1989.

[2] B. K. Liew, N. W. Cheung, and C. Hu, "Projecting interconnect electromigration lifetime for arbitrary current waveforms," to be published in *IEEE Trans. on Electron Devices*,

[3] B. K. Liew, P. Fang, N. W. Cheung, and C. Hu, "Circuit reliability simulator for interconnect and contact electromigration," to be published in *Proc. 30th Int. Reliab. Phys. Symp. IEEE*,

[4] P. M. Lee, M. M. Kuo, P. K. Ko, and C. Hu, "BERT - Circuit Aging Simulator (CAS)," *UCB/Electronics Research Lab. M90/2*, Electronics Research Laboratory of University of California, Berkeley, Jan 1990.

[5] E. Rosenbaum, P. M. Lee, R. Moazzami, P. K. Ko, and C. Hu, "BERT - Circuit Oxide Reliability Simulator (CORS)," *UCB/Electronics Research Lab. M90/4*, Electronics Research Laboratory of University of California, Berkeley, Jan 1990.

[6] B. N. Agarwala, M. J. Attardo, and A. P. Ingraham, "Dependence of electromigration-induced failure time on length and width of aluminum thin-film conductor," *J. Appl. Phys.*, vol. 41, no. 10, p. 3954, 1979.

[7] E. Kinsbron, "A model for the width dependence of electromigration lifetimes in aluminum thin-film stripes," *J. Appl. Phys.*, vol. 36. no. 12. p. 968, 1980.

[8] A. V. Ferris-Prabhu, "Reliability Modeling," *Int. Reliab. Phys. Symp. Tutorial Notes*, p. 3-1, Apr 1984.

[9] D. T. Fitzpatrick, "Mextra: A Manhattan Circuit Extractor," *UCB/Electronics Research Lab. M82/42*, Electronics Research Laboratory of University of California, Berkeley, May 1982.

Appendix A. Lognormal Distribution and Length Dependence Model

The following example illustrates the conflict in choosing a lognormal distribution and using the length dependence model. The experimental data (indicated by closed circles) for a long test line ($2\times10^4\mu$) is given in Figure ?. The failure probability for this line can be described by a lognormal distribution function $F(t)$ with σ=1.0 and MTF=1000. From this data, the simulator can construct a failure distribution for a line half the length of the test line, i.e. at any time t the $1\times10^4\mu$ line has a failure probability $G_2(t)$ equal to:

$$G_2(t) = 1 - [1 - F(t)]^{1/2}$$

For example at time $t = 1000$, the failure probability for this line is 0.29. Similarly, the failure distribution of lines 1/3 of the long test line $G_3(t)$ can be constructed. In general, for a line that is $1/x$ of the long test line, $G_x(t)$ is:

$$G_x(t) = 1 - [1 - F(t)]^{1/x}$$

Figure 9 shows the plot of $G_2(t)$, $G_3(t)$, $G_5(t)$ and $G_{10}(t)$ on a lognormal paper. It is clearly shown that as the line become shorter, the extrapolated failure distribution deviates more from lognormal function.

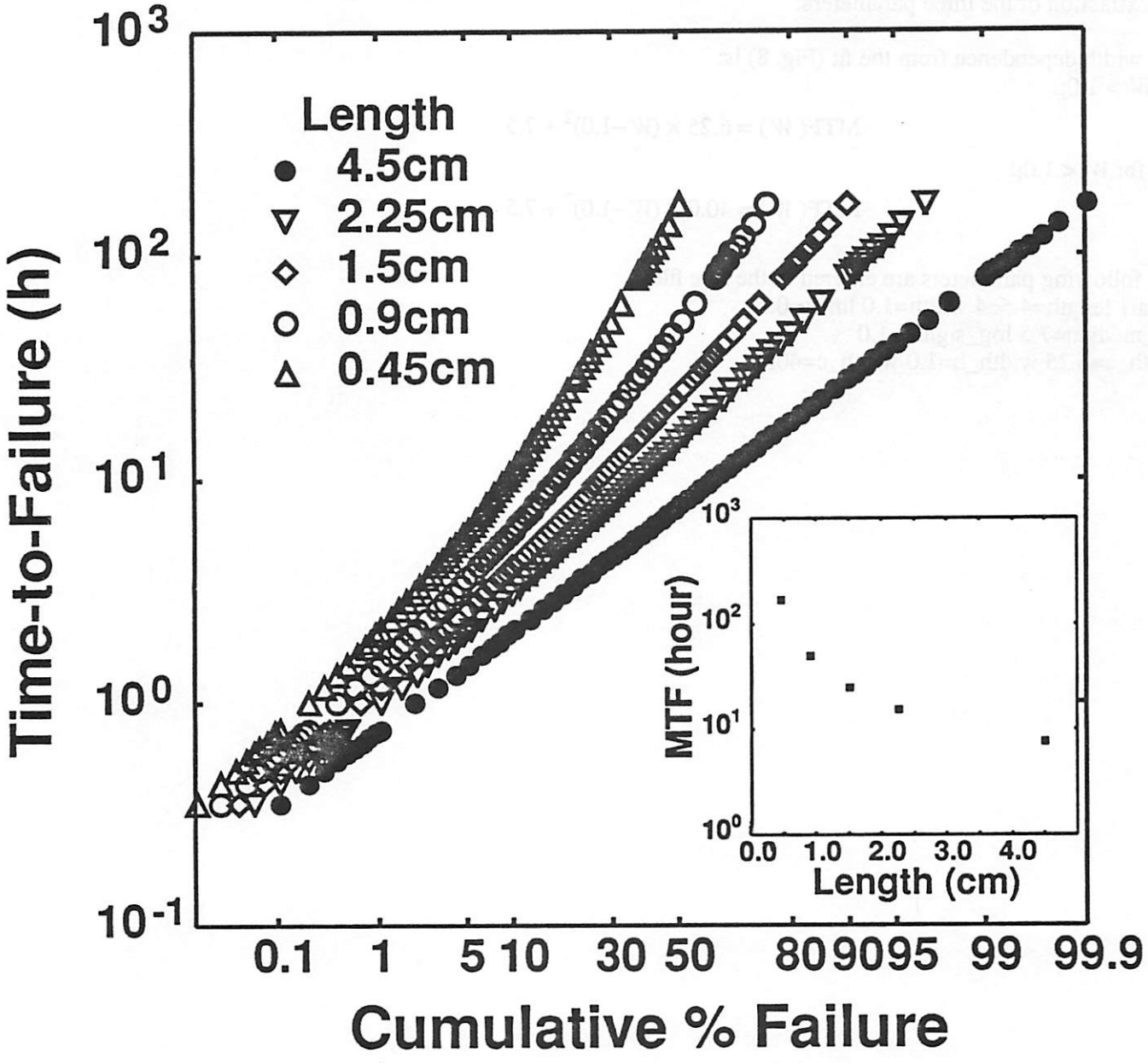

Figure 8. The failure data from the 4.5 cm long interconnect is assumed to be lognormally distributed. The plot shows deviation from the lognormal distribution when failure statistics for shorter lines are calculated from the length dependence model. The calculated MTF versus length is plotted in the inset.

Appendix B. Width Dependence Parameters

The simulator accepts two parameters for the dependence of interconnect time-to-failure on width: A_w, B_w and C_w which are entered using WIDTH_A, WIDTH_B and WIDTH_C in the rule file. B_w is the linewidth where the test data shows an increase in TTF. We will demonstrate the extraction of the three parameters:

The width dependence from the fit (Fig. 8) is:
for $W > 1.0\mu$

$$\text{MTF}(W) = 6.25 \times (W-1.0)^2 + 7.5$$

and for $W < 1.0\mu$

$$\text{MTF}(W) = 40.0 \times (W-1.0)^2 + 7.5$$

The following parameters are entered in the rule file:
metal1 length=4.5e4 width=1.0 thick=0.5
log_median=7.5 log_sigma=1.0
width_a=6.25 width_b=1.0 width_c=40.0

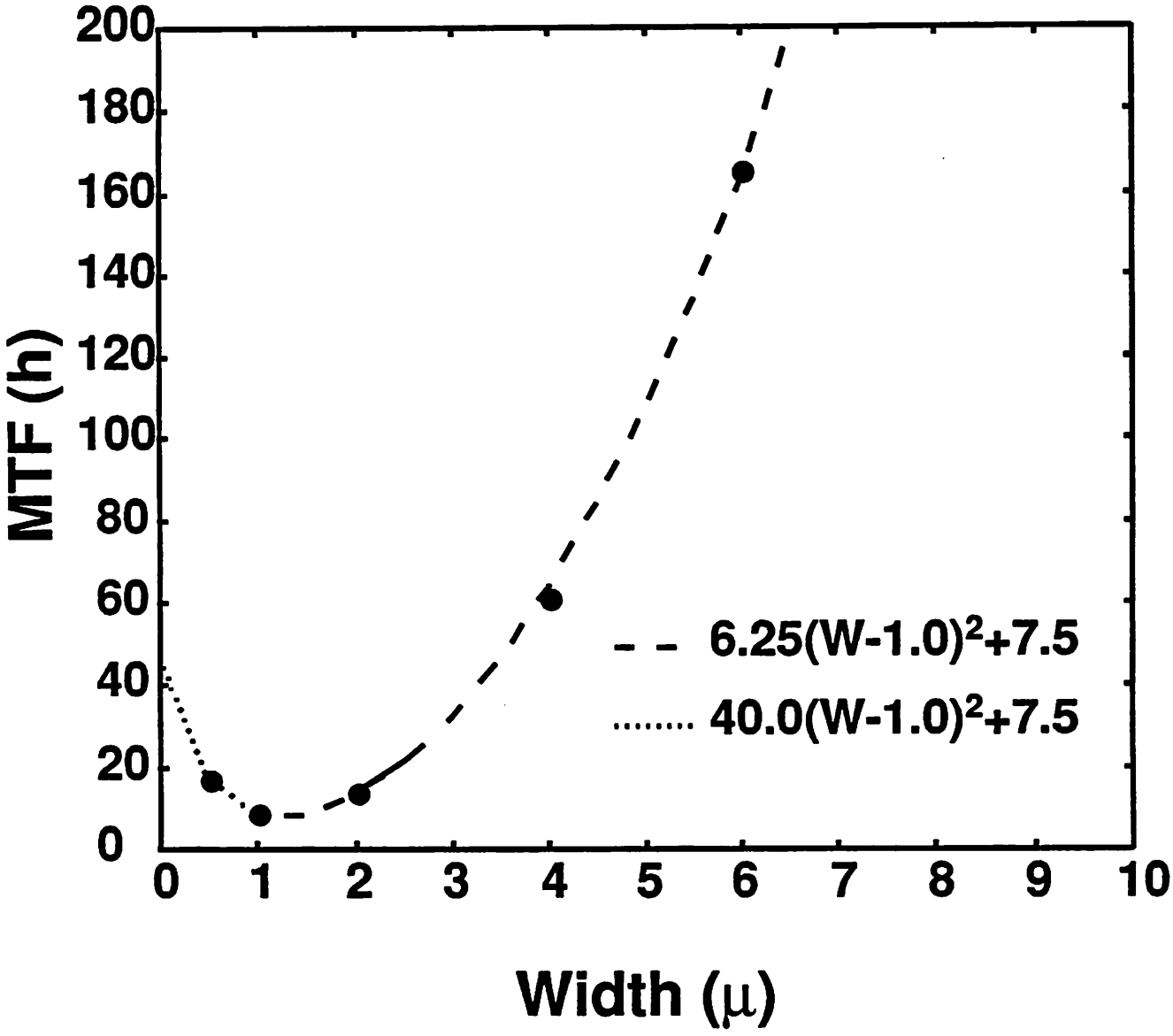

Figure 9. Extracting width dependence parameters for the simulator

Appendix C. Default Values in EM Simulator

The following if not specified by the user in the rule file will have the default values:

ncurrent=3 0.50e-3 1.00e-3 2.00e-3
WorstList=1.0
AC_define=0.0
MinJCurrent=1e2
Top=25.0
NCurrent=3 0.5e-3 1.0e-3 2.0e-3
NWidth=4 1.0 2.0 4.0 10.0
NCV=2 1.0 2.0
Spec_Time= 1.0e4
Spec_Failrate= 1.0e-9
metal1 Width_A= 0.0 Width_B=0.0 Width_C=0.0
metal2 Width_A= 0.0 Width_B=0.0 Width_C=0.0
metal3 Width_A= 0.0 Width_B=0.0 Width_C=0.0
via nchain=1.0
contact nchain=1.0

Appendix D. CIF Layer Names in the Layout Extractor

The technologies known to mextra are: nMOS ("nmos"), MOSIS P well CMOS/Bulk, also known as CBPM ("cmos-pw" or "cmos-p"), MOSIS Scalable CMOS/Bulk N-well, also known as SCN ("cmos-nw" or "cmos-n"), MOSIS Scalable CMOS/Bulk P-well, also known as SCP ("cmos-s" or "scmos"), and MOSIS Scalable CMOS/Bulk Generic, also known as SCG ("cmos-g"). The CIF layer names for each technology are listed in the two tables below:

Technology					
nmos		cmos-p/cmos-pw		cmos-n/cmos-nw	
cif layer	mextra/cif2ps internal names	cif layer	mextra/cif2ps internal names	cif layer	mextra/cif2ps internal names
NM	METAL	CW	WELL	CWN	WELL
NP	POLY	CM	METAL	CMS	METAL2
ND	DIFF	CM2	METAL2	CMF	METAL
NC	CUT	CP	POLY	CPG	POLY
NB	BURIED	CP2	POLY2	CAA	DIFF
NI	ION	CD	DIFF	CVA	CUT2
NG	GLASS	CC	CUT	CCA	CUT
		CC2	CUT2	CSP	CION
		CS	CION	CSN	GLASS
		CG	GLASS	COG	GLASS
				XP	GLASS

Technology					
cmos-s/scmos		cmos-g		isocmos	
cif layer	mextra/cif2ps internal names	cif layer	mextra/cif2ps internal names	cif layer	mextra/cif2ps internal names
CWP	WELL	CWN	WELL	CPW	WELL
CMS	METAL2	CMS	METAL2	CM	METAL
CMF	METAL	CMF	METAL	CM2	METAL2
CPG	POLY	CPG	POLY	CP	POLY
CAA	DIFF	CAA	DIFF	CP2	NOP
CVA	CUT2	CVA	CUT2	CD	DIFF
CCA	CUT	CCA	CUT	CC	CUT
CCP	CUT	CCP	CUT	CC2	CUT2
CSP	CION	CSP	CION	CPP	CION
COG	GLASS	CWP	GLASS	CNP	NOP
XP	GLASS	CSN	GLASS	CG	GLASS
		COG	GLASS	CS	GLASS
		XP	GLASS		

Appendix E. Setting up Alternative Technology for the Layout Extractor

The mask names that **mextra** uses in generating geometry information for electromigration simulator are: **METAL** for metal-one, **METAL2** for metal-two, **CUT** for metal-one to poly or diffusion contact, **CUT2** for metal-one and metal-two via. To determine whether a transistor is PMOS or NMOS in a CMOS process, **mextra** uses: **POLY** for polysilicon, **DIFF** for diffusion (both p+ and n+), and **WELL** for either p-well or n-well. A PMOS transistor for example can be found when a polysilicon line (**POLY**) is found adjacent to the diffusion (**DIFF**) in the n-well (**WELL**) in a n-well CMOS process. For NMOS process, the mask **ION** will be checked to see if the transistor is subjected to depletion implant. Buried contacts are found from **BURIED** mask.

User can create a set of new CIF layer names by using the **-L** *ciflayerfile* option in both **mextra** and **cif2ps**. The format for *ciflayerfile* is:

(1) **wellohmic** *type*
Example:

wellohmic P

type is one of **N**, **P** or **0** for n-well, p-well or nmos process.

(2) *internal_layer_name cif_layer_name*
Example:

METAL CM5

The *cif_layer_name* is associated with a **mextra/cif2ps** *internal_layer_name.*

One example of *ciflayerfile* is:

```
wellohmic N
WELL CW1
POLY CP2
DIFF CD3
CUT CC4
METAL CM5
CUT2 CV6
METAL2 CM7
GLASS CSN
GLASS COG
GLASS XP
```

Note that internal layers **GLASS** and **NOP** are dummy layers in **mextra** and **cif2ps.**
The command line:

```
vivante>cif2ps -L cifnames fulladder.cif | lpr
```

will use the CIF layer names defined in **cifnames** to plot **fulladder.cif.**

Technology defined by user:

Technology name	
WellOhmic	
WELL	
METAL	
METAL2	
POLY	
POLY2	
DIFF	
CUT	
CUT2	
BURIED	
ION	
CION	
GLASS	
GLASS	
GLASS	

Appendix F: EM Simulator Error Messages

Error messages from the pre-processor:

(1) EM01: Could not open input file in EMPreFilter
Input file is not available to prebert

(2) EM10: Could not open input file
Could not open input file in EMSetup

(3) EM20: Could not open user set up geometry file
Could not open user set up geometry file when .EMSTAT is specified

(4) EM30: Spice Deck error at element *xxx*
User has set up SPICE deck incorrectly at element *xxx*

(5) EM31: Exceeding MAXXNODE at element *xxx*
Pre-processor can not handle a subcircuit card with more than MAXXNODE, recompile with higher value of MAXXNODE

(6) EM40: Could not write to temporary geometry file
Pre-processor fails to open a temporary geometry file for writing

(7) EM41: Could not write geometry file
Pre-processor fails to open a temporary geometry file for reading

(8) EM50: Error in user set up geometry file at element *xxx*
User makes an error in setting up the geometry file at element *xxx*

Error messages from the post-processor:

(1) EM101: Could not open EM design rule file
Could not open design rule file in EMPost

(2) EM102: Could not open geometry file
Could not open geometry file in EMPost

(3) EM103: Could not open input file
Could not open input file in EMPost

(4) EM104: Could not open temporary output file
Could not write to temporary output file in EMPost

(5) EM105: Could not open simulator output file
Could not write to simulator output file in EMPost

(6) EM106: Could not write to ciffile
Could not write to ciffile file in EMPost

(7) EM107: Could not write to rate statfile
Could not write to rate statfile in EMPost

(8) EM108: Could not write to percent statfile
Could not write to percent statfile in EMPost

(9) EM109: Could not write to current table file
Could not write to current table file in EMPost

(10) EM120: Exceeding MAXXNODE at element
Post-processor can not handle a subcircuit card with more than MAXXNODE, recompile with higher value of MAXXNODE

(11) EM130: Node *x* not found in geometry file (LookUpNode)
This is unusual because the geometry file is set up by the pre-processor. Check to see if the geometry file is the right one.

(12) EM140: Error in EM design rule file in line *x* at field *y*
User has made an error in EM design rule file in line *x* at field *y*.

(13) EM141: No one complete parameters set in EM design rule file
No one complete parameter set is found in the rule file

(14) EM150: Node *x* not found in geometry file (Failstat)
This is unusual because the geometry file is set up by the pre-processor. Check to see if the geometry file is the right one.

(15) EM151: Error in geometry file at *xxx* (Failstat)
This is unusual because the geometry file is set up by the pre-processor. Check to see if the geometry file is the right one.

Appendix G: Manual Pages for mextra, sim2spice and cif2ps

The following manual pages for **mextra**, **sim2spice** and **cif2ps** are available on line.

NAME

mextra, valtbs – Manhattan circuit extractor for VLSI simulation

SYNOPSIS

mextra [**–t** *tech*] [**–g**] [**–d** *temp_dir*] [**–c** *cadrc_line*] [**–f** *cadrc_file*] [**–u** *scale*] [**–O**] [**–o**] [**–W**] [**–w**] *basename*

valtbs *basename*.**tbs**

DESCRIPTION

Mextra will read the file *basename*.**cif** and create a circuit description. From this circuit description various electrical checks can be done on your circuit. The circuit description is directly compatible with *esim*, *powest*, and *erc*. There are translation programs to convert *mextra* output to acceptable *spice* input (see *sim2spice*, *pspice* and *spcpp*).

Mextra creates several new files, *basename*.**log**, *basename*.**al**, *basename*.**sim** and *basename*.**nodes**. It also creates *basename*.**tbs** (if -w option is used) or *basename*.**ohm** (if no -w option). After *mextra* finishes it is a good idea to read the *.log* file. This contains general information about the extraction. It has a count of the number of transistors and the number of nodes, and it contains messages about possible errors. The *.al* file is a list of aliases which can be used by *esim*. The *.tbs* file is a list of transistors in the *.sim* file that contains the substrate/well node name for each transistor (as the 5th field - see sim file description below). This file can be used to manually create spice input for analog circuits, such as sense amps, and to check substrate/well connectivity (see *valtbs* below). The *.ohm* file is used by *ohmics*. The *.nodes* file is a list of node names and their CIF locations listed in CIF format. It can be read by *cifplot* to make a plot showing the circuit with the named nodes superimposed. The form of this cifplot command is:

cifplot *basename*.**nodes** *basename*.**cif**

The .**sim** file is the circuit description for use with simulation programs and electrical rule checkers.

Names

Mextra uses the CIF label construct to implement node names and attributes. The form of the CIF label command is as follows:

94 name x y [layer];

This command attaches the label to the mask geometry on the specified layer crossing the point (x, y). If no layer is present then any geometry crossing the point is given the label.

Mextra interprets these labels as node names. These names are used to describe the extracted circuit. When no name is given to a node, a number is assigned to the node. A label may contain any ASCII character except space, tab, newline, double quote, comma, semi-colon, and parenthesis. To avoid conflict with extractor generated names, names should not be numbers or end in '#*n*' where *n* is a number.

A problem arises when two nodes are given the same name although they are not connected electrically. Sometimes we want these nodes to have the same names, other times we don't. This frequently happens when a name is specified in a cell which is repeated many times. For instance, if we define a shift register cell with the input marked 'SR.in' then when we create an 8 bit shift register we could have 8 nodes names 'SR.in'. If this happens it would appear as though all 8 of the shift register cells were shorted together. To resolve this the extractor recognizes three different types of names: *local*, *global*, and *unspecified*. Any time a local name appears on more than one node it is appended with a unique suffix of the form '#*n*' where *n* is a number. The numbers are assigned in scanline order and starting at 0. In the shift register example, the names would be 'SR.in#0' through 'SR.in#7'. Global names do not have suffixes appended to them. Thus unconnected nodes with global names will appear connected after extraction. (The –g causes the extractor to append unique suffixes to unconnected nodes with the same global name.) Names are made local by ending them with a sharp sign, '#'. Names are global if they end with an exclamation mark, '!'. These terminating characters are not considered part of the name, however. Names which do not end with these characters are considered unspecified. Unspecified names are treated similar to locals. Multiple occurrences are appended with unique suffixes. By convention, unspecified names signify the designer's intention that this name is a local name, but is connected to only one node. It is illegal to have a

name that is declared two different types. The extractor will complain if this is so and make the name local.

It makes no difference to the extractor if the same name is attached to the same node several times. However, if more than one name is given to a node then the extractor must choose which name it will use. Whenever two names are given to the same node the extractor will assign the name with the highest type priority, global being the highest, unspecified next, local lowest. If the names are the same type then the extractor takes the shortest name. At the end of the .log file the extractor lists nodes with more than one name attached. These lines start with an equal sign and are readable by *esim* so that it will understand these aliases.

Attributes

In addition to naming nodes *mextra* allows you to attach attributes to nodes. There are two types of attributes, *node attributes*, and *transistor attributes*. A node attribute is attached to a node using the CIF 94 construct, in the same way that a node name is attached. The node attribute must end in an at-sign, '@'. More than one attribute may be attached to a node. *Mextra* does not interpret these attributes other than to eliminate duplicates. For each attribute attached to a node there appears a line in the .sim file in the following form:

A *node attribute*

Node is the node name, and *attribute* is the attribute attached to that node with the at-sign removed.

Transistor attributes can be attached to the gate, source, or drain of a transistor. Transistor attributes must end in a dollar sign, '$'. To attach an attribute to a transistor gate the label must be placed inside the transistor gate region. To attach an attribute to a source or drain of a transistor the label must be placed on the source or drain edge of a transistor. Transistor attributes are recorded in the transistor record in the .sim file.

Transistors

For each transistor found by the exractor a line is added to the .sim file. The form of the line is:

type gate source drain length width x y
g=*attributes* s=*attributes* d=*attributes*

Type can be one of three characters, 'e' for enhancement, 'd' for depletion, or 'u' for unusual implant. (Unusual implant refers to transistors which are only partially in an implanted area. It will be necessary to write a filter to replace these transistors with the appropriate model in terms of enhacement and depletion transistors.) *Gate*, *source*, and *drain* are the gate, source, and drain nodes of the transistors. *Length* and *width* are the channel length and width in CIF units. *X* and y are the x and y coordinates of the bottom left corner of the transistor. *Attributes* is a comma seperated list of attributes. If no attribute is present for the gate, source, or drain, the g=, s=, or d= fields may be omitted.

The extractor guesses the length and width of a transistor by knowing the area, perimeter, and length of diffusion terminals. For rectangular transistors and butting transistors the reported length and width is accurate. For transistors with corners or for unusually shaped transistors the length and width is not as accurate.

It is possible to design a transistor with three or more diffusion terminals. The extractor considers these as *funny transistors*. They are entered in the .sim file in the form:

f*type gate node1 node2 ... nodeN xloc*

The 'f' is followed by the type : 'e', 'd' or 'u'. *Node1 ... nodeN* are the diffusion terminal nodes. As with any circuit with 'u' transistors, any circuit with 'f' transistors must be run through a filter replacing each of the funny transistors with the appropriate model in terms of enhancement and depletion transistors.

Capacitance

The .sim file also has information about capacitance in the circuit. The lines containing capacitance information are of the form:

C *node1 node2 cap-value*

cap-value is the capacitance betweens a node and substrate is in femto-farads. Capacitance values below a certain threshold are not reported. The default threshold is 50 femto-farads.

Transistor capacitances are not included since most of the tools that work on the .sim file calculate them from the width and length information.

The capacitance for each layer is calculated separately. The reported node capacitance is the total of the layer capacitances of the node. The layer capacitance is calculated by taking the area of a node on that layer and multiplying it by a constant. This is added to the product of the perimeter and a constant. The default constants are given below. Area constants are in femto-farads per square micron. Perimeter constants are femto-farads per micron.

Layer	Area	Perimeter
metal	0.03	0.0
metal2	0.015	0.0
poly	0.05	0.0
diff	0.10	0.1
poly/diff	0.40	0.0

Poly/diffusion capacitance is calculated similar to layer capacitance. The area is multiplied by constant and this is added to the perimeter multiplied by a constant. Poly/diffusion capacitance is not threshold, however.

The **–o** and the **–O** options are complementary. The **–o** option is the default. It supresses the calculation of capacitance, and instead, gives for each node in the circuit the area and perimeter of that node on the diffusion, poly, and metal layers. (The **-O** option causes the calculation of capacitance.) The lines containing this information look like this:

N *node diffArea diffPerim polyArea polyPerim metalArea metalPerim*

Node is the node name. *x y* is the position of a point on the node. Currently this is always '0 0'. *DiffArea* through *metalPerim* are the area and perimeter of the diffusion, poly, and metal layers in user defined units. (In addtion the **–o** [default] option causes transistors with only one terminal to be recorded in the .sim file as a transistor with source connected to drain.)

If the network is being extracted from the .cif file we suggest the node capacitance not be computed by mextra. Rather the *–o* [default] option should be used. This puts the burden of computing node capcitance on the programs *presim* and *sim2spice*. We feel this is advantageous because *presim* and *sim2spice* are filter programs linked directly to the type of simulation that is to be done. This will hopefully reduce some of the confusion associated with calibration.

Normally, mextra ignores connectivity through wells and substrate. If this connectivity is desired to give base connections of transistors, the *–w* option can be used. This allows one to check for proper connections of substrate and to detect shorts through wells and/or substrate. Since wells and substrate are highly resistive, it is desirable to detect what are, essentially, breaks in signals that travel through wells and substrate. This check should be part of the final check of a circuit before submission and should not be used for simulation, since, for instance, cmos-pw input pad circuitry is connected through a well. TO REITERATE: designs to be submitted should be simulated/compared using the **–w** and then reextracted normally and resimulated/compared to check for possible breaks in power/ground and other signals.

Changing Default Values

As part of its start up procedure *mextra* tries to read two files. It reads ~cad/.cadrc and then searches through the list of (colon separated) directories defined in environment variable CADRC until it finds a .cadrc file. This file is also read. If environment variable CADRC is not defined, *mextra* tries instead to read the .cadrc file in the home directory. *Mextra* reads these files to set up constants to be changed

without recompiling. The keywords for *mextra* are contained within the mextra environment of the .cadrc file. Declaration of environments in the .cadrc file are described in .cadrc(5). Cadrc lines may also be included on the command line by entering them as double quoted strings after the "-c" option. An entire cadrc file may also be read by including the option "-f <file>" on the command line, where the string <file> is the name of the cadrc file.

The temporary directory used may be set using the "-d" option. The string following this option is the temporary directory that will be used by mextra. The default is "/usr/tmp".

By default, *mextra* reports locations in CIF coordinates. A more convenient form of units may be specified either in the .cadrc file or on the command line. The form of the line in the .cadrc file is:

units *scale*

where *scale* is in centi-microns. The user may type in the chosen value for the scale directly.

To set units on the command line use the **–u** option.

mextra –u *scale basename*

The parameters used to compute node capacitance may be changed by including the following commands in your .cadrc file.

areatocap *layer value*
perimtocap *layer value*

value is atto-farads per square micron for area, and atto-farads per micron for perimeter. *layer* may be "poly", "diff", "metal", "metal2", or "poly/diff".

To set the capacitor values to those given in Mead and Conway the following lines would appear in the .cadrc file:

areatocap poly 40
areatocap diff 100
areatocap metal 30
areatocap poly/diff 400
perimtocap poly 0
perimtocap diff 0
perimtocap diff 0
perimtocap metal 0
perimtocap poly/diff 0

The threshold for reporting capacitance may be set in the .cadrc file with the following line.

capthreshold *value*

A negative value sets the threshold to infinity.

Mextra knows of the technologies, nMOS ("nmos"), MOSIS P well CMOS/Bulk (3.0 micron), also known as CBPM ("cmos-pw"), MOSIS Scalable CMOS/Bulk N-well, also known as SCN ("cmos-nw"), MOSIS Scalable CMOS/Bulk P-well, also known as SCP ("cmos-s"), and MOSIS Scalable CMOS/Bulk Generic, also known as SCG ("cmos-g"). The technology can also be set by use of the "-t <tech>" option; the string <tech> is the technology. Nmos is assumed by default. To set an alternate technology, include a line similar to the following in your .cadrc file with the appropriate string:

tech cmos-pw

Valtbs is a simple sed script that extracts all transistor records from the .tbs file except for 'p' transistors whose base is connected to Vdd, vdd, or VDD, and except for 'e' transistors whose base is connected to Gnd, gnd, or GND. Input files are named explicitly on the command line or via standard input. Output is on standard out.

FILES

~/.cadrc
basename.cif

basename.al
basename.log
basename.nodes
basename.sim
basename.ohm
basename.tbs

SEE ALSO

powest(1.vlsi), pspice(1.vlsi), spcpp(1.vlsi), sim2spice(1.vlsi), spice(1.vlsi), drc(1.vlsi), erc(1.vlsi), valtbs(1.vlsi), caesar(cad1), cadrc(cad5)

AUTHOR

Dan Fitzpatrick (UCB)

MODIFICATIONS

Wayne E. Winder (Northwest LIS, University of Washington)

BUGS

Accepts manhattan simple CIF only, use **cifplot –O** to convert complicated CIF. For unusually shaped transistors the UW/NW modified *mextra* should be used, otherwise values will be quite inaccurate. The modified *mextra* will either yield accurate values or a "reasonable" guess, depending on the complexity of the unusual transistor. The modified *mextra* will tell you when the output values are only best estimates. The length/width ratio for unusually shaped transistors may be inaccurate. This is true for snake transistors. Attributes for funny transistors are not recorded. Node attributes are ignored unless the –o switch is present.

NAME

sim2spice – convert from .sim format to spice format

SYNOPSIS

sim2spice [–d defs] file.sim

DESCRIPTION

Sim2spice reads a file in **.sim** format and creates a new file in spice format. The file contains just a list of transistors and capacitors, the user must add the transistor models and simulation information. The new file is appended with the tag **.spice**. One other file is created, which is a list of **.sim** node names and their corresponding spice node numbers. This file is tagged **.names**.

Defs is a file of definitions. A definition can be used to set up equivelences between **.sim** node names and spice node numbers. The form of this type of definition is:

set *sim_name spice_number* [*tech*]

The *tech* field is optional. In NMOS, a special node, 'BULK', is used to represent the substrate node. For CMOS, two special nodes, 'NMOS' and 'PMOS', represent the substrate nodes for the 'n' and 'p' transistors, repectively. For example, for NMOS the **.sim** node 'GND' corresponds to spice node 0, 'Vdd' corresponds to spice node 1, and 'BULK' corresponds to spice node 2. The *defs* file for this set up would look like this:

```
set GND 0 nmos
set Vdd 2 nmos
set BULK 3 nmos
```

A definition also allows you to set a correspondence between **.sim** transistor types and and spice transistor types. The form of this definition is:

def *sim_trans spice_trans* [*tech*]

Again, the *tech* field is optional. For NMOS these definitions would look as follows:

```
def e ENMOS nmos
def d DNMOS nmos
```

Definitions may also be placed in the '.cadrc' file, but the definitions in the *defs* file overrides those in the '.cadrc' file.

SEE ALSO

ext2sim(1), magic(1), spice(1), cadrc(5), ext(5), sim(5)

AUTHOR

Dan Fitzpatrick
CMOS fixes by Neil Soiffer

BUGS

The only pre-defined technologies are **nmos**, **cmos-pw**, and **cmos** (the same as **cmos-pw**). Only one definition file is allowed.

NAME

cif2ps – CIF to PostScript output

SYNOPSIS

cif2ps [-w *width*] [-h *height*] [-t *technology*] [-L *ciflayernames*] [-o *output.ps*] *input1.cif input2.cif ...*

DESCRIPTION

cif2ps takes a CIF file that has been produced by Magic or EGS graphics editor and creates a PostScript file that can be sent to the PostScript printer of choice. The code was written with the CIF layer names hard coded in. The technologies known are: nMOS ("nmos"), MOSIS P well CMOS/Bulk, also known as CBPM ("cmos-pw"), MOSIS Scalable CMOS/Bulk N-well, also known as SCN ("cmos-nw"), MOSIS Scalable CMOS/Bulk P-well, also known as SCP ("cmos-s"), and MOSIS Scalable CMOS/Bulk Generic, also known as SCG ("cmos-g").

Normally (without the -m option), upon invoking **cif2ps**, the program waits for user to enter the symbol number to be plotted. Entering invalid number will cause the program to list the valid symbol numbers. To terminate the input loop, enter a blank line.

Cif2ps options are :

–w specify width of plot (in pages, default is 1)

–h specify height of plot (in pages, default is 1)

Cif2ps will rescale a user-specific dimension if neccesary to avoid producing blank pages. The largest dimesion allowed is 5 pages by 5 pages (25 pages of output).

–t specify technology.

–m specify that plots are to be superimposed. This is useful for plotting node numbers generated from layout extractor onto the cif file. *cif2ps* will not wait for user input and will proceed and plot all symbols in one page.

–o specify the output file, otherwise cif2ps writes to standard output.

–L specify that the CIF layer names are defined in the file *ciflayernames*.

magic

AUTHOR

Arthur Simoneau wrote the version 'cifp'.
Marc Lesure modified 'cifp' to produce 'cif2ps'.
Boon-Khim Liew added -m, -o and -l options, support for other technologies and ability to draw polygons, wires.

FILES

ciflayernames, input.cif and output.ps

NOTES

PostScript is a trademark of Adobe Systems, Inc.