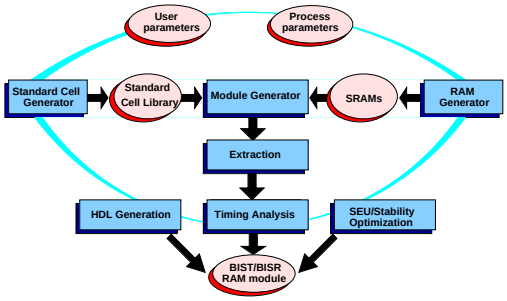
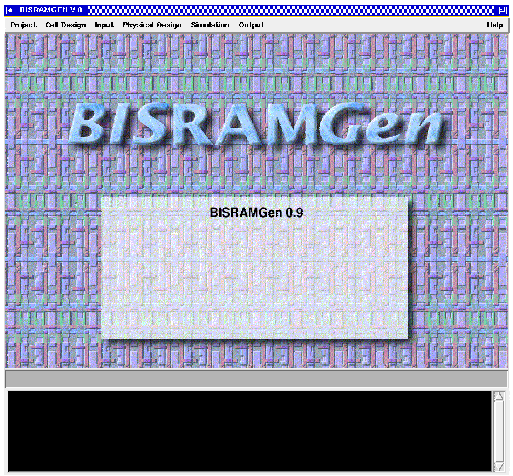
BISRAMGen

Advanced ULSI Memory Synthesis

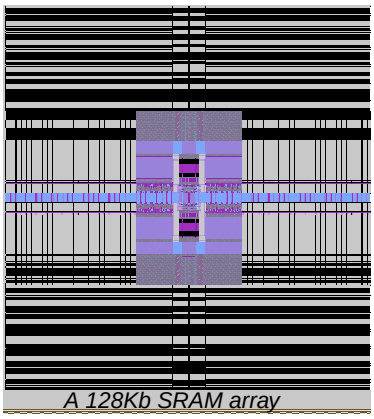
BIS*RAM*Gen is a Solaris/Linux based design rule independent SRAM generator that can be used for a wide range of deep-submicron CMOS processes developed by commercial vendors.

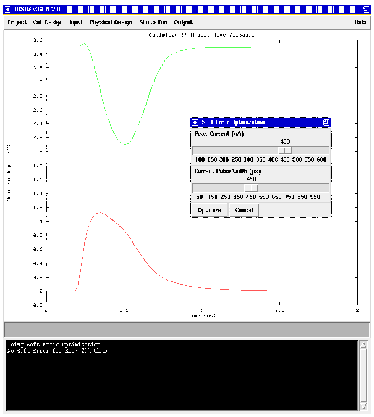
- *User-specified geometry parameters*
- *SEU and cell stability analysis and optimization*
- *Fast and accurate timing analysis and back-annotation to VHDL/Verilog description of the RAM*
- *Programmable memory test function - based on layout and technology related fault algorithms to guarantee high physical defect coverage*
- *Innovative circuit technique for self-repair improves yield considerably*
- *Capable of generating stand-alone modules -- RAM, BIST RAM, BIST + BISR RAM*
- *Independent of commercial standard cell libraries or placement and routing tools*





NanoSys


A 128Kb SRAM array



Layout and HDL generation

*BIS*RAM*Gen* takes in technology data (process parameters and design rules) and user specified geometry parameters. A key feature of the tool is its independence from commercial standard cell libraries. This provides the user the flexibility to specify the sizes of critical components in the system and thus optimize the performance of the layouts. *BIS*RAM*Gen* produces high quality layouts for RAMs with optional BIST and BISR (Built-In Self-Repair) circuitry which closely emulate full custom design style. A fast ATD circuit turns on word lines and sense amps only as needed thus saving power. A VHDL/Verilog netlist is generated that provides complete timing and functional specification of the RAM.

SEU and stability

The user can choose to analyze or optimize a RAM design for SEU hardness and stability. SEU hardness is reported in terms of the critical charge of the RAM cell design. The SEU optimizer computes a decoupling resistance value for user-specified alpha particle dosage. Memory stability is analyzed via phase plots for varying RAM array sizes and the optimizer sizes cell transistors to reflect a chosen stability value.

BIST and BISR

The BIST approach used by *BIS*RAM*Gen* achieves a high diagnostic fault coverage for word-oriented RAMs including single-cell faults, all possible 2-coupling faults between cells of the same word, and certain types of pattern-sensitive and parametric faults such as sleeping sickness (or data retention faults). The BISR optimizer provides user with optimal number of spares for a targeted reliability, yield, and area overhead. For embedded memory sizes of a few KB, the area overhead of BISR is less than 1%.

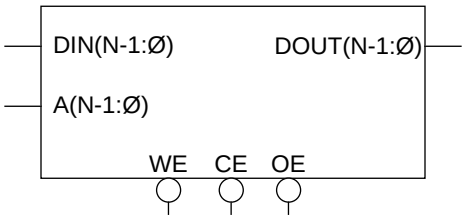
Timing analysis

*BIS*RAM*Gen* uses a new approach to timing analysis that effectively combines accuracy of analog simulation with speed of high-level timing analysis to provide fast and accurate back annotation of timing parameters to the generated Verilog/VHDL specification of the RAM.

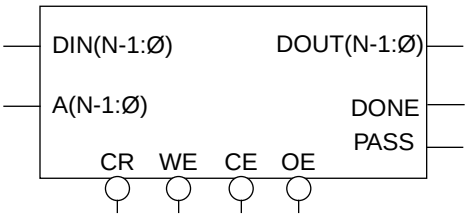
BISRAMGen

HIGH-SPEED LOW-POWER SRAM WITH BIST/R OPTION

SRAM



BIST/R SRAM



The high speed, low power SRAM uses a 6-T memory cell that can be designed for user specified stability and SEU hardness. A fast ATD circuit turns on word lines and sense amplifiers only as needed, thus minimizing power consumption. The two stage sensing scheme with a differential amplifier output is fast and robust. Aspect ratio and floorplan is user adjustable by means of the BPC and ARRAY parameters.

SIGNALS

INPUTS		OUTPUTS	
PORT LABEL	**FUNCTION**	**PORT LABEL**	**FUNCTION**
A	Address	DOUT	Output Data
DIN	Input Data	DONE	BIST/R done, active high
$\overline{CE}$	Chip Enable, active low	PASS	BIST/R pass, active high
$\overline{WE}$	Write Enable, active low		
$\overline{OE}$	Output Enable, active low		
$\overline{CR}$	BIST/R Start, active low		

PARAMETERS

Name	Definition	Values	Name	Definition	Values
BPW	bits per word	1 - 144	BIST	choose BIST	0, 1
WORDS	number of words	8 - 16384	BISR	choose BISR	0, 1
SPARES	number of spares	0 - 16	ARRAY	floorplan	0, 1, 2, 3
BPC	bits per column	2, 4, 8, 16, 32	FREQUENCY	frequency in MHz	1-1000

TRUTH TABLE

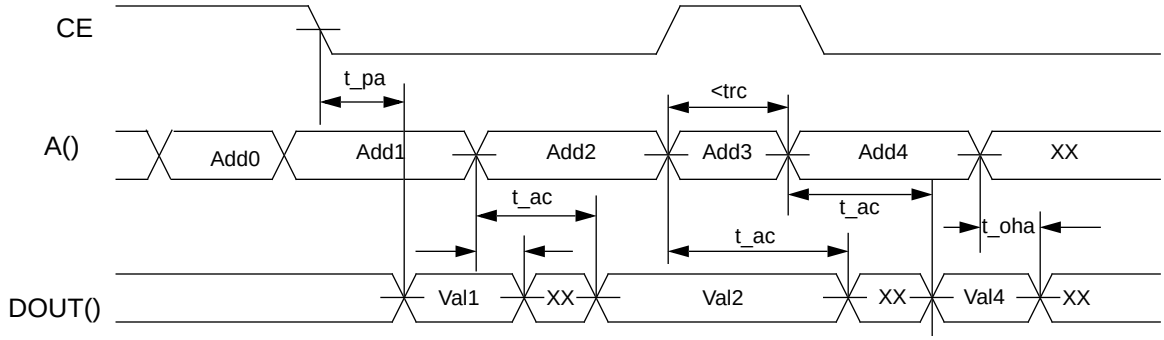
INPUTS					OUTPUTS	COMMENT
A	**$\overline{CE}$**	**$\overline{OE}$**	**DIN**	**$\overline{WE}$**	**DOUT**	
Ø/1	Ø	Ø	X	1	Data	Read
X	Ø	1	X	1	Z	Output Disabled, RAM Active
X	1	1	X	X	Z	Output Disabled, RAM Disabled
X	1	Ø	X	X	Data	Outputs Stable, RAM disabled
Ø/1	Ø	Ø	Ø/1	Ø	Data	Write with Write-Through
Ø/1	Ø	1	Ø/1	Ø	Z	Write

VERILOG DESCRIPTION

```
ram       inst_name (A, DIN, CE, WE, OE, DOUT);
bisram    inst_name (A, DIN, CE, WE, OE, DOUT, CR, DONE, PASS);
```

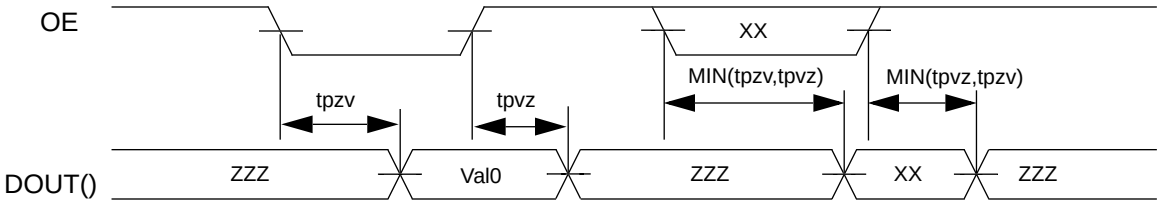
SWITCHING TIME WAVEFORMS

RAM Read Cycle ($\overline{CE}$ = '0', OE = '0', $\overline{WE}$ = '1')



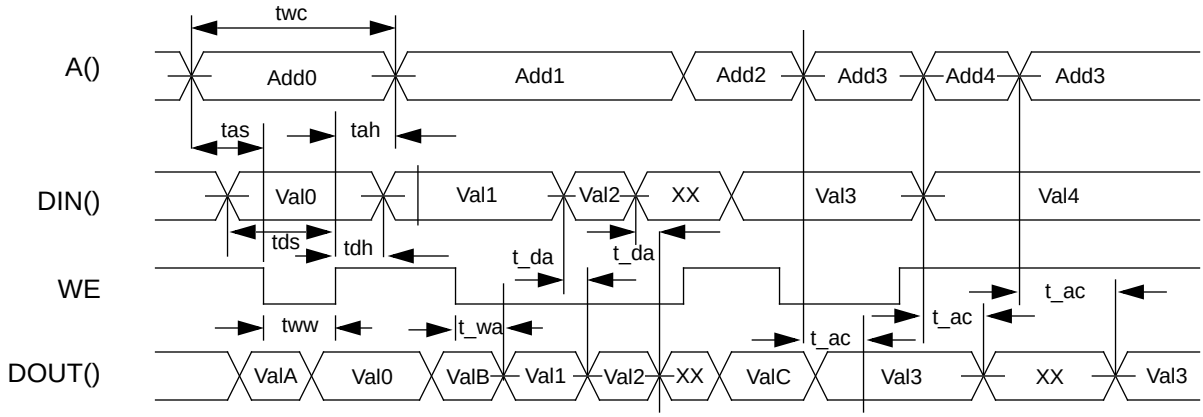
t_ac: Address access time	acv
t_pa: $\overline{CE}$ access time (maximum of Rise/Fall)	pav
t_rc: Read cycle time	rcv

RAM Read Cycle ($\overline{CE}$ = '0', $\overline{WE}$ = '1')



tpvz: Delay to propagate valid DOUT() to high impedance.	pvv
tpzv: Delay to propagate high impedance to valid DOUT().	pzvl

RAM Write Cycle ($\overline{CE}$ = '0', OE = '0')



t_da: Output delay time from Data (t_da < t_ac) (MAX of Rise/Fall)	dav
t_wa: Output delay time from $\overline{WE}$ (MAX of Rise/Fall)	wav
tas: Address setup time	asv
tah: Address hold time	ahv
tds: Data setup time	dsv
tdh: Data hold time	dhv
tww: Write pulse width low	wwv
twc: Write cycle time	wcv

ALLEGRO: Contaminant to Fault Analysis Program

ALLEGRO generates statistical information regarding fault probabilities in a VLSI cell layout based on the layout and contaminant density distribution information provided by the user.

In a real VLSI chip, the contaminants are usually 3-D particles which cause bridge or break types of faults. If the particle is insulating in nature, it may cause a break fault by preventing normal signal flow through conducting paths. On the other hand, if the particle is conductive, it may short two or more signals in a circuit. The existing inductive fault analysis approaches fail to take into account the 3-D contaminant effect and instead deal with 2-D spot defects. This can lead to unrealistic probabilities for certain faults. Our approach — the 3-D approach — is the most realistic one.

The value of ALLEGRO to the layout designer and the VLSI circuit designer will be enormous. ALLEGRO will be useful for:

Test pattern generation ALLEGRO generates fault statistics which is helpful in finding out the relative importance of different types of probable functional failures that a circuit can suffer from. This will definitely help the circuit designer to use particular fault models (eg. simple stuck-at, or bridge/break, or delay) based on which he/she can generate test patterns for the circuit.

Layout quality comparison Usually, a circuit designer will have a choice of several different layouts for the same circuit which may otherwise be equivalent. ALLEGRO can then be used to determine the relative vulnerability of the layouts to contaminants, thereby enabling the designer to choose the best layout.

Layout quality improvement One of the outputs of ALLEGRO is a fault density map of a given layout. The layout is superimposed with a fault density bitmap which clearly shows the layout designer which portions of the layout are the most vulnerable so far as contaminants are concerned. This information can then be used by the layout designer to make local modifications in the layout and improve the overall quality of the layout.

Brief description of ALLEGRO

The user has to supply the layout as well as the 3-D geometry of each layer in it. Additionally, the defect density distribution can be specified by the user. Based on these information, the software injects 3-D contaminants into the layout in a Monte Carlo fashion and analyzes the effect of each contaminant in terms of circuit functionality to determine the corresponding fault-type, if it causes any. At the end of the Monte Carlo simulation, the fault statistics along with a fault density map is produced (Fig. 1).

One of the main advantages of this software is its speed. For instance, it can analyze an SRAM 4×4 array within a few minutes on a Sun Ultra 1 workstation.

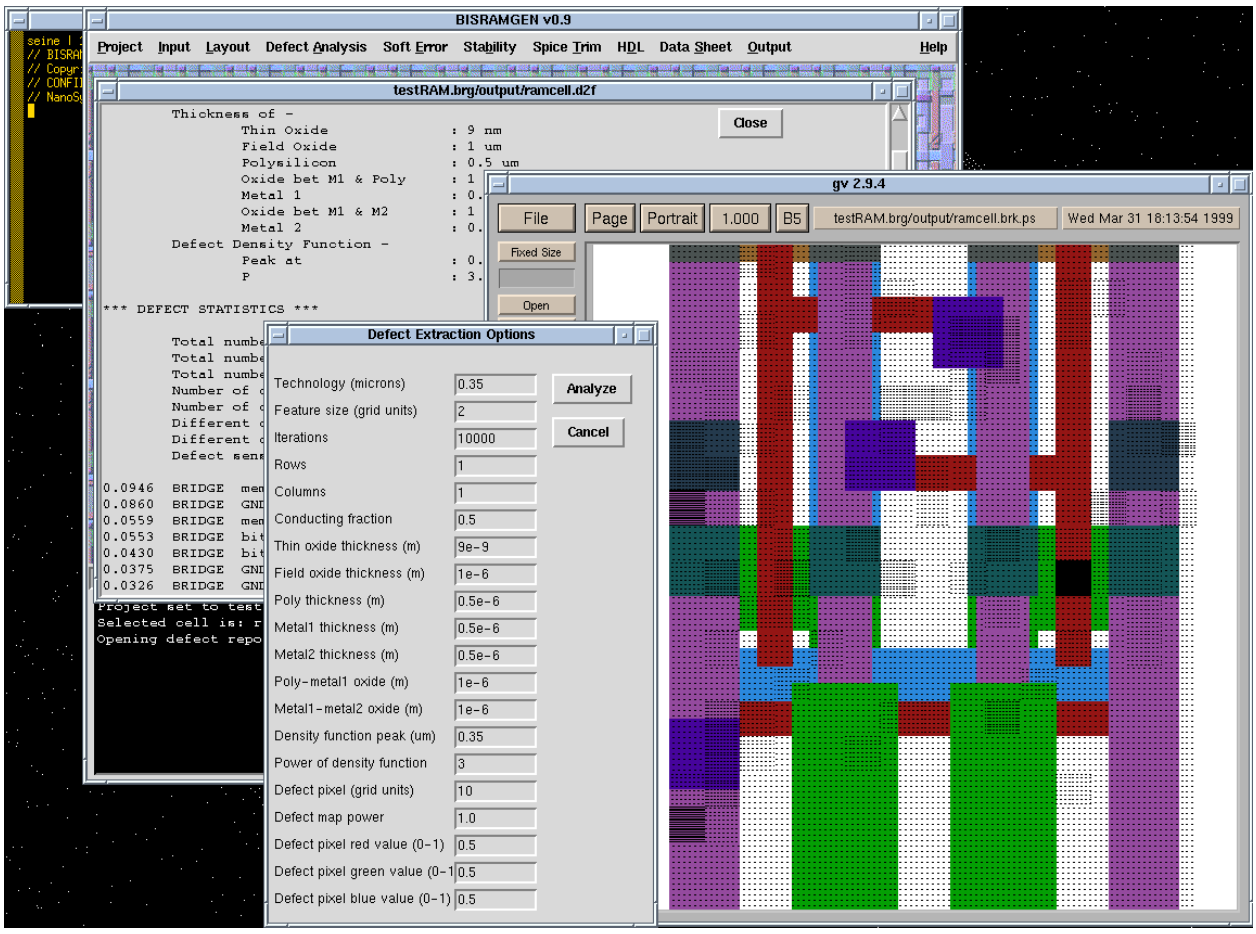


Figure 1: ALLEGRO in action — fault statistics window in the background, user entry and fault density bitmap windows in the foreground

Work required before commercialization

Several different aspects of ALLEGRO need to be worked on before achieving commercial grade.

1. The contaminant injection will have to be done with respect to fabrication process steps.

2. Layout geometric features should be made more realistic. In real chips, the feature shapes tend to be rounded in the edges and corners, whereas our program assumes sharp right-angled features. In deep-sub-micron circuits, the error percentage due to such approximation may be significant.

3. The code will have to be optimized for speed when process steps and contaminant injection are tied up.

4. Contaminants are currently cubic in shape. They have to be made more realistic. Randomization of shape has to be introduced. This will result in introducing complications to the fault analysis phase of the simulation, but this can be completed within three to four months.

SERUM: Soft-Error Rate Evaluation Program

SERUM calculates the soft-error rate (SER) in VLSI circuits due to on-chip sources of radiation.

The value of SERUM to the layout designer and the VLSI circuit designer will be enormous. In particular, SERUM will be useful because:

It calculates SER due to interconnects and solder bumps Off-chip radiation sources like cosmic rays can be dealt with by advanced packaging technology, but on-chip radiation sources like metal wiring and solder bumps can cause significant single-event upset (SEU) problems in deep-sub-micron (DSM) VLSI circuits. Thus SERUM can be a powerful tool in the arsenal of the DSM circuit designer.

It helps in assessing circuit reliability SERUM performs a complete topological analysis of the chip while calculating the SER. This means that the effect of the chip layout on its reliability can be ascertained by using this tool.

It helps to compare two similar layouts The SER output provided by SERUM can be used to judge the quality of a layout as compared to that of another layout so far as SER is concerned. This will enable the DSM designer to make educated choice of cell layouts while designing reliable circuits.

It helps in improving DSM VLSI layout One of the outputs of SERUM is a picture showing the given cell's layout which is superimposed with a bitmap which clearly shows which portions of the layout are responsible for causing most of the SEU problems. This information can then be used by the layout designer to make local modifications, thereby improving overall layout quality.

Brief description of SERUM

The GUI for SERUM is shown by means of the two snapshots presented in Fig. 1 and Fig. 2. The user provides the layout and its 3-D geometry. The alpha-particle emission rates of the different radiation sources (like interconnects, solder) and process parameters are also provided by the user. Via the GUI (Fig. 1), the user also identifies the vulnerable locations within the layout (eg. drains of certain transistors) where an

alpha-particle hit may cause SEU type failure. There is a critical charge (Q_{crit}) measurement routine in the program, but right now it works only for SRAMs. So the user is responsible for providing an estimate of the Q_{crit} for the circuit. This info is usually available to the user if he/she is familiar with the circuit's operation. The program uses the 3-D geometry information of the layout to do a complete topography analysis and determines the probability of an alpha-particle hitting the user-defined vulnerable spots. Using this information and the Q_{crit} data, the SER is then calculated. As stated before, the program also outputs a problem region bitmap (Fig. 2) which can be very useful for layout improvement.

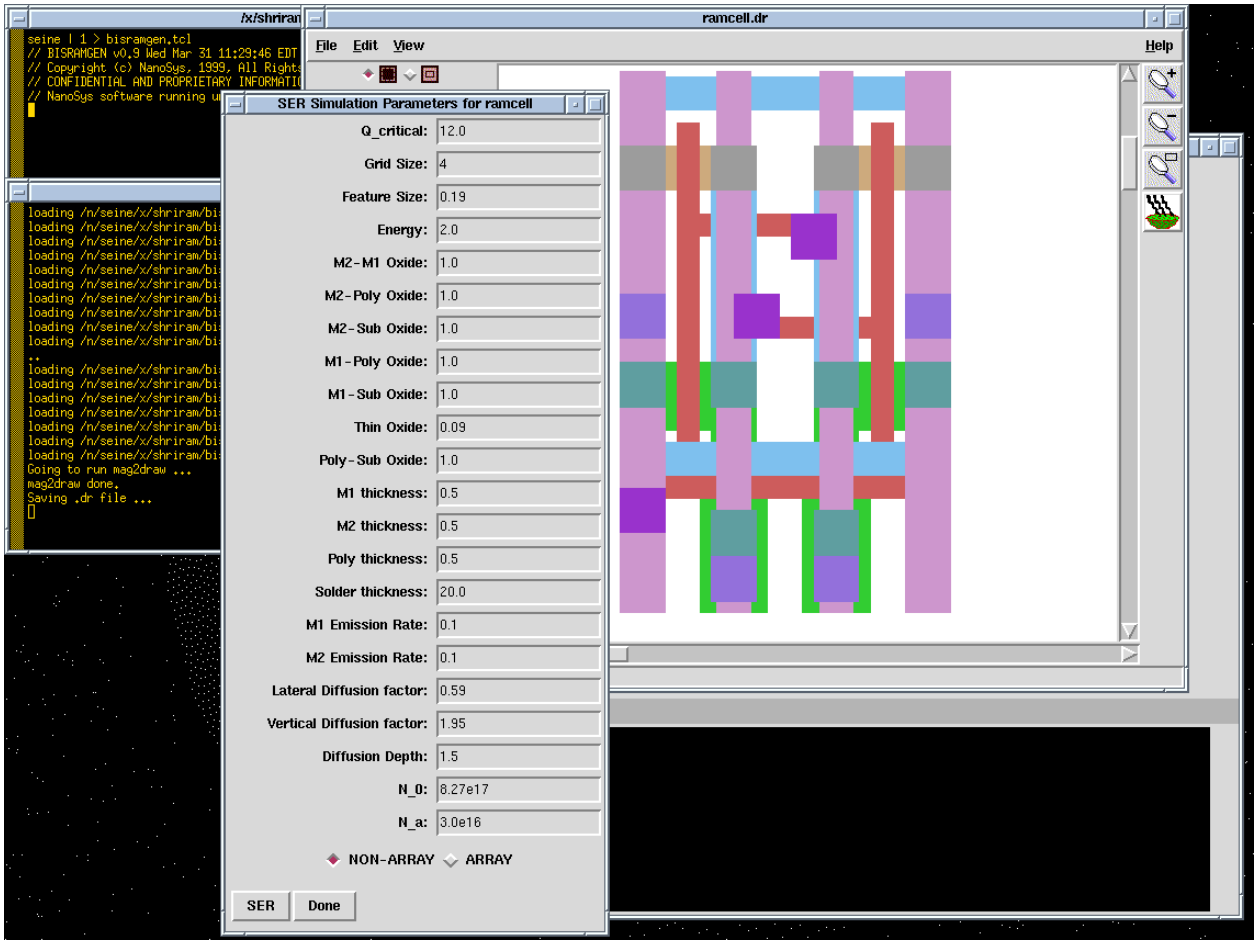


Figure 1: The GUI for entering the vulnerable spot information

Work required before commercialization

Several different aspects of SERUM need to be worked on before achieving commercial grade.

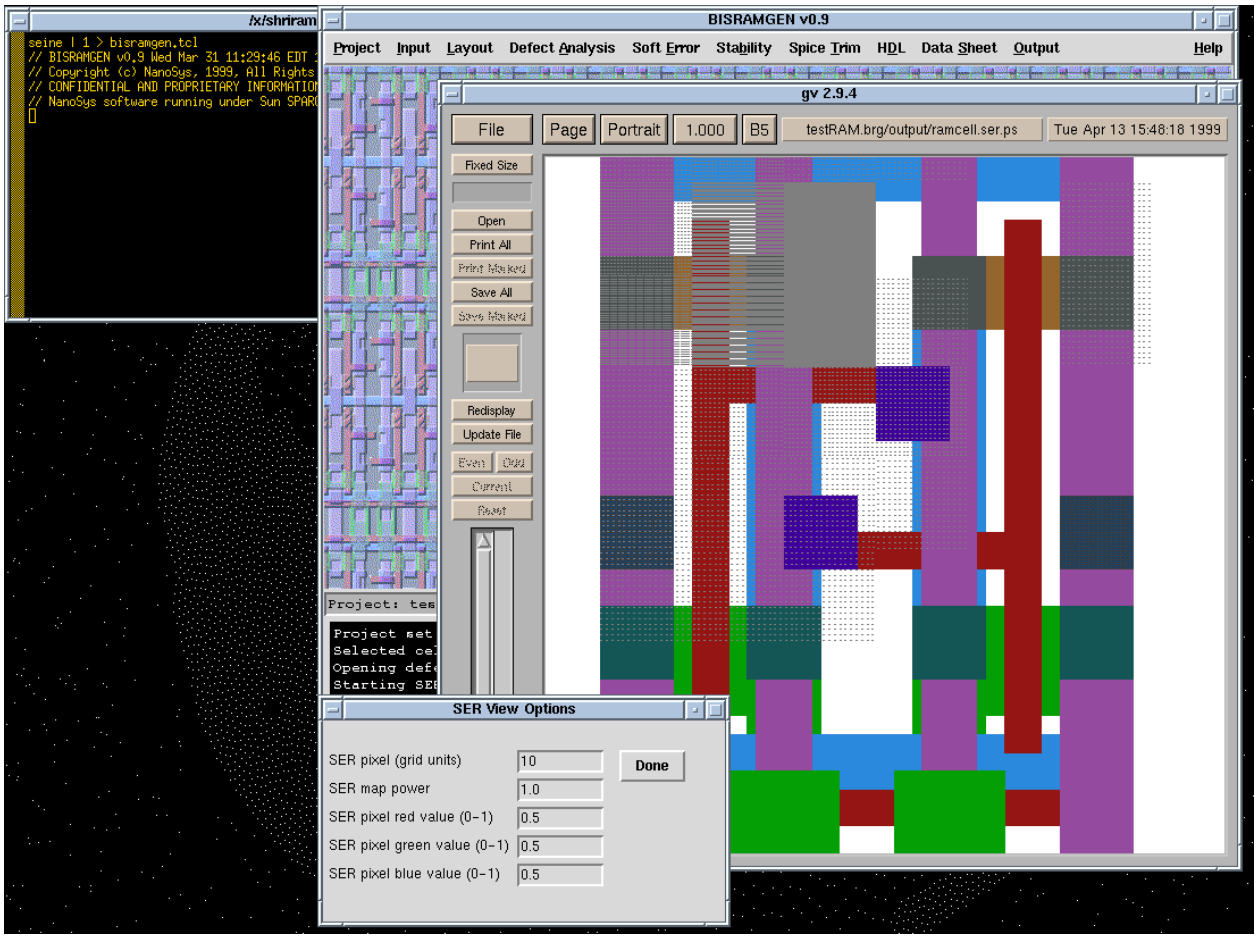


Figure 2: SER bitmap output

1. The Q_{crit} calculation for general (non-SRAM) circuits will have to be added to the program. The user will have to define the signal settings for Q_{crit} evaluation and the program will do layout based parasitic extraction and the necessary SPICE simulations.

2. Since the 3-D geometry of the circuit is extremely important for calculating particle hit probabilities, the topography analysis has to be extremely accurate. The current program's 3-D topography generation is somewhat idealized. For instance, edges and corners of materials are assumed to be right-angled. However, in DSM circuits, the features are more rounded than what the program assumes. So, some amount of effort will have to be spent on this.

3. In the current version of SERUM, the user-defined vulnerable spot can only have a rectangular cross-section on the chip surface. This will have to be changed to accommodate more complex shapes since in some cases such approximations may yield inaccurate SER projections.