

VICTORY と SmartSpice SEE による SRAM セルの物理ベース 3次元 SEU シミュレーション

はじめに

VICTORYは、最先端の半導体テクノロジーを1次元、2次元、および3次元でシミュレーションするツールから成るシミュレーション・フレームワークです。VICTORYでは、複雑な3次元形状を高速かつ高精度にシミュレーションするため、四面体メッシュ生成エンジンを採用しています。メッシュ細分化条件（ビルトイン/ユーザ定義）を利用して、シミュレーション中にメッシュをカスタマイズできます。

このメッシュ細分化条件が効果を発揮する例が、SEU (Single Event Upset: シングル・イベント・アップセット) のシミュレーションです。3次元構造においてSEU現象のシミュレーションを行う場合、SEUトラック周辺の物理量勾配が大きいため、計算が非常に複雑になります。そのため、3次元構造のSEUシミュレーションを高精度かつ確実にを行うには、SEUトラック中心付近のメッシュを極めて密にする一方、SEUトラックから遠い部分のメッシュは粗くして効率を良くすることが不可欠です。

本稿では、SRAMセルにおけるSEU現象を正確にシミュレーションする方法を説明します。

デバイス構造の生成

まず、DevEdit3Dを利用して、2個のnMOSトランジスタと2個のpMOSトランジスタから成る3次元構造を作成します。

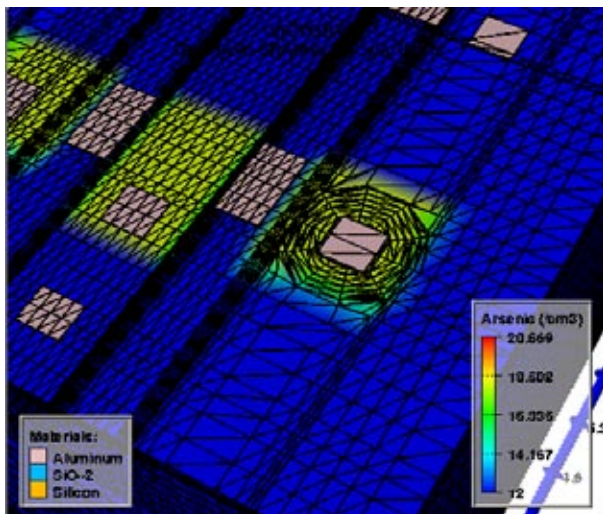


図 1. SRAM 構造のヒ素濃度および表面メッシュ

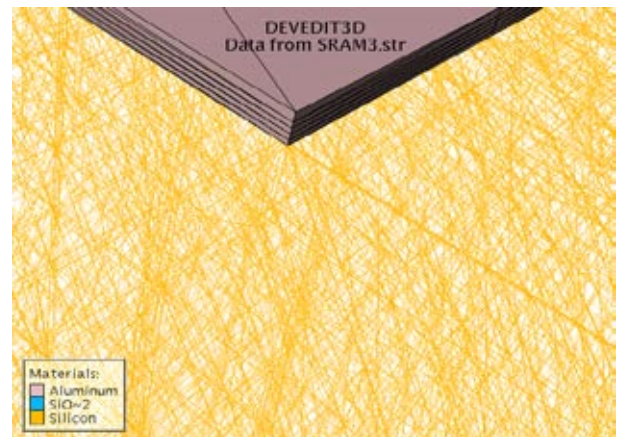


図 2. SRAM 構造のコンタクト周辺の四面体メッシュ

デバイス・シミュレーションを実行するために必要な、デバイスの形状、材料、濃度コンタ、3次元四面体メッシュなどを DevEdit3D で設定します。

作成された3次元構造の不純物分布および表面メッシュを図1に示します。SEU発生予定箇所に、円筒状のリメッシュが施されていることに注目してください。

図1の3次元構造のコンタクト周辺を拡大したのが図2で、ここでも四面体メッシュを確認できます。

トランジスタ4個から成るSRAM回路の回路図を図3に示します。この回路の全トランジスタは数値デバイス・シミュレーションで使用されるため、トランジスタ間のすべてのカップリング効果が考慮されます。どのトランジスタも、SPICEデバイスとしてはシミュレーションされません。

シミュレーションの設定

VICTORYへのシミュレーション入力は、DeckBuildを使用し、以下のような内容を定義します(図4および図5参照)。

- ・ Deveditで作成したデバイス構造のロード
- ・ 円筒状のリメッシュ
- ・ モデルの指定
- ・ DC解析および過渡解析の指定

SEU発生方向は、粒子の入口点および出口点にそれぞれ対応する1対の(x,y,z)座標で指定されます(図6参照)。

SEUトラックを円筒形と仮定します。最大となる電荷密度は、電荷発生パルスの幅(時間単位)に沿って指定できます。VICTORYにはデフォルトのSEU関数が用意されていますが、ユーザ定義のSEU関数を自由に作成することもできます(図7参照)。

図3のSRAM回路において、規定のDCバイアスで、ノード3を0.0V、ノード2を3.3Vにそれぞれ設定します。このDCバイアス電圧を、過渡解析の初期条件として使用します。過渡解析は、初期時間刻み幅を10fsとして、10ms間実行します。SEUは、4ps時に最大電荷密度を持つものとし、その幅は2psとします。

シミュレーション結果

VICTORYのシミュレーションでは、ノード電圧とノード電流が、過渡時間に対する関数として出力されます。さらに、デバイス内部の振る舞い(ポテンシャル密度、電子密度など)も、過渡時間の関数として解析できます。

図8は、ノード2/ノード3の電圧を過渡時間に対してプロットしたものです。この図から、SEU粒子の入射が、ノード2とノード3の電圧に与える影響が分かります。

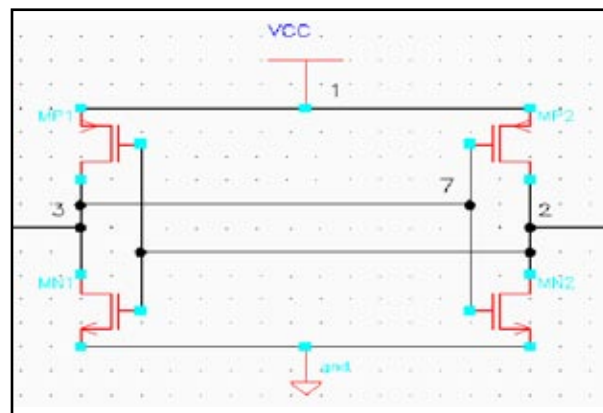


図3. SRAMセルの回路図

SEUに沿った円筒状メッシュを生成しない場合、電圧はフラットなままであり、シミュレーション結果が不正確になります(図9の青のカーブを参照)。図9において、緑のカーブは2個の円筒状メッシュで計算した場合を示し、赤のカーブは10個の円筒状メッシュで計算した場合を示します。

2個または10個の円筒状メッシュを使用する場合のメッシュ離散化係数を、SEUトラックに沿って2次元でプロットしたのが図10です。メッシュ離散化係数は、メッシュを細分化する必要がある領域を見つけるのに役立ちます。赤色は、(相対的線形スケールにおいて)最も非線形なエレメントを示します。つまり、赤色の領域は、メッシュの細分化によって、解が最も大きく変化する領域であると推定されます。

```
Deckbuild V3.23.17.R - (NONE) (edited), dir: /home/eric/workdir
File View Edit Find Main Control Commands Tools
go victory
mesh inf=SRAM.str hybridx
contact NAME=gate
contact NAME=source
contact NAME=bulk
contact NAME=drain WORK=4.35 COMMON=bulk
contact NAME=substrate WORK=4.35 COMMON=source
contact NAME=fgate WORK=4.25 COMMON=source
contact NAME=collector
contact NAME=base WORK=4.25 COMMON=source
contact NAME=anode COMMON=bulk
contact NAME=vss COMMON=gate
contact NAME=vgg COMMON=collector
contact NAME=cgate COMMON=bulk
models consrh auger bgn cvt klassen
method bicgst.rcm
method linear.fill_level=2
method linear.scaling_parameter=2
method NONLINEAR.MAX_ITERATIONS=30
regrid x_min=0.775 y_min=0 z_min=3.22 \
x_max=0.775 y_max=5 z_max=3.22 \
radius=0.05 \
lmax=20 rmax=10 num.cylinders=10 constraint
next line stop cont run quit Line: 2
paste init pause clear restart kill Stop: None
Thu Feb 2 10:52:32 2006
Executing on host: triesves
Starting: DEVICE module.
VICTORY>
VICTORY started
```

図4. DevEdit構造のロードと、円筒状リメッシュの定義

```
Deckbuild V3.23.17.R - (NONE) (edited), dir: /home/eric/workdir
File View Edit Find Main Control Commands Tools
solve vgate=0.1 vstep=0.1 vstop=3.3 name=gate
solve vsource=0.1 vstep=0.1 vstop=3.3 name=source
# contact definition
contact name=source current
contact name=bulk current
contact name=drain current COMMON=bulk WORK=4.35
contact name=substrate current COMMON=source WORK=4.35
contact name=fgate current COMMON=source WORK=4.25
contact name=base current COMMON=bulk WORK=4.25
contact name=anode current COMMON=source
contact name=vss COMMON=gate
contact name=vgg COMMON=collector
contact name=cgate current COMMON=bulk
models consrh auger bgn cvt klassen
method bicgst.rcm
method linear.fill_level=2
method linear.scaling_parameter=2
solve isource=0
solve ibulk=0
beam f.radiate=ion_c_inter_v LET_3.5.lib
log outf=SRAM.log
solve tfinal=1e-5 tstep=1.e-14
next line stop cont run quit Line: 2
paste init pause clear restart kill Stop: None
Thu Feb 2 10:52:32 2006
Executing on host: triesves
Starting: DEVICE module.
VICTORY>
VICTORY started
```

図5. モデル、DC解析、過渡解析の指定

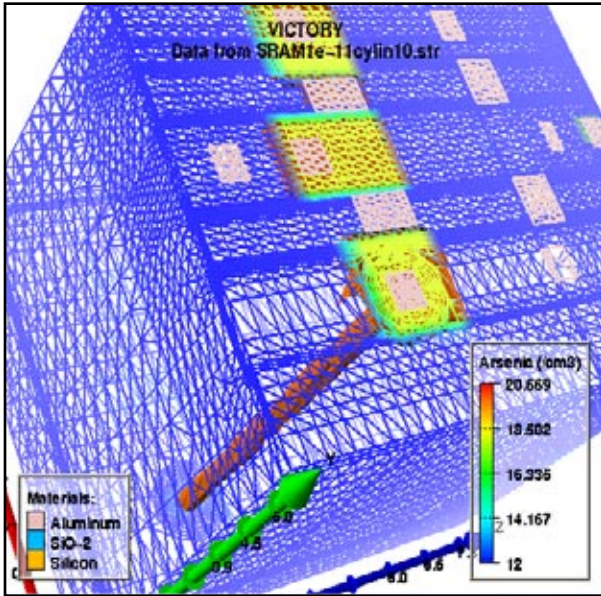


図 6. 3次元SRAM構造でSEU位置を表示

以上のことから、正確なシミュレーション結果を得るには、2個の円筒状メッシュによる細分化で十分と考えられます。

次に、入射粒子の強度を変えたときの、過渡時間に対するノード2/ノード3の電圧を図11に示します。これによって、SRAMセルの振る舞いが、入射粒子の強度の関数として解析できます。

```

Text Editor V3.6.2 FCS - NewDocument (edited), dir: /home/eri
File View Edit Find
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
#include <malloc.h>
#include <string.h>

/*
 * Generation rate as a function of position (3D)
 * Statement: BEAM
 * Parameter: F,RADIATE
 * Arguments:
 *   = location x (microns)
 *   = y location y (microns)
 *   = z location z (microns)
 *   = t time (seconds)
 * *rat generation rate per cc per sec.
 */
int radiate(double x, double y, double z, double t, double *rat)
{
    double LET; /* en pC/um*/
    double v0z; /* en um*/
    double x0; /* en um*/
    double z0; /* en um*/
    double y0; /* en um faire attention au sens de propagation du faisceau z>0
    ou z<0? */
    double l0; /* en s*/
    double Tc; /* en s*/
    double Range; /* en um*/

    LET=0.035;
    v0=0.05;
    x0=0.725;
    z0=0.22;
    y0=0;
    T0=1;
    Tc=2;
    Range=40;

    *rat =
    (LET*t+12*t+4/1.6e-19)/
    (5.56827997*(x0*t+4)*(y0*t+4))*
    exp(-(
    (x-x0)*(x-x0)/
    
```

図 7. C-インタプリタで作成したユーザ定義のSEU関数

低強度（青のカーブ）では、SRAMセルは変化しません。中強度（緑のカーブ）では、電圧が逆転します。高強度（赤のカーブ）では、SRAMセルが不安定になります。

VICTORYの3次元電子電流密度コンタを利用すると、SRAMセルに反転現象が起こるときの、nMOSFET内の電子電流密度の段階的变化を追跡することができます。

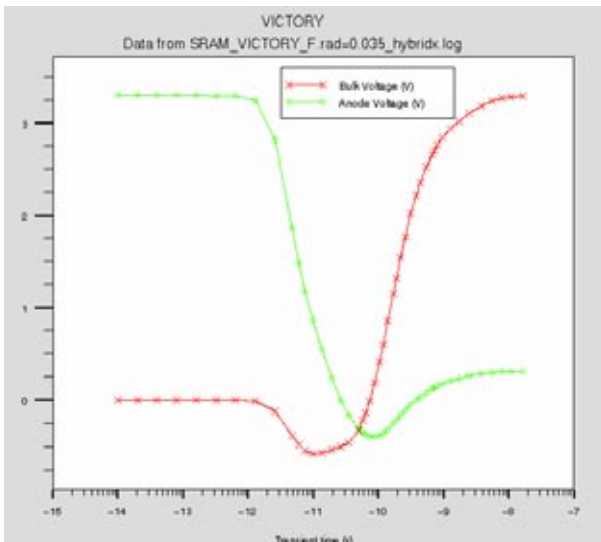


図 8.

ノード2/ノード3 (図3参照) の電圧vs.過渡時間。Anode Voltage がノード2、Bulk Voltageがノード3にそれぞれ対応。

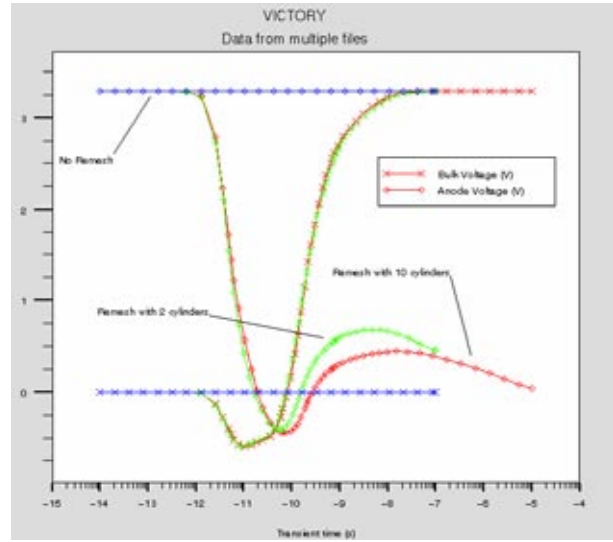


図 9.

ノード2/ノード3 (図3参照) の電圧vs.過渡時間およびSEU周辺のメッシュ密度。Anode Voltage がノード2、Bulk Voltageがノード3にそれぞれ対応。

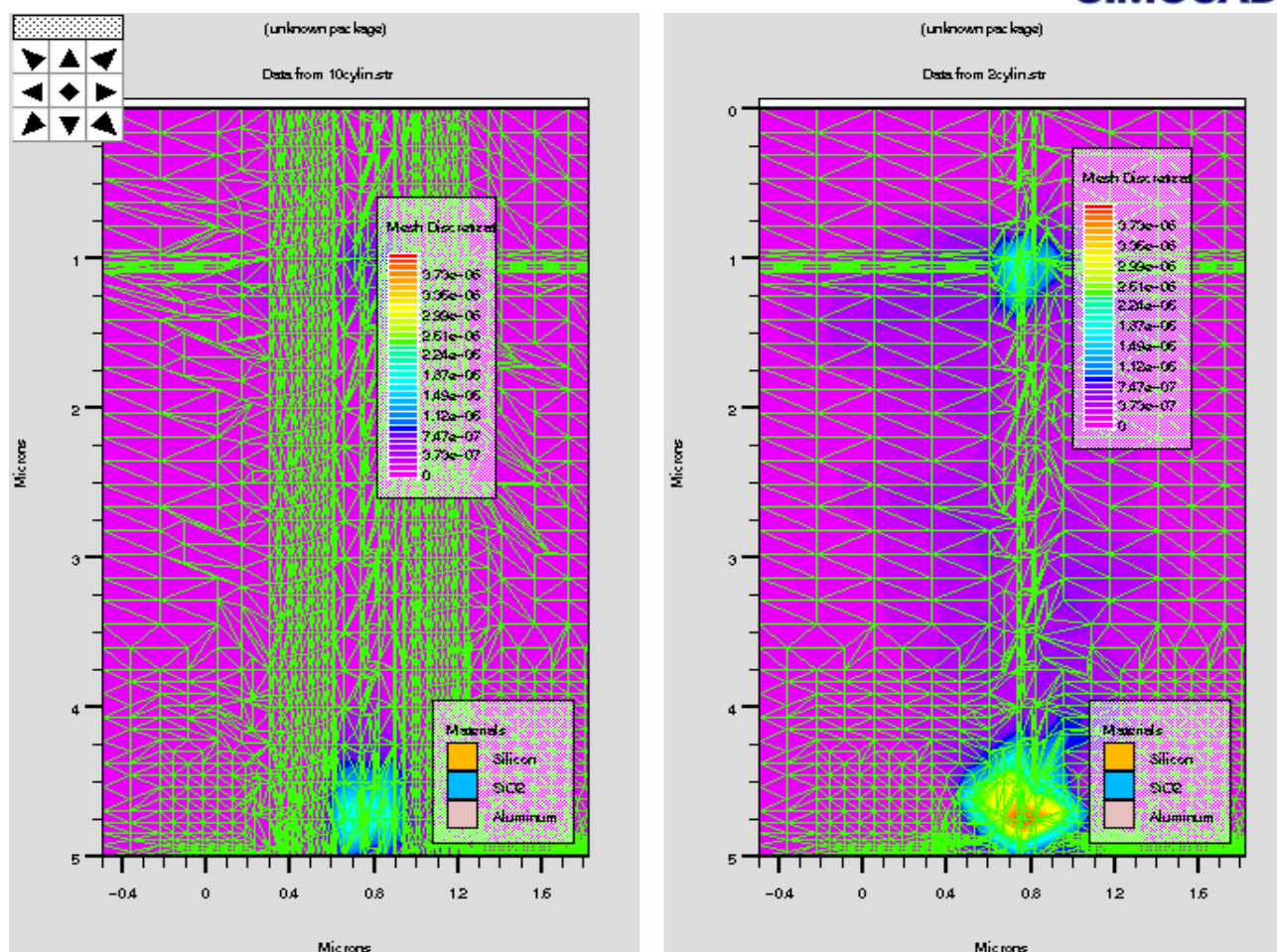


図 10. 10個 (左図) / 2個 (右図) の円筒状メッシュを使用した場合の、ノード3におけるメッシュ離散化係数

初期状態では、nMOSFETのソース・ドレイン領域間に流れる電子電流の密度は低くなっています (トランジスタはオフ)。粒子がnMOSFETに入射すると、その飛跡に沿って電子・正孔対が発生してデバイス全体の電子分布が変化し、nMOSFETの電子電流密度が増加します (トランジスタがオン) (図 12 を参照)。粒子がデバイス構造から出て行くと、外部的な電荷供給源がなくなるので、元の電子電流密度が復元されます (トランジスタはオフに戻る)。

回路シミュレーション

以上に述べたように、TCADシミュレーションでは、トランジスタ単体または少数トランジスタで構成される回路の振る舞いを解析できます。ポテンシャル、電子密度、電流密度などの数値を求めることにより、SEUに影響を受けるデバイスの振る舞いが詳細に解析できます。

しかし、より大規模な回路でSEU効果を調べたい場合は、TCADシミュレーションは利用できません。

SmartSpice SEEモジュールが開発されたのはそのためです。SmartSpice SEEモジュールは、MOS (バルクおよびSOI) や、バイポーラ・デバイスにおけるSEE (Single Event Effect: シングル・イベント効果) を正確に解析したい、というニーズに応じて開発されました。

粒子が入射すると、電子・正孔対が発生します。SmartSpice SEEでは、入射粒子による影響が予想されるノードで発生した電荷をモデル化するため、回路に電流発生器を挿入します。ASIC (特定用途向けIC) については、影響されやすいノードを特定することができます。発生する電流の波形は、SmartSpiceで利用可能な2つの指数関数電源で高精度に近似できます。SmartSpiceでは、その他の波形も利用できます。

ラッチ回路の電圧を過渡時間に対してプロットしたのが図 13 です。入射粒子が電圧に与える影響がわかります。

図 13 において、上図はラッチの安定状態を示します。中図は、 $t=10\text{ns}$ におけるSEU入射の影響を示します。寄生効果は発生しているものの、ラッチの状態には影響を与えていないことがわかります。さらにエネルギー(LEF)を増加させていったのが下図で、ラッチの状態が変化し、回路にエラーが生じます。

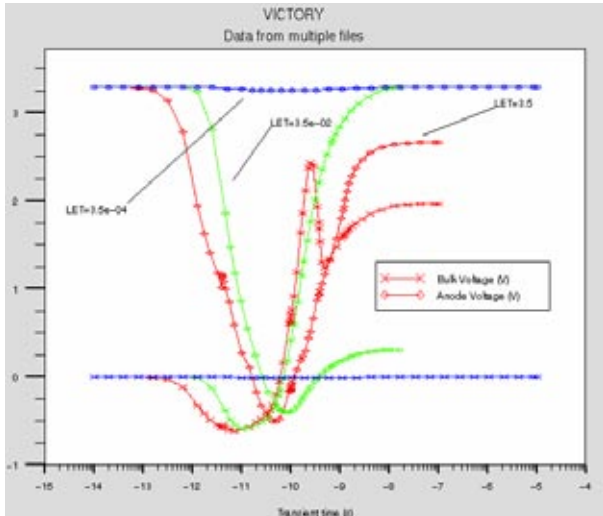


図 11.

入射粒子の強度を変化させたときの、ノード2/ノード3の電圧(図3参照) vs. 過渡時間。Anode Voltageがノード2、Bulk Voltageがノード3にそれぞれ対応。

SmartSpice SEEは、回路内の各ノードにおいて同時にまたは異なる時間に発生した複数のSEE効果をシミュレーションできます。アップセットの検出はユーザによってカスタマイズ可能で、絶対閾値または相対閾値を、任意のノードに対して一度に指定できます。

状態変化を引き起こすのに必要な、臨界電荷(QCRIT)は、自動で計算されます(詳細は、SmartSpice SEEの資料を参照)。

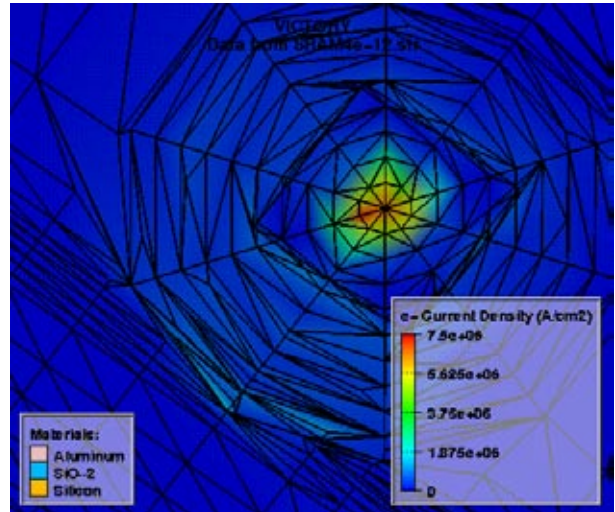


図 12.

nMOSトランジスタのドレイン領域(ノード3)の $t=4e-12s$ における電流密度

まとめ

VICTORYは、優れたメッシュ生成機能や、正確な物理モデルを使用して、SEUによる電荷生成を表現するI-V特性を出力できます。VICTORYで得られた電流特性をSmartSpiceの電流源とすることで、さまざまな種類の回路におけるSEU現象を、SmartSpice SEEの機能を使用して予測できるようになります。

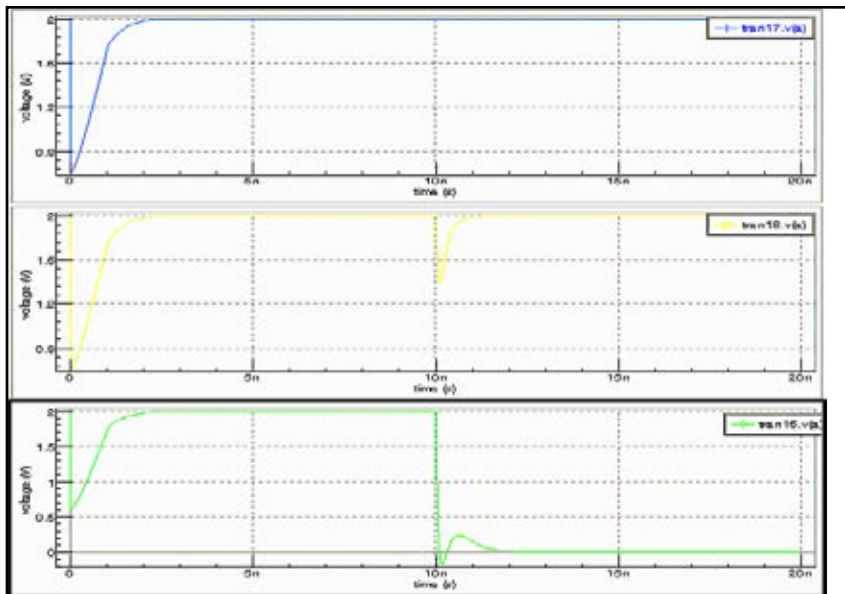


図 13. 入射SEU粒子の影響を受けたラッチ回路の振る舞い

