HICUM/L2 version 2.4.0

Release Notes

27 March 2017

michael.schroter@ieee.org

andreas.pawlak@tu-dresden.de

Content

- New model version scheme
- Strong avalanche
- Conditional statement for weak avalanche breakdown
- Increased parameter ranges
- Gmin for operating point values
- Renamed operating point value names of resistances
- Calculation of Vciei
- Summary model parameters

New model version scheme

- CMC model versioning policy
  <version #>.<subversion #>.<revision#>
- <subversion #> and <revision#> only 1 digit
- Version number increase
  - when new model formulations which are not backward compatible (with the exception of bug fix) are introduced
  - subversion number is greater than 9
- Subversion number increases
  - when new model formulation which is backward compatible is introduced
  - reset to 0 when the model version number is changed
  - reset to 0 with subversion number greater than 9 (increment of the version number and reset of revision number to zero)
- Revision number increase
  - when bug fixes are applied, including non-backward compatibility
  - increase for different implementations of the same set of model equations
  - reset to 0 when the model subversion or model version are changed

New model equations changes

Strong avalanche

- The new formulation for strong collector avalanche breakdown is given by

$$I_{AVL} = I_T \frac{f_{AVL}(V_{DCi} - V_{B'C'}) \exp\left(-\frac{q_{AVL}}{C_{jCi}(V_{DCi} - V_{B'C'})}\right)}{1 - f_{AVL}(V_{DCi} - V_{B'C'}) \exp\left(-\frac{q_{AVL}}{C_{jCi}(V_{DCi} - V_{B'C'})}\right)}$$

=> based on weak avalanche formulation, rewrite and add more flexibility

$$I_{AVL}^{weak} = I_T g$$

$$I_{AVL} = I_T \frac{g}{1 - k_{AVL} g}$$

- Parameter kAVL can be used to turn the strong avalanche off (kAVL = 0) or to fine-tune the model to measured data (kAVL > 0).
- For capturing impact of simplifying assumptions and allowing high accurcay, upper limit of kAVL is set to 3: $0 \leq \mathrm{kAVL} \leq 3$

Strong avalanche

- Parameter kAVL is modeled temperature dependent by

$$k_{AVL}(T) = k_{AVL}(T_0)\exp(\alpha_{KAV}\Delta T)$$

with the additional model parameter α_{KAV}.

- empirical equation, resulting from model verification on different technologies

- The denominator $1 - k_{AVL}g$ is limited to values greater zero by the following smoothing function

$$h_l = \frac{(1 - k_{AVL}g) + \sqrt{(1 - k_{AVL}g)^2 + a_{AVL}}}{2}$$

with the smoothing model parameter a_{AVL}, fixed to 0.01.

Parameter extraction

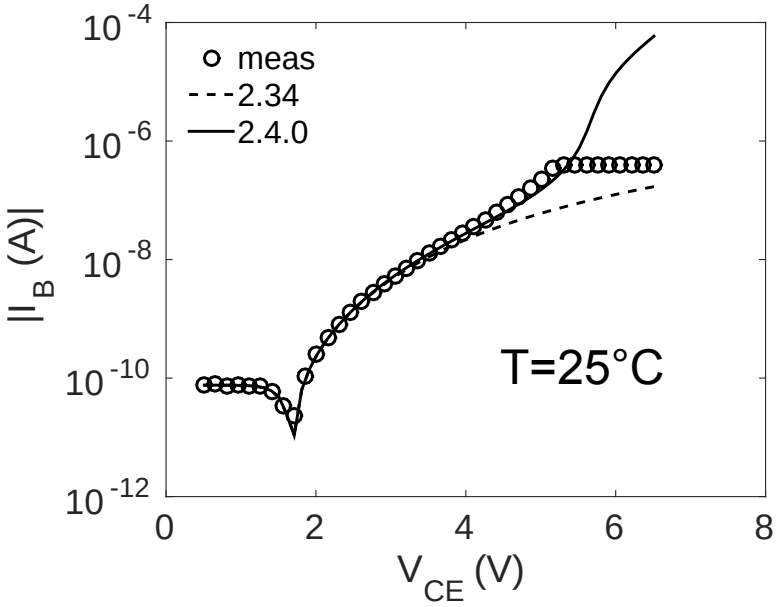
- No re-extraction of favl and qavl required
- Extraction of k_{avl} from

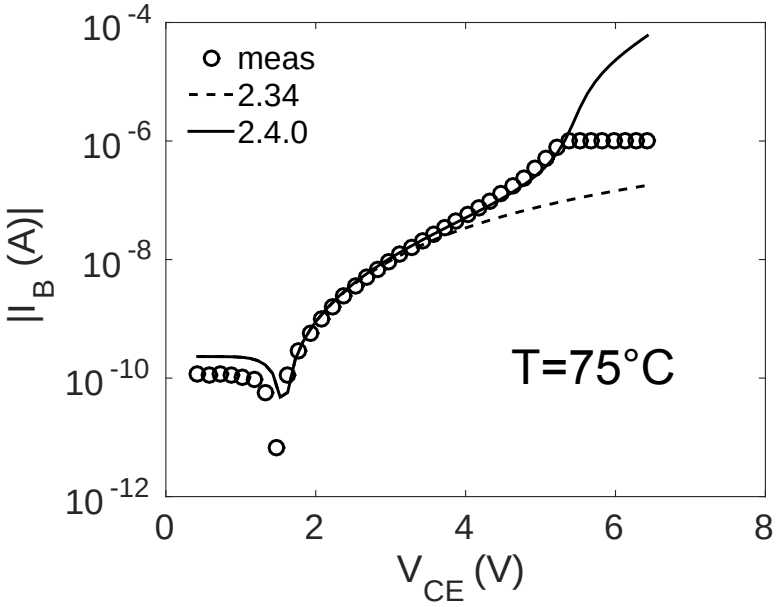
$$k_{avl} = \frac{1}{g} - \frac{1}{M-1}$$

=> simple extraction based on measured multiplication factor at high voltages

Backward compatibility and application

- Extension is completely compatible to previous HICUM versions!
- Values for weak avalanche model parameters fAVL and qAVL can be re-used to extend existing model cards by the new feature
- Application to experimental data
  - The same favl and qavl are used for 2.34 and 2.4.0
  - New parameters: kavl = 0.088, alkav=0.0037



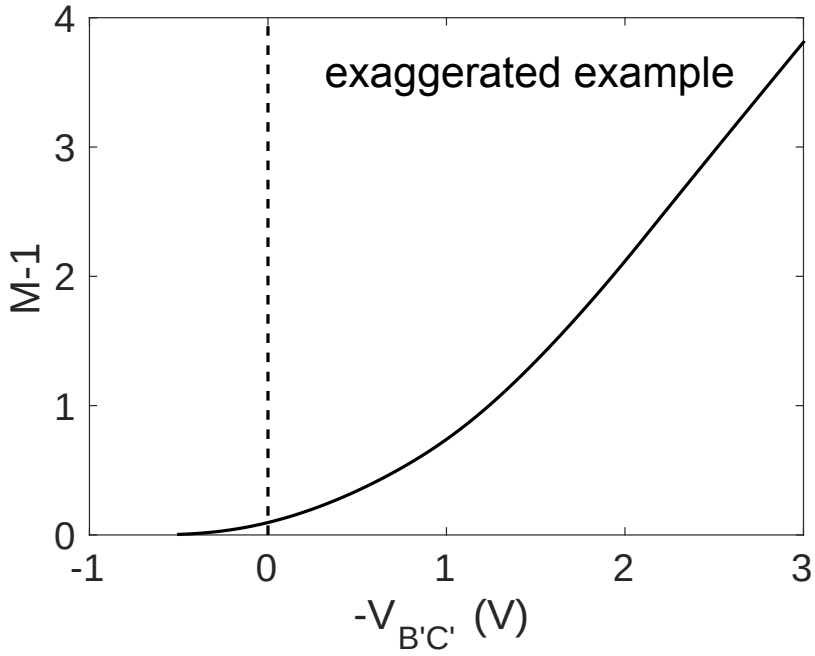


Conditional statement for weak avalanche breakdown

- Present conditional statement:

```
if((Vbici < 0.0) && (favl_t > 0.0) && (cjci0_t > 0.0)) begin
    ....
end else begin // (Vbici >=0)
    iavl = 0;
end
```

- Depending on the parameters, iavl can be greater 0 for Vbici = 0
  => possible discontinuity
  - note, in production model cards the effect is generally much less, e.g.
    M-1 = 1e-6 at Vbici = 0



Conditional statement for weak avalanche breakdown

- Test for actual voltage v_bord > 0
  - right-hand limit for v_bord (-> +0) is 0, as well as its derivative

```
if(use_aval == 1) begin : HICAVL // set in model evaluation
    real v_bord,v_q,U0,av,avl;
    v_bord  = vdci_t-Vbici;
    if (v_bord > 0) begin
        ...
    end else begin
        iavl = 0.0;
    end
end else begin
    iavl    = 0.0; // already set during model evaluation
end
```

Conditional statement for weak avalanche breakdown

flag during model intialization

- use_aval is set once during model initialization
  - now based on favl and cjci0 and not favl_t and cjci0_t

```
if ((favl > 0.0) && (cjci0 > 0.0)) begin
    use_aval = 1;
end else begin
    use_aval = 0;
    iavl = 0.0;
end
```

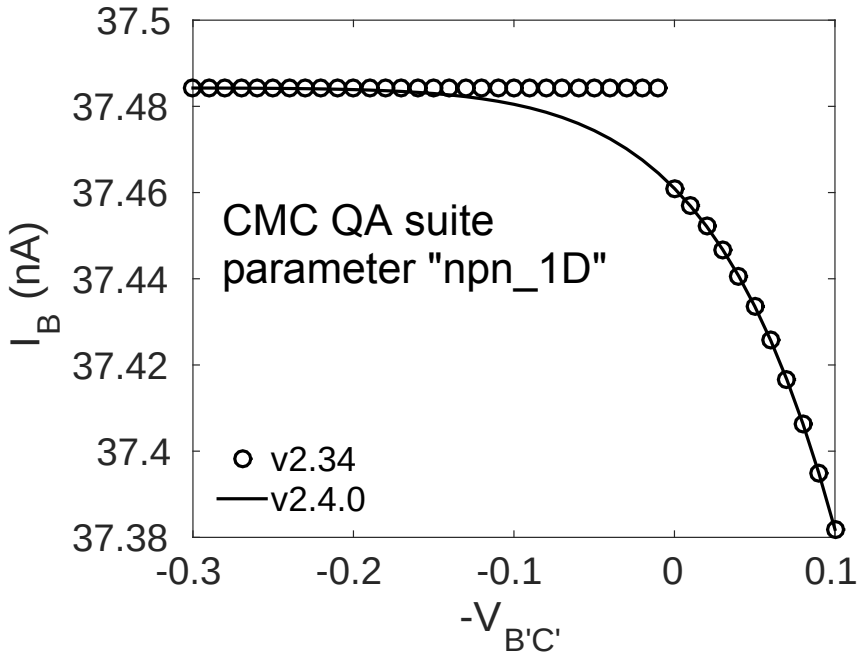
- Usage of an integer flag for faster evaluation of the if-statement

```
if(use_aval == 1) begin : HICAVL
```

- Settings `iavl = 0.0;` during model initialization avoids the additional else block for this if-statement

Results

- Tested with all simulations of HICUM/L2 CMC QA setup
  => no convergence issues with new formulation
- More smooth results for output curves



- Deviations introduced compared to HICUM/L2 version 2.34
  - maximum 0.0003% for DC quantities
  - maximum 0.065% for small-signal simulations

Miscellaneous changes

Increased parameter ranges

- Increase of the ranges for the model parameters ibets and ahc

  IBETS => [0:50]

  AHC => (0:50]

Gmin for operating point values

- For all conductances calculated in the section for operating point informations, Gmin is added to avoid divisions by zero for some of the derived values (e.g. BETAAC).

```
gPIi = type*ddx(ibei,V(bi))+type*ddx(irei,V(bi))+`Gmin;
gPIx = type*ddx(ibep,V(bp))+type*ddx(irep,V(bp))+`Gmin;
```

- This avoids a bug reported by IFAG for BETAAC becoming nan for zero gPIi and gPIx.
  - Added further conditional statement for BETAAC, if Gmin = 0.0 provided by the simulator

```
if (gPIi+gPIx > 0.0) begin
   BETAAC = GMi/(gPIi+gPIx);
end else begin
   BETAAC = 0.0;
end
```

Renamed operating point value names of resistances

- Replaced previous OP variables RE, RCX by **re_t and rcx_t**
  (as calculated in the model evaluation)

  => avoids a bug reported by IFAG for case insensitive simulators

  - model parameters re, rcx collide with OP variables RE, RCX in case-insensitive codes

  - For consistency, same change is applied to RBI and RB
    => Use of variables rbi and rb as calculated inside the model

Renamed operating point value names of resistances

- Model implementation

```
`ifdef CALC_OP
    (* desc="...", units="Ohm" *) real rcx_t;
    (* desc="...", units="Ohm" *) real re_t;
    (* desc="...", units="Ohm" *) real rbi;
    (* desc="...", units="Ohm" *) real rb;
`else
    real rcx_t, re_t, rbi;
`endif
```

  - note that variable rb is only required for output of operating point data

```
rb = rbi+rbx_t;
```

Calculation of Vciei

- Previous versions until v2.34

```
Vciei = V(ci,ei);
```

- For model compilers with derivatives based on branches, variables for the high-current region dependent on four branches
  - bi,ei
  - bi,ci
  - ci,ei
  - t,gnd
- Changed to

```
Vciei = Vbiei-Vbici;
```

  - Removes the dependence on the ci,ei branch
  - Reduces number of derivatives for each variable to three
- Results from Mentor, Keysight and Cadence
  - Reduced code size (2% to 7.5%) and model evaluation time (up to 10% for some cases)

No impact on model accuracy and results

Summary model parameters

- The following list summarizes the changes in the model parameters for HICUM from version 2.34 to 2.4.0

Parameter	default	range	remark
kavl	0	[0:3]	introduced in 2.4.0
alkav	0		introduced in 2.4.0
ibets	0	[0:50]	increased maximum
ahc	0.1	(0:50]	increased maximum