

Status of HICUM/L2 Model

A. Mukherjee, A. Pawlak, M. Schröter

CEDIC, University of Technology Dresden, Germany

Dept. of Electrical and Computer Engin., University of Calif. at San Diego, USA

http://www.iee.et.tu-dresden.de/iee/eb/eb_homee.html

13th HICUM Workshop

TU-Delft, NL

2013

Outline

- New Additions Hicum/L2 => Hicum/L2 Version 2.32
summary of code changes & issues
- Future plans for L2 Model
- Conclusion

Hicum/L2 Version 2.32 changes

- Hicum/L2v2p32 has been officially released on April 23, 2013
- Operating point values are calculated directly in VA code
 - Introduction of compiler flag CALC_OP

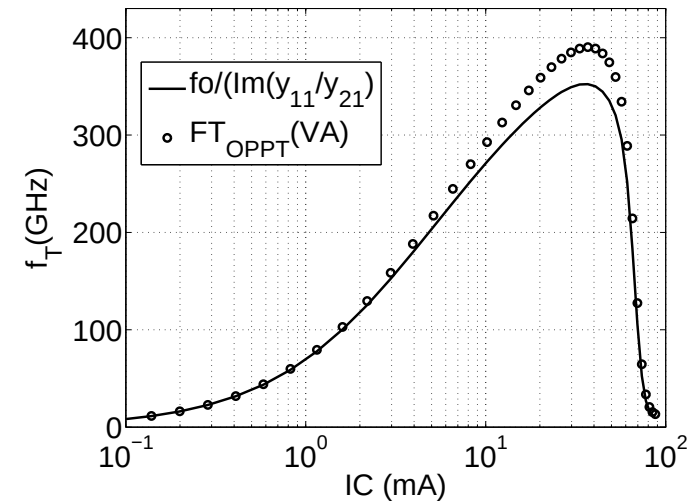
```
`define CALC_OP // turn off to remove complete OP sections from code  
...`  
`ifdef CALC_OP  
  (* desc="Base terminal current", units="A" *) real IB;  
`.`  
`endif`
```

=> calculations are included in the code only when CALC_OP is turned-on

- issues (presently discussed within CMC)
 - Calculation of OP-values also during transient simulation was requested
=> strong increase of simulation time
 - additionally requested parameters: actual device temperature, temperature increase, other (thermal resistance)?
 - See further slides for details

OP calculation

- Example for accuracy of analytical f_T calculation
- most parameters requested in OP list require *ddx* operator and more sophisticated calculations



=> significant impact on simulation time

- 10 transistors in parallel, 7881 DC operating points, mean value of five simulations

CALC_OP	Simulator	Size C-code	Size compiled code	Simulation time
defined	ADS	1527 KB (1.48)	615 KB (1.18)	22.6 s (1.37)
	Spectre	3983 KB (1.34)	1308 KB (1.17)	44.35 (1.31)
not defined	ADS	1029 KB (1)	522 KB (1)	16.45 s (1)
	Spectre	2971 KB (1)	1118 KB (1)	33.9 (1)

Code changes

Conditional statement for barrier term was changed
(in response to concern of one EDA vendor)

- Original conditional statement

```
` if (icbar<1e-10) begin\
```

```
...
```

=> changed to

```
` if (icbar<0.05*ick0) begin\
```

```
...
```

- V_{lim}/R_{Ci0} is a rough bias and temperature independent estimation of I_{CK}

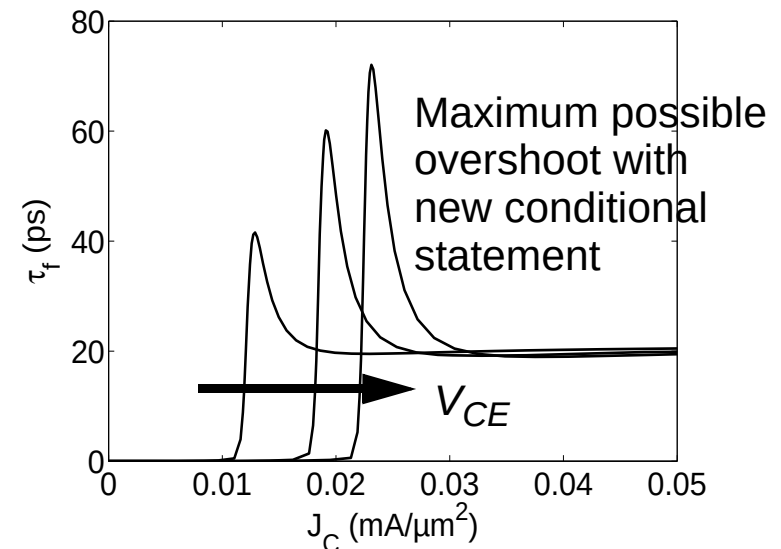
- limitation of the transit time overshoot (maximum possible overshoot shown in plot)

=> better convergence of GICCR iteration

- removal of potential numerical issues for too low (but non-physical) **ICBAR** values:

- Too low (high negative) values of FFib may lead to FFcbar=0 and therefore to a division by zero
- FFib $\approx$ -20 was chosen as arbitrary value securing numerical stability

=> limit **FFib** = (itf-I_{CK})/**ICBAR** for very small itf to $-I_{CK}/\mathbf{ICBAR} \approx -ick/(0.05*ick) \approx \mathbf{-20}$



Code changes (cont'd)

- The bug in the calculation of the flicker noise was resolved.

flicker_Pwr = *kf*pow((ibei+ibep),af);*

has been changed to

flicker_Pwr = *kf*pow(abs(ibei+ibep),af);*

Additional actions/issue

- Issues with the QA suite and ADS
 - Slightly different results for ADS with OP section turned on and off
 - Differences in the maximum resolution of floating values -> no influence on simulation results
 - Was listed by QA suite (obviously) as difference
 - Issue could not be reproduced by Agilent
 - Request to CMC to also check absolute differences rather than only relative
 - Don't result in "differ" if differences are as small as numerical precision

=> to be decided and fixed by CMC

- Implementation of constants in the model code
 - Request by simulator vendors to include physical constants in model code rather than to take the simulator values (by include constants.h)
 - Problem basically triggered by different constant definitions in different simulators
 - Leads to problems when using same model (parameters) in different simulators (frequent case in circuit design)

=> not suggested by HICUM model developer

Future plans

- Improve NQS and correlated noise section
 - VA only allows standard network elements
 - Second-order effects have to be implemented using adjunct networks
 - increases node count => appears to be slower than direct implementation in simulator code
 - Current implementation lead to capacitances of 1 F and branch voltages in the order of fV
 - May lead to convergence issues during transient simulation
 - Model improvement and/or suggestions to VA-compiler develop. for more efficient implementation
- Improvement of computational efficiency
 - increase computation speed by optimizing model equations and their implementation
 - Improved initial value of GICCR iteration loop
 - Replace V_T by V_{T0} in some smoothing functions (no derivative with respect to $V(\text{temp})$)
 - ...
 - provide suggestions for VA-compiler (developers) to result in faster code
 - Remove undesired and unnecessary derivatives
 - ...
- Manual update for released version
 - Simplification of the manual => Focus mostly on model equations
 - Remove of most of the physical background => already discussed in detail in HICUM book

Conclusion

- Provided overview on latest code changes
- Calculation of operating point inside verilog-A code can significantly increase computation time (appears to be somewhat dependent on simulator)
- Future action items for L2 model have been discussed