ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

New Verilog A Model Compiler for SPICE 3F5

1− Introduction

2− Features

3− Development

4− Examples :

4 − 1 : Simple R C model

4 − 2 : Bipolar : HICUM models
level 0 , level 2.1 , level 2.2 ...

4 − 3 : Mos : EKV 2.6 models

5− Remaining Works ...

6− Other Developments

7− SUMMARY of Works & results (extracts)

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

1− Introduction

=> Why Verilog A ?? (Main Goals) :

=> Rigth derivatives (Jacobian for Newton−Raphson...)

=> be sure of Right Matrix loading ...

=> simplify , decrease a lot the time, to implement a new model into SPICE

=> avoid the polemics when convergency PB :

=> no convergency or iteration number too high

=> low convergency : TRAN : too long simulation time

Where does convergency PB come from ??

=> from the model itself ??

=> from some bad derivatives in the reference C ??

=> from the bad implement from CAD supplier ??

=> from no Right Limiting Function strategy ??

=> Why not using Mot−Adms Verilog A compiler ?? :

=> I test the last 4 versions : Aug 2004 (Verilog A compiler from Laurent LEMAITRE...)

=> but :

=> not compatible with SPICE3F5 ... (could be in the future ?? using XML ??)

=> Aug 2004 : Works only for DC Analysis ?? .... (main problems for HICUM is TRAN ...)

=> do not support Verilog A Analog Function :

=> as : 1 return value Function :

ex: "ia = func_ia(V(a) , V(b) , xb) ;"

=> no IN & OUT parameters for Analog Function

(Std Verilog A : only IN parameters)

=> no return value Function :

ex: "func_ia(V(a) , V(b) , xb) ;"

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

2− Features supported in this new SPICE Verilog A compiler :

=> Errors , warnings (syntax && variables ...) decoding...
=> Only 1 module per Verilog A code (only 1 model per code)
=> today : only the Model way allowed in Verilog A code ...
=> no Device (instance) of Model (using std SPICE way)
=> no test banch ... (instead using the std SPICE way)

=> some features not yet implemented :
=> function idt()
=> V(n1,n2) <+ exp
(in fact only : I(n1,n2) <+ exp is needed in Model ...)
=> Noise Analysis ...

=> today : only available on PC LINUX

=>Verilog A Analog Functions
=> with return value
ex: "ia = func_ia(V(a) , V(b) , xb) ;"
=> with In−Out parameter values : "xb"
=> with no return value :
ex: "func_ia(V(a) , V(b) , xb) ;"

=> Device && Model parameter may be separated
=> "temp" && "area" are automatically added ...

=> "getveriloga" : SPICE cmd to load the V.A compiled model

=> Spice3F5 Analysis supported :
=> OP , DC and related Analysis
=> AC , PZ and related Analysis
=> TRAN
=> RF analysis (Harmonic Balance) : HB , QP , CE

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

3− Development phases : using some GNU tools

=> flex 2.5 (fast scanner generator , "lex" compatible)
=> bison 1.35 (YACC−compatible Parser Generator)

and starting from Bison examples : calc, ltcalc, mfcalc ...

3−1 be able to decode & generate 1 line of expression & Derivatives validate all features :
+ − * / = cos() Analogfunc() ... etc

3−2 be able to decode & generate all lines & expression & Derivatives of 1 model :
=> on simple models ex : RCMOD ....etc
=> on complex model : HIC213 ....

then validate all features : parameters, module ...etc

3−3 generate Matrix loading & files : for OP ...

then validate :
=> on simple circuits ex : RCMOD ....etc
=> on complex circuits using BJT : HIC213 ....

3−4 generate Matrix loading & files : for TRAN

then validate :
=> on simple circuits ex : RCMOD ....etc
=> on complex circuits using BJT : HIC213 ....

3−5 generate Matrix loading & files : for other Analysis :
AC , PZ , ... RF Analysis : HB, QP, CE

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

4 − 1− a : example of : Simple R C model

```
module modrc(a,b,nx,ny,nz) ;
  inout a,b ;
  electrical  a,b,ci,nx,ny,nz;
  real rx,iac,qcb,vab,vac,vcb ;

  device parameter real r = 1k ;
  device parameter real c = 1p ;
  model  parameter real vc1 = 0.0 ;
  model  parameter real vc2 = 0.0 ;

analog begin
  vab = V(a,b);
  vac = V(a,ci);
  vcb = V(ci,b);
  rx  = r * (1.0 + vc1*vab + vc2*(vab*vab)) ;
  iac = vac / rx ;
  qcb = c * vcb ;
  I(a,ci) <+ iac;
  I(ci,b) <+ ddt(qcb) ;
  I(ci) <+ V(ci)/1.0e12;
  I(nx) <+ V(nx)/1.0e12;
  I(ny) <+ V(ny)/1.0e12;
  I(nz) <+ V(nz)/1.0e12;
end
endmodule
```

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

4 − 1 − b : example of : Simple R C model

```
int
MODRCevaluate (a, b, ci, nx, ny, nz, cktin, herein, hboff)
double     a, b, ci, nx, ny, nz ;
CKTcircuit     *cktin;
MODRCinstance  *herein;
int   hboff;
{
   .............          Values & Derivatives setting
   vab.val=a−b ;
   vab.D_a=1.0 ;
   vab.D_b=−1.0 ;

   ..........................  Expressions  & Derivatives setting
    rx.D_a=r*(vc1*(vab.D_a)+vc2*(vab.val*(vab.D_a)+vab.val*(vab.D_a))) ;
    rx.D_b=r*(vc1*(vab.D_b)+vc2*(vab.val*(vab.D_b)+vab.val*(vab.D_b))) ;
    rx.val=r*(1.0+vc1*vab.val+vc2*(vab.val*vab.val)) ;

    ..................................  Matrix   loading   I_a_ci
     *(here−>MODRCINaAptr)  += I_a_ci.D_a ;
     *(here−>MODRCINaBptr)  += I_a_ci.D_b ;

    ..................................  RHS   loading   I_a_ci
     _ieq  = I_a_ci.val ;
     _ieq −= I_a_ci.D_a*a ;
     _ieq −= I_a_ci.D_b*b ;
    ..................................................
    (ckt−>CKTrhs[NUM_Na]) −= _ieq ;
    (ckt−>CKTrhs[NUM_Nci]) += _ieq ;
```

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

4 − 2 : example of Bipolar : HICUM models

HIC203 : HICUM level 0 (5 external nodes & Self Heating)

HIC213 : HICUM level 2.1 (5 external nodes & Self Heating)

=> numerical values results very close of previous HICUM2125 (classical manual implement from reference Fortran code)

=> good convergency behaviour when OP && TRAN (even for complex circuits)

=> still needs some Model specific limiting functions (specially for complex circuits)

notes : complex circuits means :
=> with a high number of BJT transistors
=> with high GAIN
=> with feedback ...
=> with saturated BJT

ex : the ST BJT ECL ring oscilator of 21 stages ...

4 − 3 : MOS : examples close of EKV 2.6 models :

EKV26VA : Verilog A close of ekv26_SDext_tiburon.va
EKV26VASH : idem EKV26VA & Self Heating

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

5− Remaining Works ...

=> Noise Analysis ...

=> remaining features : "idt ()" , "V(n1,n2) <+ exp" ....

=> How to introduce the Limiting functions needed when Strongly no linear model ??

=> Spice Verilog A Compiler for HP workstation under HPUX

6− Other Developments :

=> a lot of new features & improvement for Spice3F5

- Spice driven from schematic ...
- Subckt Parameters
- Process Block
- New Spice RF :
  - S parameter Analysis & port + nport
  - New Harmonic Balance Analysis :
    - HB : 1 forced frequency Steady State
    - QB : Quasi Periodic 2 forced frequencies Steady State
    - CE : Circuit Envelope 2 forced frequencies Transient Time domain

=> ATELECAD kit

- => schematic capture (SPICE driven by schematic capture ...)
- => logic simulation
- => Layout : MAGIC 6 :
  - => extended version for Bipolar : DRC etc ...
  - => new feature : component generators ....

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SOLVE CONVERGENCE PROBLEMS OF A NEW MODEL :

=> set the Model && Simulator in "Overflows & NAN & INF" debugging state ...
(checks & kill SPICE , if any of unwanted values exist when iterate...)

=> then debug the found Problems that comes :

1− from Model itself : any discontinuity in the functions :
Currents (applied to Nodes) = f (VoltageNodes)
D(Currents) / D(VoltageNodes) = f (VoltageNodes)
Charges (applied to Nodes) = f (VoltageNodes)
D(currents) / D(VoltageNodes) = f (VoltageNodes)

use 3 simple circuits :
circuit 1 => 1 transistor + 1 VSRC at each EXT Nodes

2− from Bad Matrix Loading : OP & DC & TRAN
=> Starting point ... init the calculation
=> when iterate : Limiting functions ...
circuit 2 => 1 transistor (diode: cb , es) + 1 VSRC
circuit 3 => 1 transistor (diode: cb , es) + 1 ISRC
(also checks symetry (NPN & PNP))
circuit 4 => 1 OTA & Active Load & mixed model
(also checks symetry of input drive)
circuit 5/6 => 1 Bandap & ECL Divide by 2
circuit 7/8 => ECL Ring Oscillator(21) Not Saturated
(2 supply voltages)
circuit 9/10 => ECL Ring Oscillator(21) Hard Saturated
(2 supply voltages)

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

ref :

mail1 (27 dec 2004) from J. C PERRAUD to Micheal Shroter
mail2 (19 avril 2005) from J. C PERRAUD to Micheal Shroter

SUMMARY of Works & results (extracts) :

=> all Circuits and Results files

=> all these data are availables at this HICUM Workshop for demos when somebody wants it ...

DIR_TEST_VERILOGA_EKV26 : Mos Model EKV level 2.6

DIR_TEST_VERILOGA_GUMMEL : BJT Model GUMMEL−POON

DIR_TEST_VERILOGA_HIC203 : BJT Model HIC203 (HICUM Level 0)

DIR_TEST_VERILOGA_HIC2125 : BJT Model HIC2125 (HICUM Level 2.1 − 5 ext nodes)

DIR_TEST_VERILOGA_HIC213 : BJT Model HIC213 (close of HICUM Level 2.1)

DIR_TEST_VERILOGA_HIC223 : BJT Model HIC223 (HICUM Level 2.2 − Self Heating OFF)

DIR_TEST_VERILOGA_HIC223_SHON : BJT Model HIC223 (HICUM Level 2.2 − Self Heating ON)

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 1
==========

OP , DC , ImpOP , TRAN => Nb of iterations
Time in seconds
(G1000) => Gmin Stepping using 1000 steps
NO => No convergency
(UIC + TTsmall) => No Implicit DC & Time Step Too Small during TRAN

	GUMMEL POON	HIC203	HIC203 No Lim	HIC2125	HIC213	HIC223 SH_OFF	HIC223 SH_OFF No Lim	HIC223 SH_ON	HIC223 SH_ON No Lim
VSRC									
OP	6	9	5	8	8	8	5	8	5
DC	23	47	34	35	41	42	29	43	30
ImpOP	4	10	3	5	10	10	3	10	3
TRAN	594	628	580	607	677	522	471	579	533
AnalTime	0.01	0.41	0.41	0.05	0.27	0.09	0.09	0.11	0.1
loadTime	0.01	0.4	0.4	0.05	0.24	0.06	0.08	0.08	0.09
ISRC									
OP	11	10	(G40) 351	10	11	8	(invalid) 82	11	(G1000) NO
DC	66	84	(G100) 681	178	82	63	481	85	(G1000) NO
ImpOP	26	26	5	24	25	25	3	25	(G1000) NO
TRAN	539	620	640	569	641	554	553	627	(UIC + TTsmall)
AnalTime	0	0.43	1.34	0.1	0.39	0.11	0.24	0.15	0.05
loadTime	0	0.42	1.33	0.1	0.36	0.09	0.21	0.15	0.03

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 2
==========

OP , DC , ImpOP , TRAN => Nb of iterations
Time in seconds
(G1000) => Gmin Stepping using 1000 steps
NO => No convergency
(UIC + TTsmall) => No Implicit DC & Time Step Too Small during TRAN

	GUMMEL POON	HIC203	HIC203 No Lim	HIC2125	HIC213	HIC223 SH_OFF	HIC223 SH_OFF No Lim	HIC223 SH_ON	HIC223 SH_ON No Lim
ECL Ring Osc − BJT NOT SATURATED Valim = 2.7 V									
OP	**6**	**10**	**(G10000) NO**	**8**	**13**	**11**	**(G1000) NO**	**11**	**(G1000) NO**
ImpOP	**323**	**34**	**(G40)172**	**11**	**32**	**29**	**(G1000) NO**	**29**	**(G1000) NO**
TRAN	**2428**	**3028**	**3067**	**6871**	**3124**	**3793**	**(UIC + TTsmall)**	**3842**	**(UIC + TTsmall)**
AnalTime	**1.84**	**112.27**	**189.89**	**28.71**	**83.49**	**39.36**	**9.42**	**41.72**	**9.79**
loadTime	**0.9**	**111.09**	**162.23**	**26.27**	**80.96**	**36.22**	**7.58**	**39.21**	**7.89**
ECL Ring Osc − BJT NOT SATURATED Valim = 5.2 V									
OP	**9**	**26**	**(G1000) NO**	**11**	**28**	**28**	**(G1000) NO**	**28**	**(G1000) NO**
ImpOP	**(G40) 309**	**52**	**(G40) 104**	**12**	**50**	**52**	**(G1000) NO**	**52**	**(G1000) NO**
TRAN	**2515**	**3172**	**(TTsmall) 3533**	**7769**	**4367**	**3723**	**(UIC + TTsmall)**	**4714**	**(UIC + TTsmall)**
AnalTime	**2.16**	**118.23**	**217.87**	**32.35**	**115.47**	**39.33**	**8.02**	**51.52**	**7.84**
loadTime	**0.83**	**117.1**	**186.83**	**29.53**	**112.42**	**37.02**	**6.65**	**48.07**	**6.56**

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 3
==========

OP , DC , ImpOP , TRAN => Nb of iterations
Time in seconds
(G1000) => Gmin Stepping using 1000 steps
NO => No convergency
(UIC + TTsmall) => No Implicit DC & Time Step Too Small during TRAN

ECL Ring Osc − BJT SATURATED
Valim = 2.7 V

	GUMMEL POON	HIC203	HIC203 No Lim	HIC2125	HIC213	HIC223 SH_OFF	HIC223 SH_OFF No Lim	HIC223 SH_ON	HIC223 SH_ON No Lim
OP	8	17	(G1000) NO	12	20	16	(G1000) NO	17	(G1000) NO
ImpOP	(G40) 323	34	(G40) 173	14	32	29	(G1000) NO	29	(G1000) NO
TRAN	15321	5933	(TTsmall) 6143	6791	5115	13296	(UIC + TTsmall)	16970	(UIC + TTsmall)
AnalTime	8.94	218.38	311.87	28.54	137.88	145.59	8.23	191.02	7.87
loadTime	4.45	216.3	282.01	25.68	134.22	137.00	6.74	178.87	6.6

ECL Ring Osc − BJT SATURATED
Valim = 5.2 V

OP	8	23	(G1000) NO	11	26	25	(G1000) NO	26	(G1000) NO
ImpOP	326	52	(G100) 340	13	50	34	(G1000) NO	52	(G1000) NO
TRAN	10657	6650	(TTsmall) 5855	8118	5746	12011	(UIC + TTsmall)	14323	(UIC + TTsmall)
AnalTime	7.62	244.43	305.52	33.9	150.49	125.52	7.92	157.82	7.94
loadTime	3.36	241.95	261.13	30.69	146.27	117.36	6.58	147.91	6.62

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 4
==========

OP , DC , ImpOP , TRAN => Nb of iterations
Time in seconds
(G1000) => Gmin Stepping using 1000 steps
NO => No convergency
(UIC + TTsmall) => No Implicit DC & Time Step Too Small during TRAN

	GUMMEL POON	HIC203	HIC203 No Lim	HIC2125	HIC213	HIC223 SH_OFF	HIC223 SH_OFF No Lim	HIC223 SH_ON	HIC223 SH_ON No Lim
ECL Ring Osc − BJT HARD SATURATED Valim = 2.7 V									
OP	**17**	**21**	**(G1000) NO**	**17**	**23**	**21**	**(G1000) NO**	**22**	**(G1000) NO**
ImpOP	**323**	**33**	**(G40) 172**	**14**	**32**	**30**	**(G1000) NO**	**30**	**(G1000) NO**
TRAN	**13408**	**7844**	**(TTsmall) 1884**	**11340**	**9996**	**15705**	**(UIC + TTsmall)**	**17198**	**(UIC + TTsmall)**
AnalTime	**8.25**	**286.6**	**95.31**	**49.24**	**275.4**	**166.08**	**8.00**	**191.72**	**7.87**
loadTime	**3.73**	**284.04**	**84.06**	**44.98**	**268.32**	**155.51**	**6.63**	**179.43**	**6.6**
ECL Ring Osc − BJT HARD SATURATED Valim = 5.2 V									
OP	**8**	**22**	**(G1000) NO**	**11**	**23**	**21**	**(G1000) NO**	**22**	**(G1000) NO**
ImpOP	**326**	**51**		**14**	**50**	**53**	**(G1000) NO**	**53**	**(G1000) NO**
TRAN	**12663**	**7579**	**(TTsmall) 5005**	**9272**	**9074**	**14252**	**(UIC + TTsmall)**	**15499**	**(UIC + TTsmall)**
AnalTime	**8.85**	**278.31**	**257.17**	**40.16**	**252.05**	**154.41**	**8.54**	**175.60**	**−−−−−**
loadTime	**4.08**	**275.58**	**228.15**	**36.99**	**245.28**	**144.09**	**6.98**	**163.34**	**−−−−−**

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 5
==================

Total of Simul time = Total of load Model Time + Total of Matrix Solving Time ;
or
Total of Simul time = Nb of Iterations * (load Model Time(1 Iter) + Matrix Solving Time(1 Iter))

to decrease the Total of Simulation time :

==> we should decrease the load Model Time(for 1 Iteration)

==> the Matrix Solving Time (for 1 Iteration) is said to vary as Matrix Size at a power of 3 (NbNodes ** 3)

the load Model Time(for 1 Iteration) depends of the Models Equations (defined by Model Leader and his team...)

=> Number of lines in Evaluate Model Function executed at each iteration ...

=> complexity of each lines of Evaluate Model Function

GUMMEL−POON :
−−−−−−−−−−−−−−−−−−−−−−−−

=> bjttemp.c : once per analysis , no Tsh derivative : 62 C lines
=> bjtload.c : once per iteration , no Tsh derivative : 672 C lines

HIC203 (HICUM level 0 from Verilog A) :
−−

=> hic203.va : 495 Va lines
=> hic203eq.cpp : 568 C lines (c, b, e, s, ci, bi, ei, tsh)
d(I) / d(c) .... etc & d(Q) / d(c).... etc

=> hic203eq.c : once per iteration , no Tsh derivative : 3369 C lines
it includes 4 C procedures :

Qjfun : 265 C lines 3 ? times per iteration
Cjfun : 254 C lines 2 ? times per iteration

cjtfun : 50 C lines 4 ? times per iteration
Cjfun : 4 0 C lines 4 ? times per iteration

but arround ?? 4423 C lines per iteration

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 6
==========

HIC213 (HICUM level 2.1 from Verilog A) :

=> hic213.va : 1000 Va lines but comments ... def macros ...
=> hic213eq.cpp : 692 C lines (c, b, e, s, ci, ei, bp, bi, si, tsh)
d(I) / d(c) .... etc & d(Q) / d(c).... etc

=> hic213eq.c : once per iteration , no Tsh derivative : 20132 C lines
(but a lot of IF case ... so all the lines are not executed
at each iterations ...)

it includes 2 C procedures :

HICRBI : 236 C lines ?? times per iteration
HICFCI : 222 C lines ?? times per iteration
HICFCT : 46 C lines ?? times per iteration

but difficult to evaluate the number of C lines per iteration

HIC223 (HICUM level 2.2 from Verilog A) :

=> hic223.va : 1100 Va lines but comments ... def macros ...
=> hic223eq.cpp : 668 C lines (c, b, e, s, ci, ei, bp, bi, si, tsh)
d(I) / d(c) .... etc & d(Q) / d(c).... etc

=> hic223eq.c : once per iteration , no Tsh derivative : 14940 C lines
(but a lot of IF case ... so all the lines are not executed
at each iterations ...)

it includes 2 C procedures :

HICRBI : 215 C lines ?? times per iteration
HICFCI : 150 C lines ?? times per iteration

but difficult to evaluate the number of C lines per iteration

ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 7
===================

CONCLUSIONS :
=============

1− see files ringXXX_ecl_YYY.cir and ringXXX_ecl_YYYEXT.cir :
−−−

when complex circuit (high gain or active load ...etc) you cannot get directly convergency for NO linear models (WITHOUT limiting functions) , specially when when circuit is driven by assymetric inputs ...

More the circuits are strongly NOT linear with High Gains and Assymetric inputs , More you will get convergency problem when using models WITHOUT limiting functions ....

But sometimes , you may get convergency using tricks as GMIN stepping or SOURCE stepping ...

When you get convergency to a realistic point , numerical results are the same ...

SAme conclusions applies also to Mos models (See DIR_TEST_VERILOGA_EKV26 ...)

2− In general , Now , All HICUM model gives low Number of iteration (as it is now in my SPICE)

BUT a relatively High loadTime (as it is now in my SPICE), as compared to the GUMMEL−POON one's...

3− GUMMEL−POON : (with the new limiting strategy , as it is now in my SPICE)

=> it gives a low number of iterations in OP , DC , and TRAN

=> it gives also very low loadTime ...

=> but it is a relatively simple model without self−heating :

=> about Half of the Model line equation are in the BJTtemp() function that is called 1 time per analysis

=> because of NO self−heating there is no need to compute derivative again temperature and load the matrix again thermal node

so a more accurate Model with self heating would have normaly a loadTime more important that could be at least , about 2*2 = 4 times more than the GUMMEL−POON loadTime ...

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 8
==========

4− HIC2125 (HICUM level 2.1 from Fortran : Self Heating ON): (as it is now in my SPICE)

=> it gives a very low number of iterations in OP , DC , and TRAN

=> it gives also a loadTime of about :

==> 30 * GUMMEL−POON−loadTime for NO Saturated Ring−Oscillator circuit

==> 10 * GUMMEL−POON−loadTime for HARD Saturated Ring−Oscillator circuit

5− HIC213 (Close of HICUM level 2.1 from Verilog A : Self Heating ON) :
(as it is now in my SPICE , without Excess Phase ...)

=> it gives a low number of iterations in OP , DC , and very low in TRAN

=> it gives also a loadTime of about :

==> 3 * HIC2125−loadTime for NO Saturated Ring−Oscillator circuit

==> 6 * HIC2125−loadTime for HARD Saturated Ring−Oscillator circuit

note :
=> to be investigated : why loadTime differences vary from No Saturated to HARD Saturated BJT ?? Could be improved ??

6− HIC203 (HICUM level 0 from Verilog A : Self Heating ON) : (as it is now in my SPICE)

=> it gives a low number of iterations in OP , DC , and in TRAN

=> it gives also a loadTime of about :

==> 4 * HIC2125−loadTime for NO Saturated Ring−Oscillator circuit

==> 6.5 * HIC2125−loadTime for HARD Saturated Ring−Oscillator circuit

ENSICAEN
ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN
& CENTRE DE RECHERCHE

Perraud Jean Claude 26 mai 2005 < perraud@ecole.ensicaen.fr >

SUMMARY of results : part 9
==========

7− HIC223 (HICUM level 2.2 from Verilog A : Self Heating OFF) :
(as it is now in my SPICE)

=> it gives a low number of iterations in OP , DC , and in TRAN

=> it gives also a loadTime of about :

==> 1.4 * HIC2125−loadTime for NO Saturated Ring−Oscillator circuit

==> 3 * HIC2125−loadTime for HARD Saturated Ring−Oscillator circuit

8− HIC223_SHON (HICUM level 2.2 from Verilog A : Self Heating ON) :
(as it is now in my SPICE)

results close of previous one's ....

9− some Optimisations of the Model Equation lines exits today in my Verilog A compiler , but obviously for the final model implement , manual optimisation should be done ..

10− may be , optimisation the existing Verilog A code should be done also , before ?? ... (by limiting the line equations and expressions at the minimun , avoiding not used codes ...)

11− We may probably thinck that the Verilog A code of Level 0 (HIC203) is not optimised ??

: this simple model gives more loadTime that the complex one's ???