



# Dynamic Electrical Rule Checks

---

**Dolphin Integration**

39 avenue du Granier  
BP 65 ZIRST  
38242 Meylan Cedex

Tel : (33) 4 76 41 10 96  
Fax : (33) 4 76 90 29 65  
E-mail : [medal@dolphin-integration.com](mailto:medal@dolphin-integration.com)

# What is dERC?



- dERC stands for **d**ynamic **E**lectrical **R**ule **C**hecks
- Electrical Rule Checks verify that invariants are respected according to circuit design or technology related specifications.
- As opposed to static checks, dynamic electrical rule checks are dynamically applied during simulation:
  - At operating-point
  - At every point of a DC analysis
  - At every time step of the transient analysis

# When is dERC needed?



- Whenever invariants need to be checked in a SPICE circuit description, such as:
  - High voltage
  - Low power
  - ...
- In order to perform specification or performance related checks that are not otherwise performed.
- In order to detect errors and solve them early in the design process.

# How is dERC used?



- The rules are defined by the designer in the pattern file (or included using external rule definitions).
- Each electrical rule check is applied:
  - Either to specific components
  - Or to all components that use specific models
- Whenever a violation is detected, a message is issued in the simulation log indicating:
  - Which device violated the rule (**where**)
  - Which values caused the violation (**why**)
  - The time or time interval of the violation (**when**)

# Definition of Checks



```
SPICE4.4J
File Edit View Design Simulate Waveforms Tools Windows Help
E:\cv\smash-examples\spice\ERC\ERC_on_JMOD.sp4

* SHASH
* SOLFEM
* INTEGRATION
* Copyright (c) Dolphin Integration. All rights reserved.
* Unauthorized use, duplication or distribution is strictly prohibited by law.
*
* Description:
* JFET 'amplifier'
*
-----
Pls ENTERE MODEL 10
Cls MODEL GATE 100u
BY GATE 0 JMOD
JFET BRIDGE GATE BRIDGE JMOD
EG SHWDE 0 4003
CG SHWDE 0 1uF
RPI VCC SWDE 1800
OCK SWDE OUT 180u
SCE OUT 0 RCHARGE
RAIIN VCC PWATE 10M

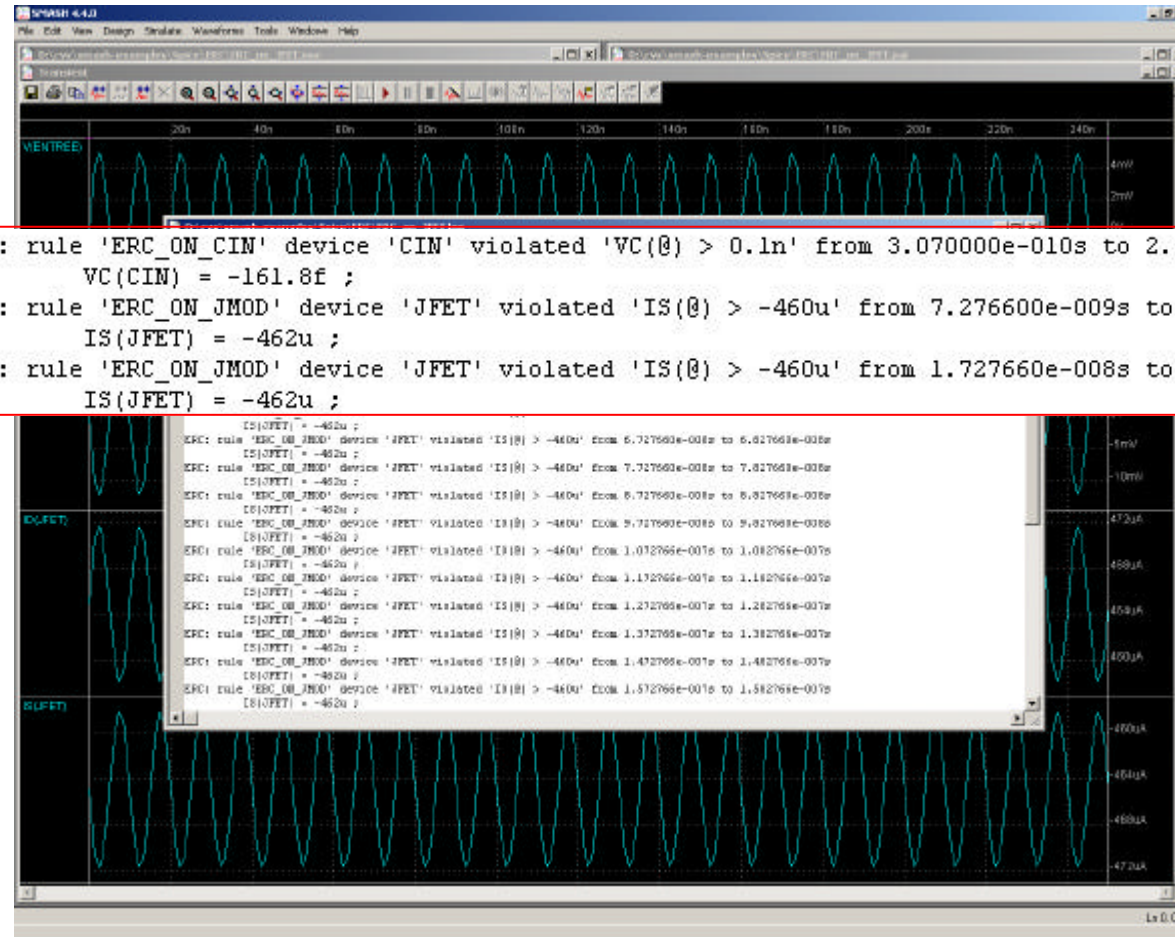
* SHASH
* SOLFEM
* INTEGRATION
* Copyright (c) Dolphin Integration. All rights reserved.
* Unauthorized use, duplication or distribution is strictly prohibited by law.
*
* Description:
* Please run this example if you are interested in:
* - analog simulation with Electrical Rules Checks
* - a standard SPICE example
*
* Type of analysis:
* - AC, TRANSCIENT
*
* To run both analyses:
* - Simulate -> Transient -> Run
*
-----
* Jfet model description *
.MODEL JMOD NMF
+LEVEL=1
+FTD = -2.0
+GETA =1003
+KREDA=0.0
+TI=1.9e-14
+FE=1
.PARAM RCHARGE=1200M
VIN ENTERE 0 SWE| 0 SW 1800M 0 0 0 | AC 1 0
VCC VCC 0 15
.KFS 1u 100u 10u
.H 1u 10u 10u 250u 2
.NETW3 TRAP
.OF ERC_V=lv VHEM=-12 VHEM=12 DELTA7=2.8 EPS_I=100y MAGSTEP=500 REPERSTIC9=-4 TRVALL=70

* electrical rule checks
.ERC NAME = ERC_ON_JMOD
+MODEL=JMOD CHECK='IS(@) > -460u'

.ERC NAME = ERC_ON_CIN
+DEVICE=Cin CHECK='VC(@) > 0.1n'
.CHECK ERC=YES TW=20p WLIMIT=200
```

\* Check whether the saturation current is below -460uA for JMOD model

# Test & Diagnostic



The simulation determines where, why and when for accelerating circuit debugging

# ERC directive syntax



- The `.ERC` directive defines electrical rules for devices, model parameters and expressions.

<code>NAME=" &lt;name&gt; "</code>	Specifies an optional name or description which will appear in the simulation log file in order to identify the rule.
<code>DEVICE=" &lt;device&gt; "</code>	Specifies the name or a wildcard name of the device(s) to be checked.
<code>MODEL=" &lt;model&gt; "</code>	Specifies the name of a model or model class (NMOS, PMOS, NPN, PNP, NJF, PJF, D, R, C) to be checked
<code>CHECK=' &lt;expression&gt; '</code>	Instance expression to be checked
<code>TW=&lt;time width&gt;</code>	Specifies the optional minimum simulation time during which a rule must be violated before warnings are issued. All checks of a given rule are impacted by the value provided. The default value is 0.
<code>ANALYSIS=TRAN   DC</code>	Specifies the analysis for which the check is to be performed.
<code>DEBUG=WARNING   PAUSE   ABORT</code>	Specifies the action to be performed when a violation is detected. The default action is to output a warning to the simulation log file. It is also possible to pause or abort the simulation, in which case the warning is also output.

# ERC directive syntax



- The syntax of the expressions is the standard mathematical expressions syntax with the special character @ as the placeholder for the instance being checked.
- Expressions can contain combinations of:
  - Logic operators (>, >=, <, <=, ==, !=, ?:, &&, ||)
  - Mathematical operators (+, -, \*, /, %, ^, \*\*)
  - Mathematical functions (abs, asin, acos, atan, cos, cosh, exp, ln, log, log10, min, max, mod, p10, power, sgn, sign, sin, sinh, sqr, sqrt, tan, tanh, time, valif, x)
  - Parameters (.PARAM)
  - Internal variables
  - Terminal currents, voltages and terminal potentials

# ERC directive syntax



## ● Examples

- `.ERC MODEL=HVPGD60 CHECK='(VDS(@) > -50) && (VDS(@) < 2)' TW=0.5`
- `.ERC MODEL=HVPGD60 CHECK='(VGS(@) > -60) && (VGS(@) < 2)'`
  
- `.ERC MODEL=HVPGD60 CHECK='(VGS(@) > -16) && (VGS(@) < 5)' TW=0.5`
- `.ERC MODEL=HVPGD60 CHECK='(VGS(@) > -20) && (VGS(@) < 5)'`
  
- `.ERC MODEL=HVPGD60 CHECK='(VD(@) > -2) && (VD(@) < 60)' TW=0.0005`
- `.ERC MODEL=HVPGD60 CHECK='(VD(@) > -2) && (VD(@) < 65)'`
  
- `.ERC MODEL=HVPGD60 CHECK='(VS(@) > -2) && (VS(@) < 60)' TW=0.0005`
- `.ERC MODEL=HVPGD60 CHECK='(VS(@) > -2) && (VS(@) < 65)'`

# CHECK directive syntax



- The **.CHECK** directive activates the checking of defined electrical rules specified with the **.ERC** directive .

<code>ERC=YES   NO</code>	Specifies whether or not to perform Electrical Rule Checks during the simulation.
<code>TSTART=&lt;start&gt;</code>	Specifies the time at which checking must start. The default start is the first time point of the analysis.
<code>TSTOP=&lt;stop&gt;</code>	Specifies the time at which checking must stop. The default stop is the last time point of the analysis.
<code>WLIMIT=&lt;count&gt;</code>	Specifies the maximum number of warnings to be output to the simulation log file. When the maximum number of warnings is reached, the simulation continues but the checks are no longer performed.