High-Level Description of Thermodynamical Effects in the EKV 2.6 MOST Model

Christophe Lallement *,
François Pêcheux *,
Wladek Grabinski **

** Laboratoire CNRS PHASE UPR 292, Strasbourg FRANCE*
*** Motorola, Geneva Modeling Center, Switzerland*

Presentation overview

- Goal : Device modeling with competitive HDL's :
  - VHDL-AMS and Verilog-AMS
- 7 Description tasks for modeling devices
- Task #2 : Describing analog behaviour
  - Modeling the MOST
  - Including Submicron effects
- Task #3,4 : Mixing several disciplines
  - Including Thermal-electronic interaction
- Task #7 : Incremental design
  - The resulting testbench
- Ongoing researches

VHDL-AMS and Verilog-AMS

- Hierarchical description of continuous and discrete interconnected models
- Modeling at various abstraction levels in electrical, thermal, mechanical and fluidic energy domains
- Discrete (digital) **AND** continuous (analog) time
- Conservative (GKL) **AND** signal-flow
- Ordinary Differential Algebraic Equations (ODAE)
- Recent approval by the IEEE Standard for VHDL-AMS

HDL's Mixed-Signals Description tasks

- Task #1 : Digital modeling of leaf sub model

Task #2 : Analog modeling of leaf sub model

Task #3 : Mixing several disciplines in the same model

Task #4 : Interfacing a sub model with others using conservative semantics

- Task #5 : Signal-flow interfacing
- Task #6 : A/D and D/A interfacing

Task #7 : Instantiation, parameterization and hierarchy

Task #1 : Describing continuous-time analog behavior

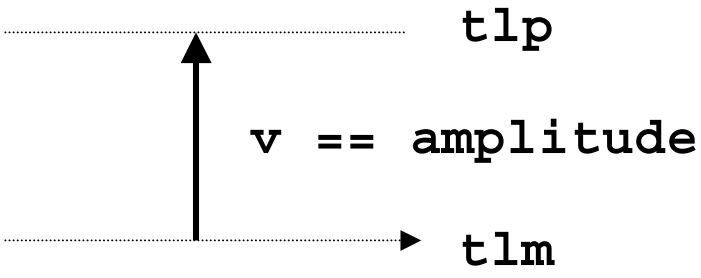
- Can be viewed as a set of ODAEs
- That have to be solved simultaneously by a specific continuous-time simulation kernel, called the analog solver.
- Most of the time, behavioral model follows conservative rules (GKL).

A simple submodel: supply

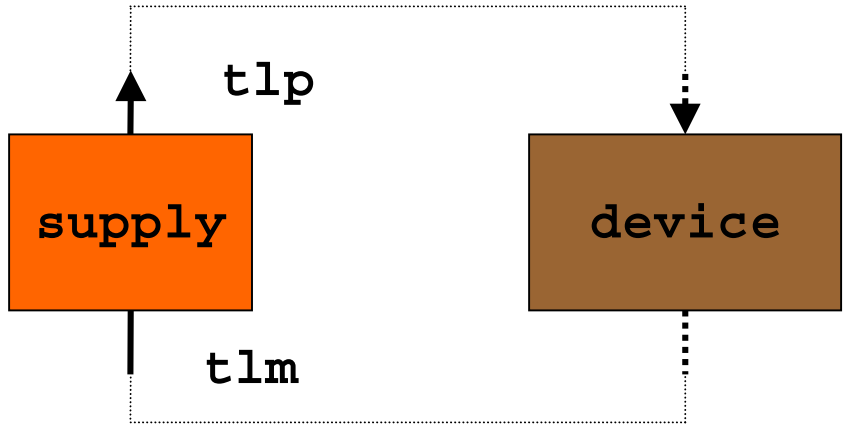
```
architecture equ of supply is
quantity v across i through tlp to tlm;
constant amplitude : real := 3.0;
begin
   v == amplitude ;
end;
```

```
`include "disciplines.h"
`include "constants.h"

module supply(tlp,tlm) ;
inout tlp,tlm ;
electrical tlp,tlm ;

parameter real amplitude=1.0 ;

analog begin
          V(tlp,tlm) <+ amplitude ;
end
endmodule
```

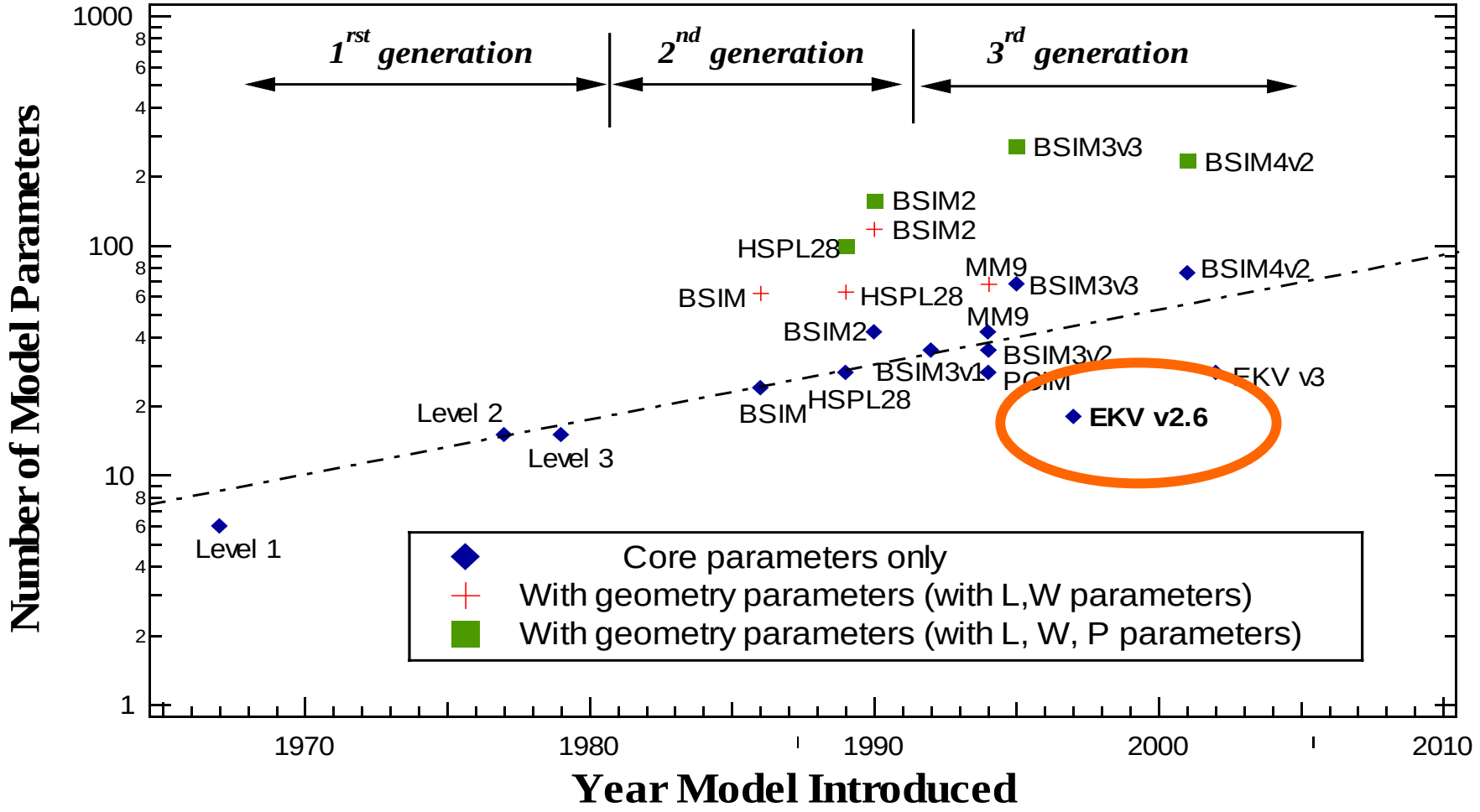
VHDL-AMS



Verilog-AMS

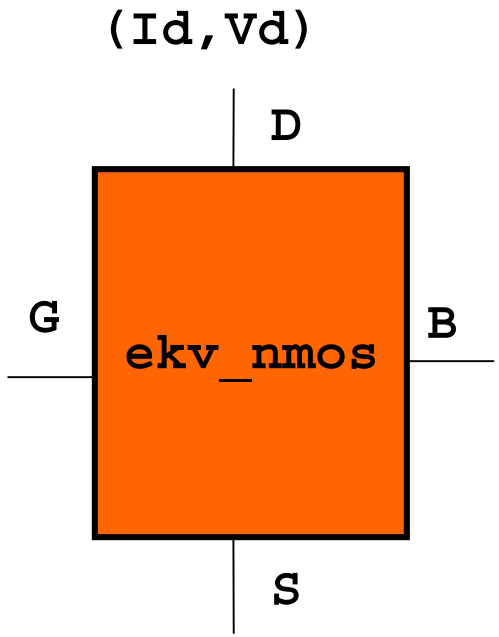


Different families of MOST models



Device modeling : the EKV 2.6 model

- Bulk referenced (source, drain symmetry)
- Charge-based model (current Id is computed through node charges)



The EKV model v2.6

- Original : **legwww.epfl.ch/ekv**
- Deep submicron** (Modeling of overlap resistance and capacitance)
- Thermo-electronic effects

**

•VHDL-AMS Design of a MOST Model Including Deep Submicron and Thermal-Electronic Effects Behavioral Modeling And Simulation (BMAS). Proceeding 2001, IEEE/ACM International Workshop on, pp. 91-96, 2001. C. LALLEMENT, F. PECHEUX et Y. HERVE

•A VHDL-AMS Case Study: The Incremental Design of an Efficient 3rd generation MOS Model of Deep Sub Micron Transistor. 11ème Conférence internationale 'on Very Large Scale Integration The Global System On Chip Design & CAD' (*VLSI-SOC*, 2001). Proceedings 2001, pp. 467–472, 2001. C. LALLEMENT, F. PECHEUX et Y. HERVE.

The `ekv_nmos` submodel interface

```
use disciplines.electromagnetic_system.all;
library ieee;
use ieee.math_real.all;

entity ekv_nmos is
 port (terminal d,g,s,b : electrical);
end;

architecture equ of ekv_nmos is
 quantity vg across g to b;
 quantity vd across id through d to s;
 quantity vs across s to b;
 quantity isource through s;
```

Bound/branch quantities

VHDL-AMS

```
`include "disciplines.h"
module ekv_nmos(d,g,s,b);

inout d,g,s,b,j ;
electrical d,g,s,b ;
```

Verilog-AMS

The `ekv_nmos` equations

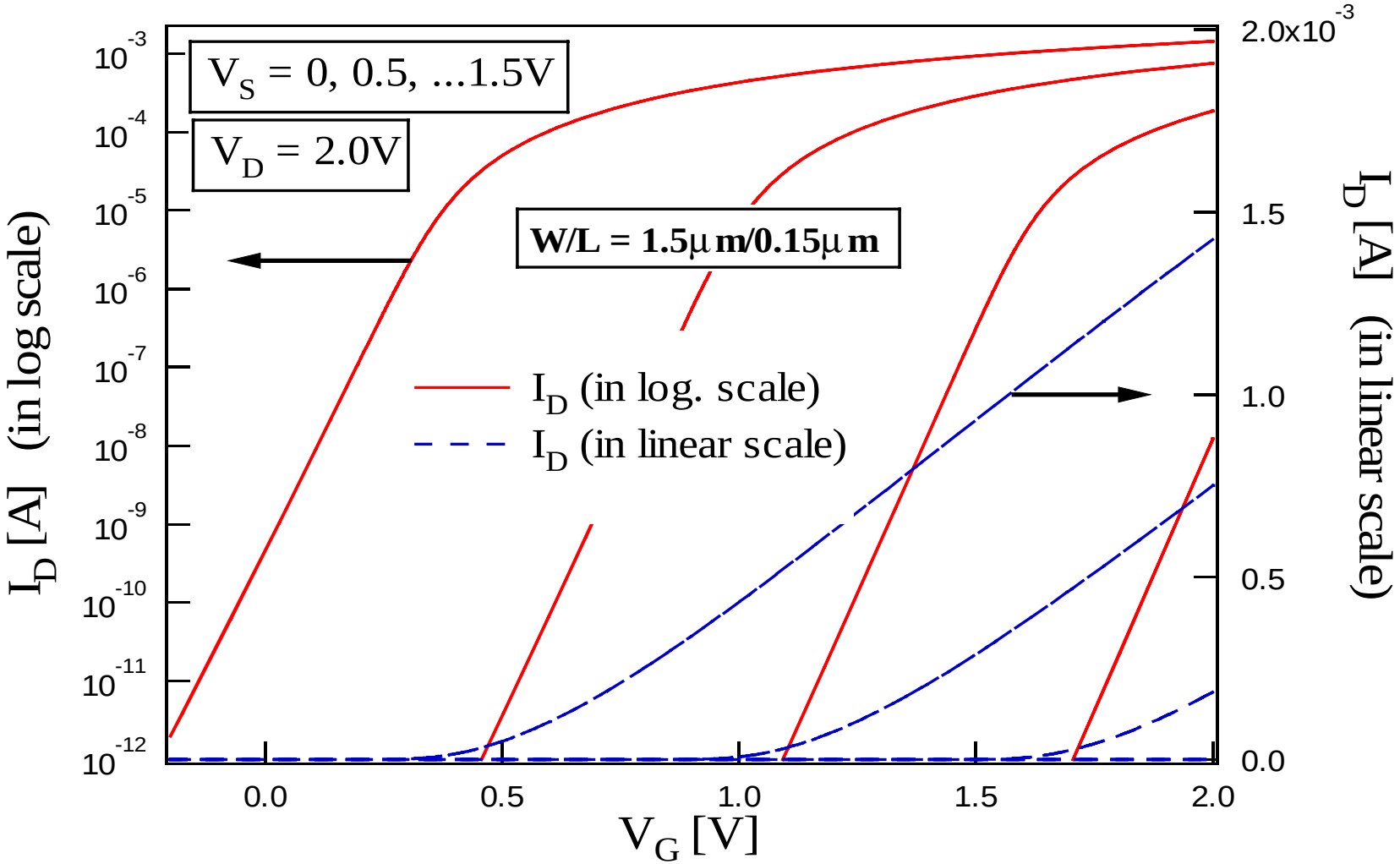
```
function F(v:real) return real is
begin
 return (log(1.0+exp(v/2.0)))**2;
end;

begin
 VGprime == vg - vto+phi+gamma*sqrt(phi) ;
 VP==VGprime-phi-gamma*
    (sqrt(VGprime+(gamma/2.0)**2)
    -(gamma/2.0));
 if_ == F((VP-vs)/vt);
 ir == F((VP-vd)/vt);
 beta == kp * (weff/leff)*(1.0/(1.0+theta*vp));
 n==1.0+(gamma/(2.0*sqrt(VP+phi+4.0*vt)));
 iss == 2.0*n*beta*vt*vt ;
 id == iss*(if_-ir) ;
 isource == -id ;
end ;
```

```
analog begin // EKV v2.6 long-channel

VG = V(g); VS = V(s); VD = V(d);

VGprime = VG - VTO + PHI + GAMMA * sqrt(PHI);

VP = VGprime - PHI - GAMMA
* (sqrt(VGprime+(GAMMA/2.0)*(GAMMA/2.0) ..

n = 1.0 + GAMMA / (2.0*sqrt(PHI + VP + 4.0..

beta = KP * (W/L) * (1.0/(1.0 * THETA * VP));

x=(VP-VS)/$vt; iff = (ln(1.0+exp( x /2.0)..;
x=(VP-VD)/$vt; ir = (ln(1.0+exp( x /2.0))..;

Ispec = 2 * n * beta * $vt * $vt;

Id = Ispec * (iff - ir);

I(d) <+ Id;

end // analog
endmodule
```

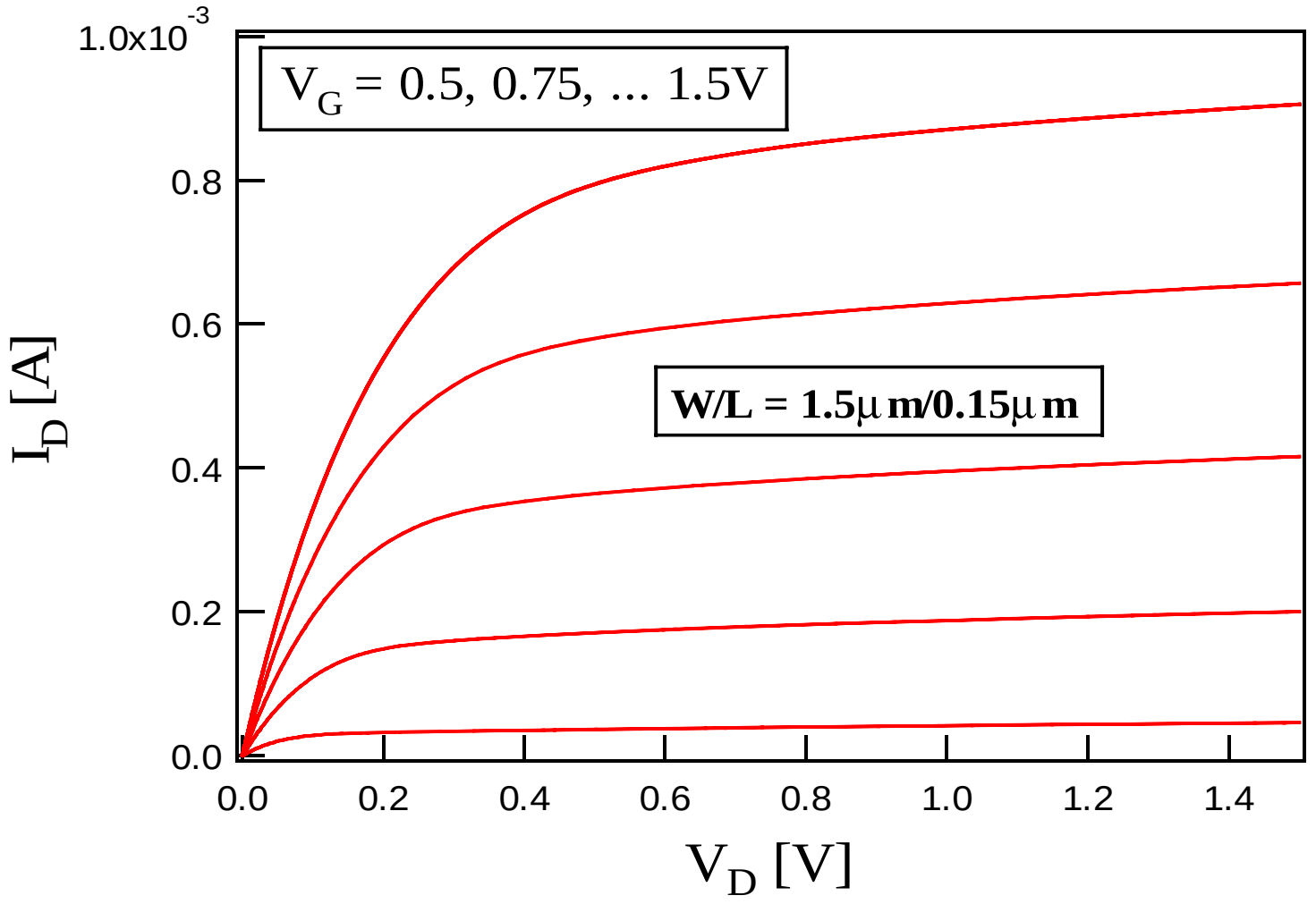
Branch contribution operator

http://legwww.epfl.ch/ekv/verilog-a/

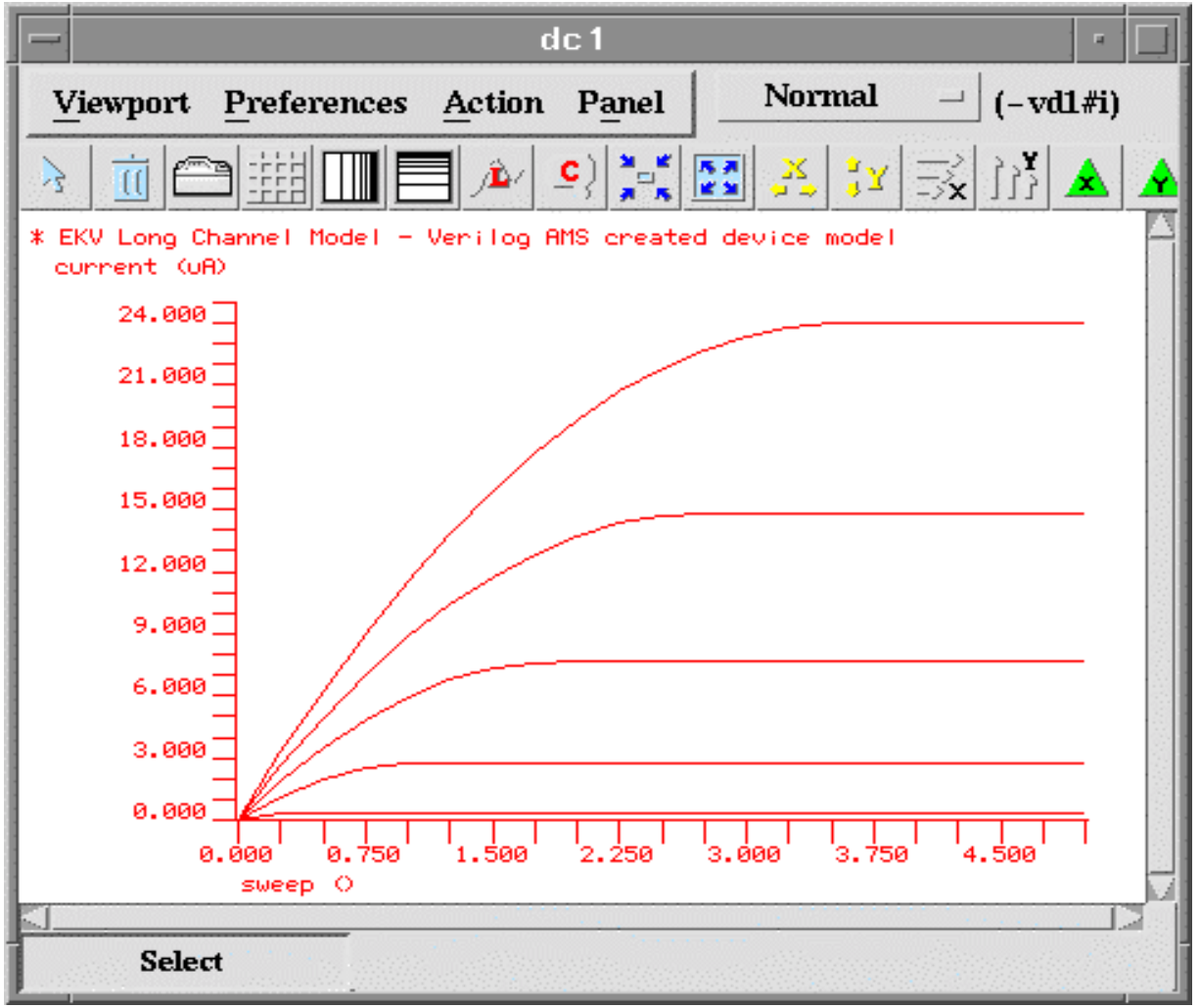
Static simulation : Id-Vg vs. Vs characteristic



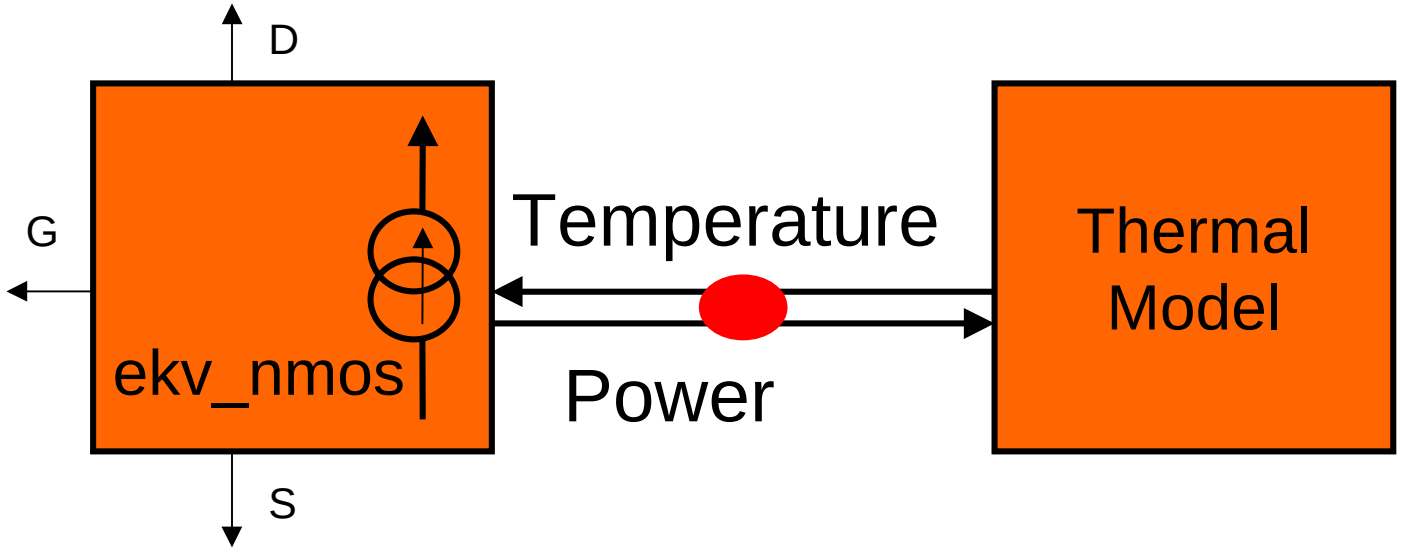
Static simulation : Id-Vd characteristic



Static simulation in Verilog-AMS : Id-Vd characteristic



Task #3,4 : Mixing several disciplines
Thermo-electronic interaction



Introducing a 5th node, of « thermal » type

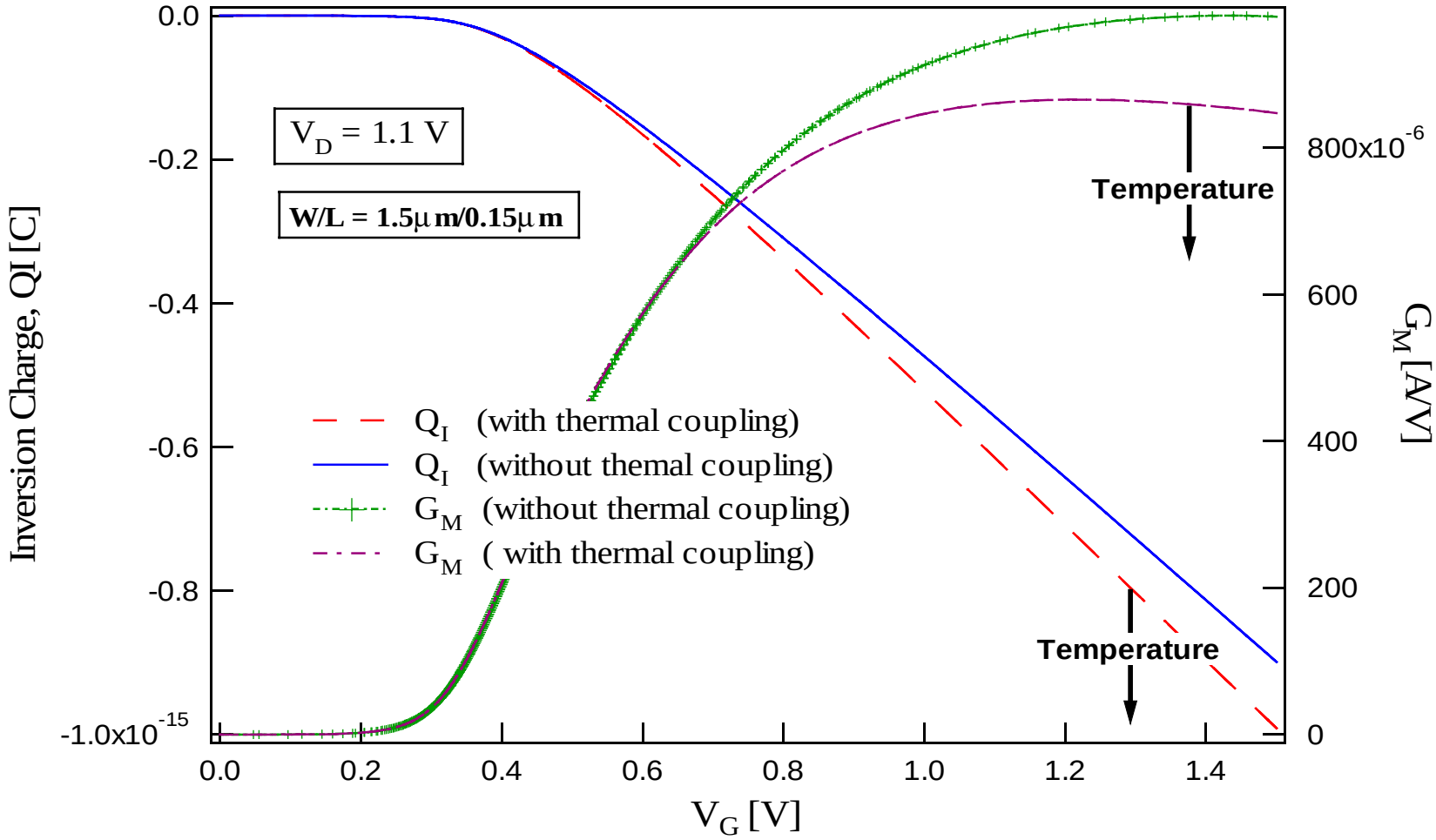
The ekv_nmos submodel (interface)

```
use disciplines.electromagnetic_system.all;
library ieee;
use ieee.math_real.all;

entity ekv_nmos is
 port (terminal d,g,s,b : electrical;
       terminal j : thermal);
end;

architecture equ of ekv_nmos is
 quantity vg across g to b;
 quantity vd across id through d to s;
 quantity vs across s to b;
 quantity isource through s;
 quantity temp across power
   through j to thermal_ground;
```

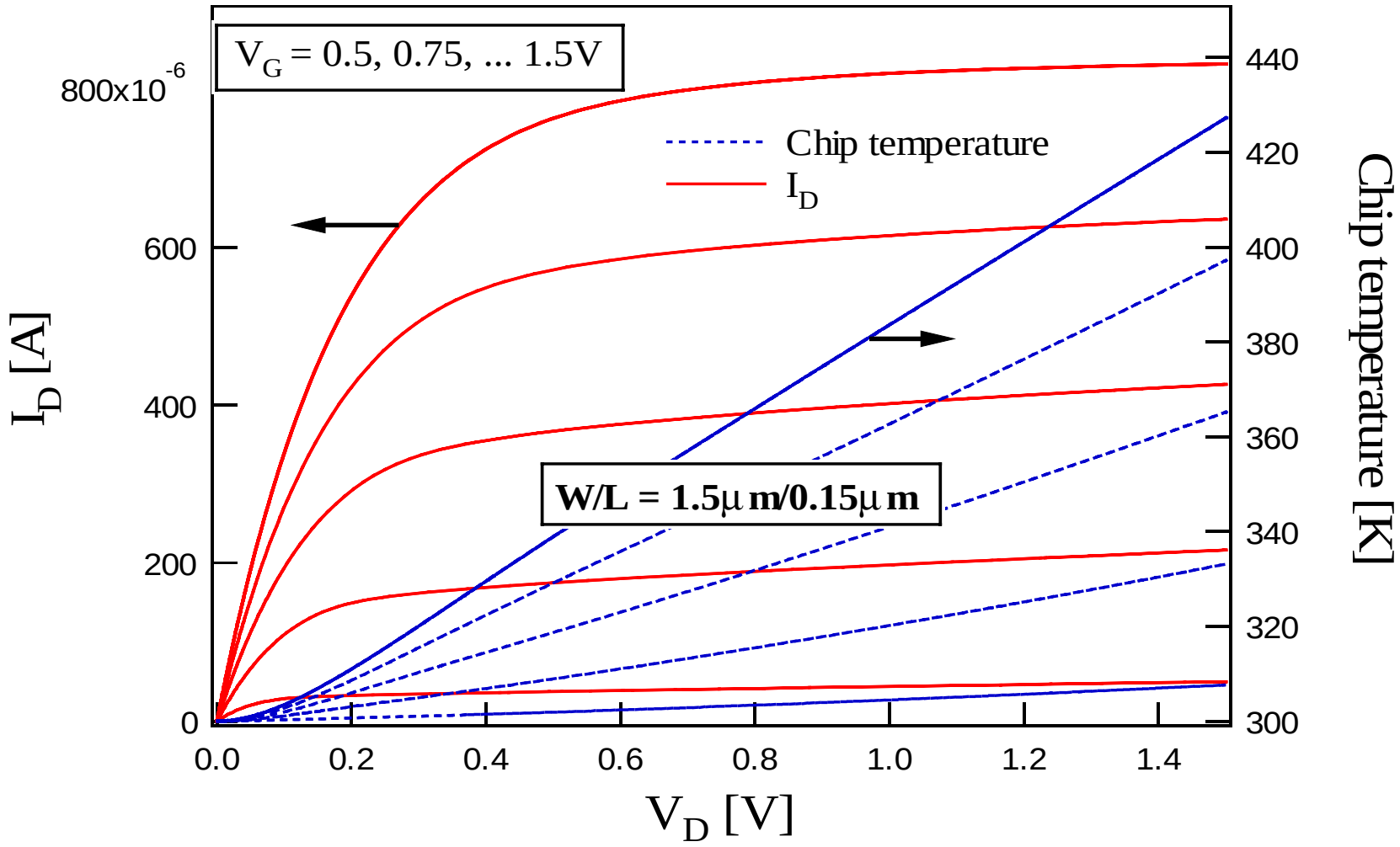
```
temp == rth * power ;
power == cth * temp'dot;
```

```
`include "disciplines.h"
module ekv_n(d,g,s,b,j);

inout d,g,s,b,j ;
electrical d,g,s,b ;
thermal j ;

analog begin // EKV v2.6 long-channel
..
  konq =boltz/charge;
  vt = konq * Temp(j) + 1.0E-20;
  ratio = Temp(j)/reftemp;
  logratio = abs(log(ratio + 1.0e-20));
  refvt = konq * reftemp;
..
  ispec = 2 * n * beta * vt * vt;
  id = ispec * (iff - ir);
  I(d) <+ id;
  isource = -id;
  ig = 0.0;
  Pwr(j) <+ abs(id * (vd-vs));
end // analog
```

Thermal-electronic effects in action: downgrading of Q_I/G_M - V_G characteristics

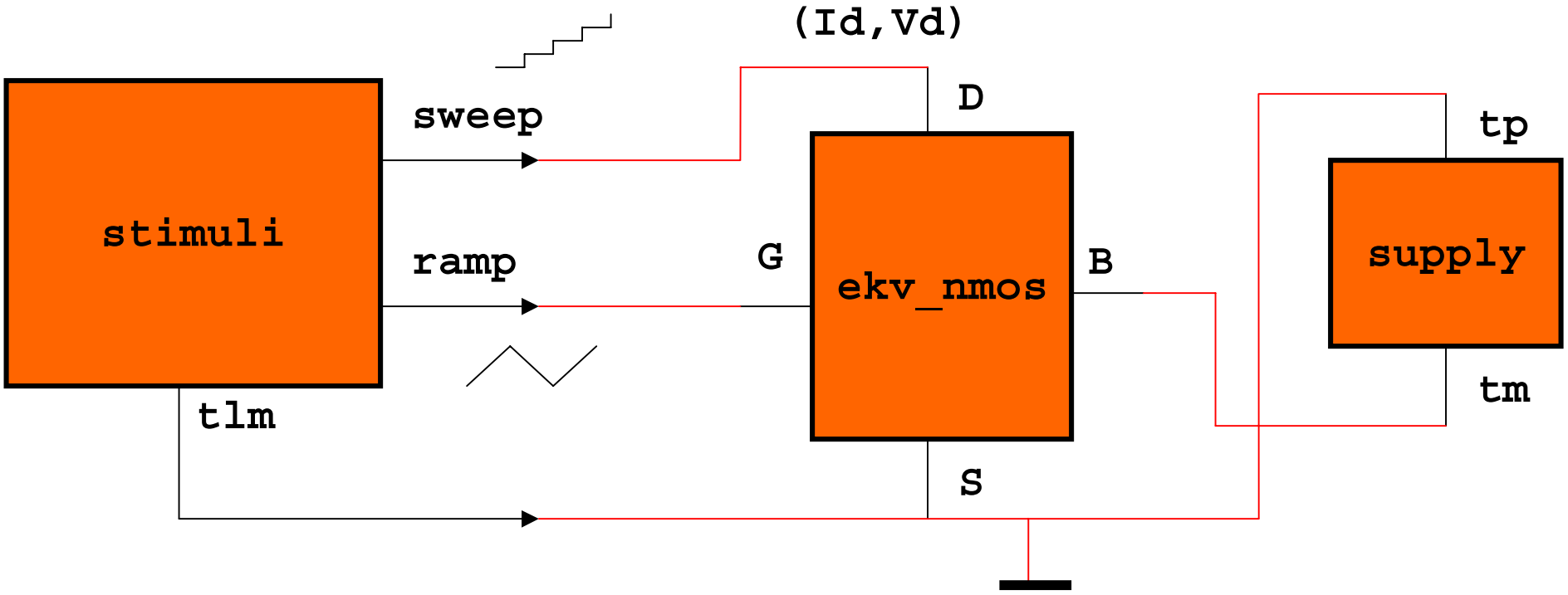


Thermal-electronic effects in action: downgrading of I_D - V_D characteristic

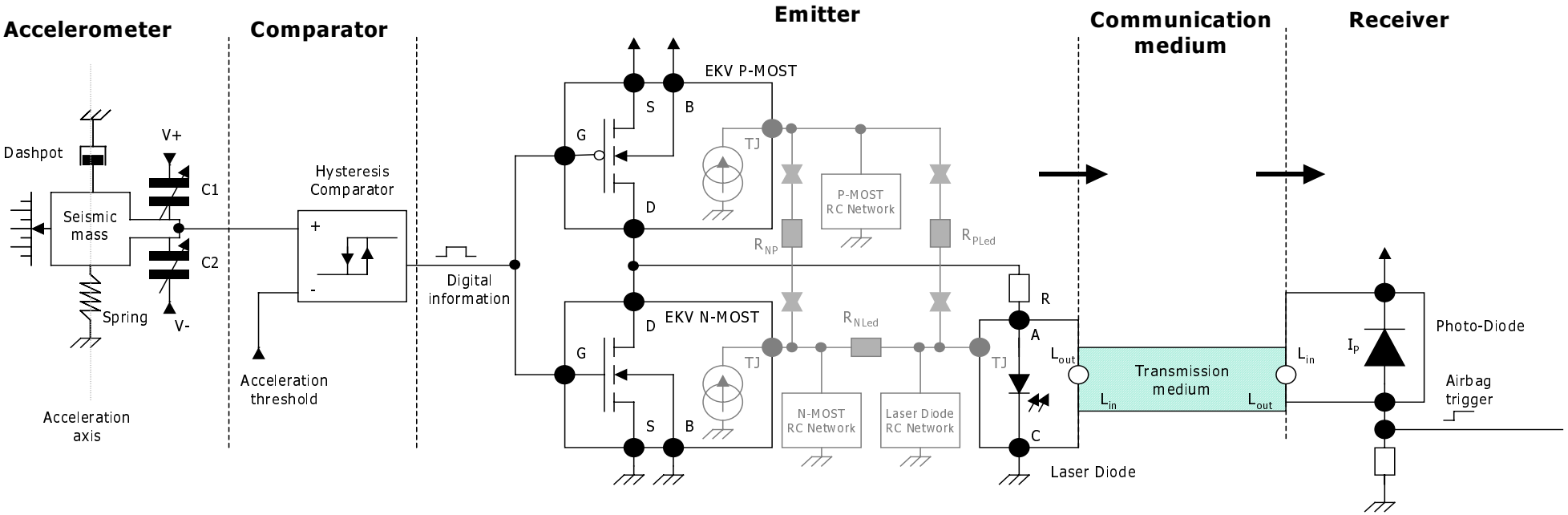


Task #7 : Instantiation, parameterization and hierarchy, Interconnecting entities

Generalized Kirchhoff laws : The ability to only detail the constitutive equations of the inner submodels (implicit in VHDL-AMS, explicit in Verilog AMS.



Ongoing researches : thermal coupling



Simulation results