Introduction to IBIS models

April 26, 2000

Arpad Muranyi
Signal Integrity Engineering
Intel Corporation
arpad.muranyi@intel.com

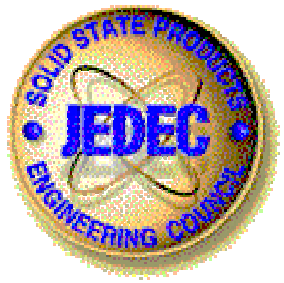
intel®

Platform Components GROUP

Sponsors


EIA
IBIS


intel


SOLID STATE PRODUCTS
JEDEC
ENGINEERING COUNCIL


SiliconTech
The Single Solution for OEMs


TRANSMETA
Corporation

Agenda

- Introduction
- Transistor vs. behavioral modeling
- I-V curves
- Ramps and V-t curves
- How to obtain C_comp
- SPICE simulations for IBIS models
- Package modeling
- Advanced I/O buffer types
- Buffer with pullup resistor example

What is IBIS?

I I/O
B buffer
I information
S specification

IBIS, common name for any of about 30 species of long-legged, long-necked wading birds.

- **IBIS is a standard for describing the analog behavior of the buffers of digital devices using plain ASCII text formatted data**
  - IBIS files are really *not* models, they just contain the data that will be used by the simulation tool's behavioral models and algorithms
- **Started in the early 90's to promote tool independent I/O models for system level Signal Integrity work**
- **It is now the ANSI/EIA-656 and IEC 62014-1 standard**

http://www.eia.org/eig/ibis/ibis.htm

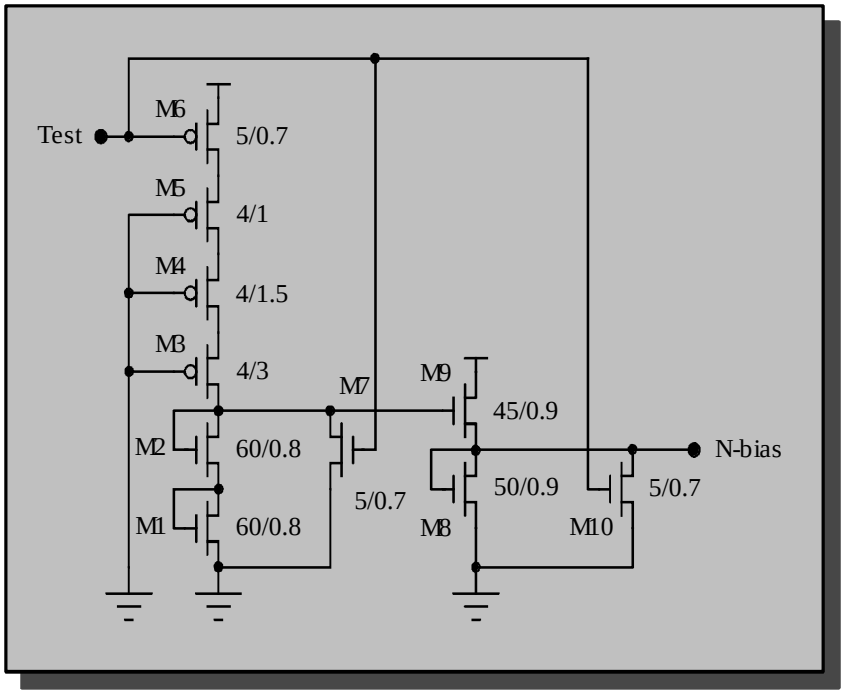
The origins of IBIS

- PCI bus Signal Integrity simulations were ramping up at Intel, but no one had a PCI buffer designed yet
- SPICE models were very difficult to get in general
- We needed an easy way to do “what if” analysis to come up with the buffer specification
- Developed a behavioral buffer model in HSPICE to be able to simulate any buffer characteristics
- The behavioral model was so successful that Intel decided to supply these to customers
- However, not all customers used HSPICE, so a tool independent model format was desirable
- Several EDA tool vendors showed interest in a common modeling format
- The IBIS Open Forum was formed and the first IBIS specification was written

Behavioral simulation background

- **Quad Design Technology**
  TLC, XTK - over 20 years old
- **Integrity Engineering**
  - SimNet - late 80's early 90's
- **Cadence**
  DFS (design for signoise) - early 90's
- **Siemens → Incases**
  Signal Integrity Workbench - early 90's
- **Intusoft**
  IsSpice SpiceMod - early 90's
- **Hyperlinx**
  Line Sim - early 90's
- **Interconnectix**
  Interconnect Synthesis - early/mid 90's

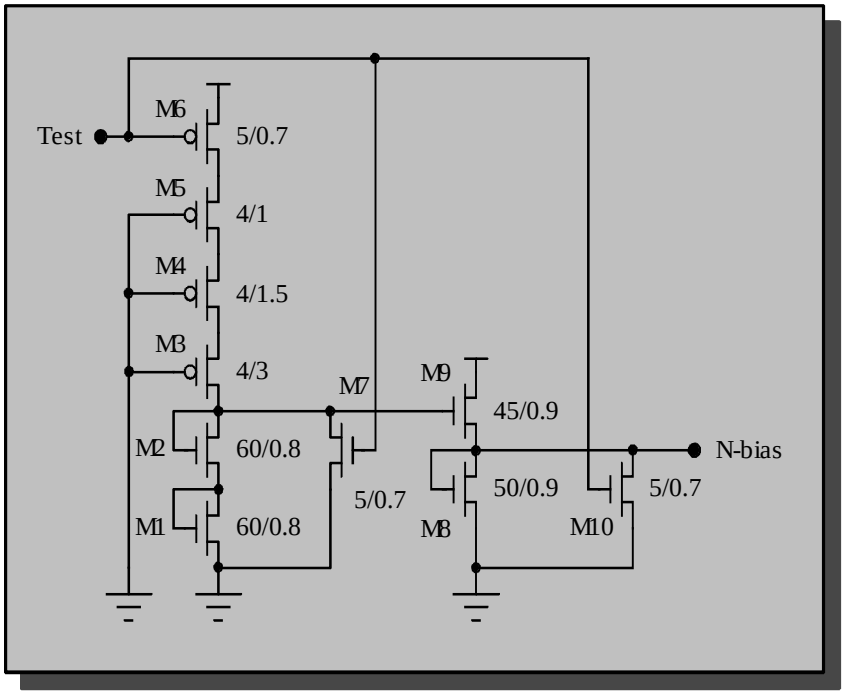
SPICE model




```
*********************************************
.MODEL NMOS NMOS
+ (LEVEL=3          UO=400.0          VTO=1.00
+   TPG=1           TOX=15E-9         NSUB=1.00E17
+   VMAX=200.0E3    RSH=50            XJ=100.0E-9
+   LD=120.0E-9     DELTA=20.0E-3     THETA=0.10
+   ETA=10.0E-3     KAPPA=20.0E-18    PB=0.40
+   CGSO=2.00E-10   CGDO=2.00E-10     CJ=0.30E-3
+   CJSW=0.20E-9    MJ=350.0E-3       MJSW=200.0E-3)
*********************************************
```

uses full schematics of the buffer
plus
manufacturing process description

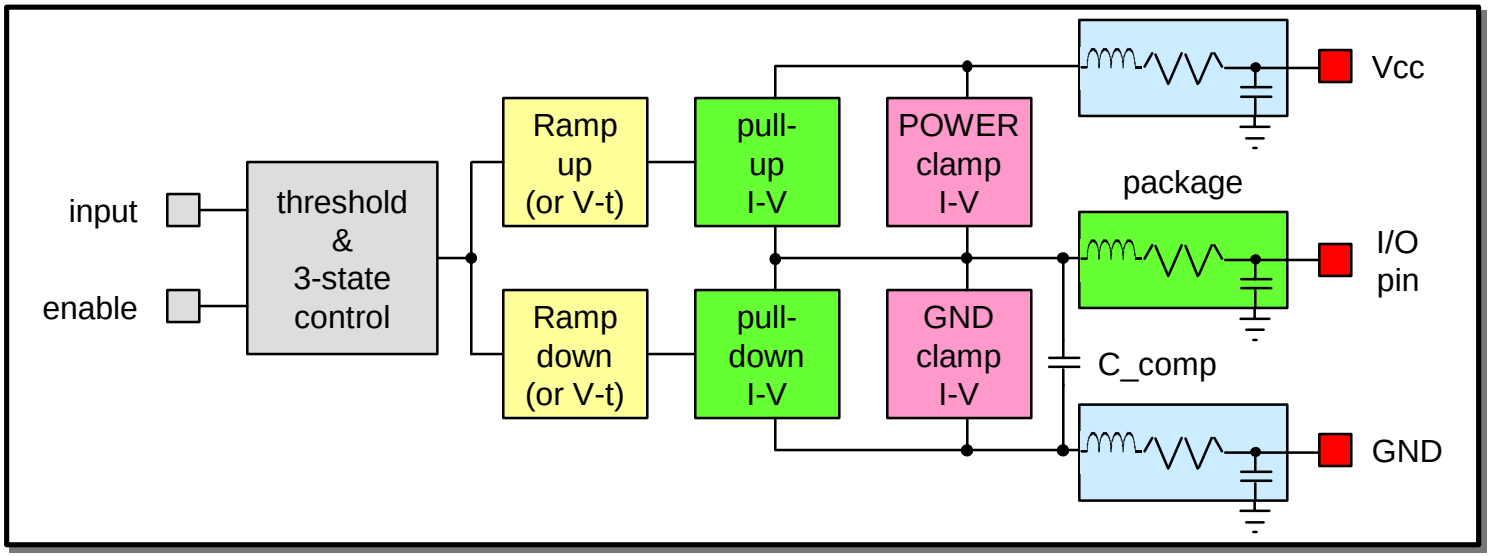
Table driven transistor model




```
****************************************
.MODEL NMOS NMOS ...
   Vgs      Ids            ...      Cgs
  -1.000   -0.2178         ...      1.230e-12
  -0.800   -6.498e-02      ...      1.250e-12
  -0.600   -2.913e-02      ...      1.270e-12
  -0.400   -1.542e-02      ...      1.290e-12
  -0.200   -7.874e-03      ...      1.310e-12
   0.000   -4.733e-05      ...      1.330e-12
   0.200    7.643e-03      ...      1.350e-12
   0.400    1.448e-02      ...      1.370e-12
   0.600    2.062e-02      ...      1.390e-12
   0.800    2.620e-02      ...      1.410e-12
   1.000    3.086e-02      ...      1.430e-12
****************************************
```

uses full schematics of the buffer
plus
I-V and C-V curves (tables) for each transistor

IBIS model



Block diagram of CMOS buffer

A basic IBIS model consists of:

four I-V curves:	- pullup & POWER clamp
	- pulldown & GND clamp
two ramps:	- dV/dt_rise
	- dV/dt_fall
die capacitance:	- C_comp
packaging:	- RLC values

for each buffer on a chip

What are these models based on?

- **SPICE (transistor) model**
  - voltage/current/capacitance relationships of device nodes are calculated with detailed equations using device geometry, and properties of materials
  - measured data is curve fitted for the equations
- **Table driven transistor model**
  - voltage/current/capacitance relationships of device nodes are based on I/C/V lookup tables
  - data tables are generated from full SPICE model simulations or die level measurements
  - data may be curve fit to equations for more efficiency and flexibility
- **IBIS (or behavioral) model**
  - current/voltage/time relationships of entire buffer (or building block) are based on lookup tables (I/V and V/t curves)
  - data tables are generated from full SPICE model simulations or external measurements
  - data may be curve fit to equations for more efficiency and flexibility

Model characteristics

- **SPICE (transistor) model**
  - simulates very slowly, because voltage/current relationships are calculated from lower level data
  - voltages/currents are calculated for each circuit element in the buffer
  - best for circuit designers
  - too slow for system level interconnect design
  - reveals process and circuit Intellectual Property
- **Table driven transistor model**
  - simulates faster, because voltage/current relationships are given directly
  - voltages/currents are calculated for each circuit element in the buffer
  - more suitable for system level design because it is faster
  - hides process, but still reveals circuit Intellectual Property
- **IBIS (or behavioral) model**
  - simulates fastest, because voltage/current/time relationships are given only for the external nodes of the entire buffer (or building block)
  - no circuit detail involved
  - useless for circuit designers
  - ideal for system level interconnect design
  - hides both process and circuit intellectual property

Why is the simulation speed different?

- **SPICE models are not slower because they are based on equations**
  - SPICE models are built on lower level information, such as physical dimensions, properties of material, full circuit netlist, etc.
- **Table driven (or behavioral) models are not faster because they are table driven**
  - lookup tables require a lot of conditional evaluations which may be slower than expression evaluations
  - processing time depends on the length of the data tables also
  - multi-dimensional tables tend to get huge and difficult to handle
  - expressions could make the information more efficient
  - expressions could make behavioral modeling more general (no new syntax would be required for each new behavior)
  - table driven models are faster because they use a higher level device description

What is the real difference?

- **The real difference is in the level of abstraction**
  - molecular/atomic description of materials (crystal structure)
  - electron behavior in crystal structure (mobility, etc.)
  - geometric device description (width, length, thickness)
  - properties of materials (doping, resistance, etc.)
  - current/capacitance/voltage relationships on a device level
  - current/capacitance/voltage/time relationships on a buffer level
- **There is really no such thing as behavioral or transistor level model!**
  - SPICE models could be re-written to be table driven from raw measurement data (don't even think of it!)
  - table driven models could be re-written to use curve fitted equations instead (may be a good thing to consider)

All models are "behavioral" on some level!

What is IBIS?

I **I/O**
B **buffer**
I **information**
S **specification**

IBIS, common name for any of about 30 species of long-legged, long-necked wading birds.

- **IBIS is a standard for describing the analog behavior of the buffers of digital devices using plain ASCII text formatted data**
  - IBIS files are really *not* models, they just contain the data that will be used by the simulation tool's behavioral models and algorithms
- **Started in the early 90's to promote tool independent I/O models for system level Signal Integrity work**
- **It is now the ANSI/EIA-656 and IEC 62014-1 standard**

http://www.eia.org/eig/ibis/ibis.htm

intel®

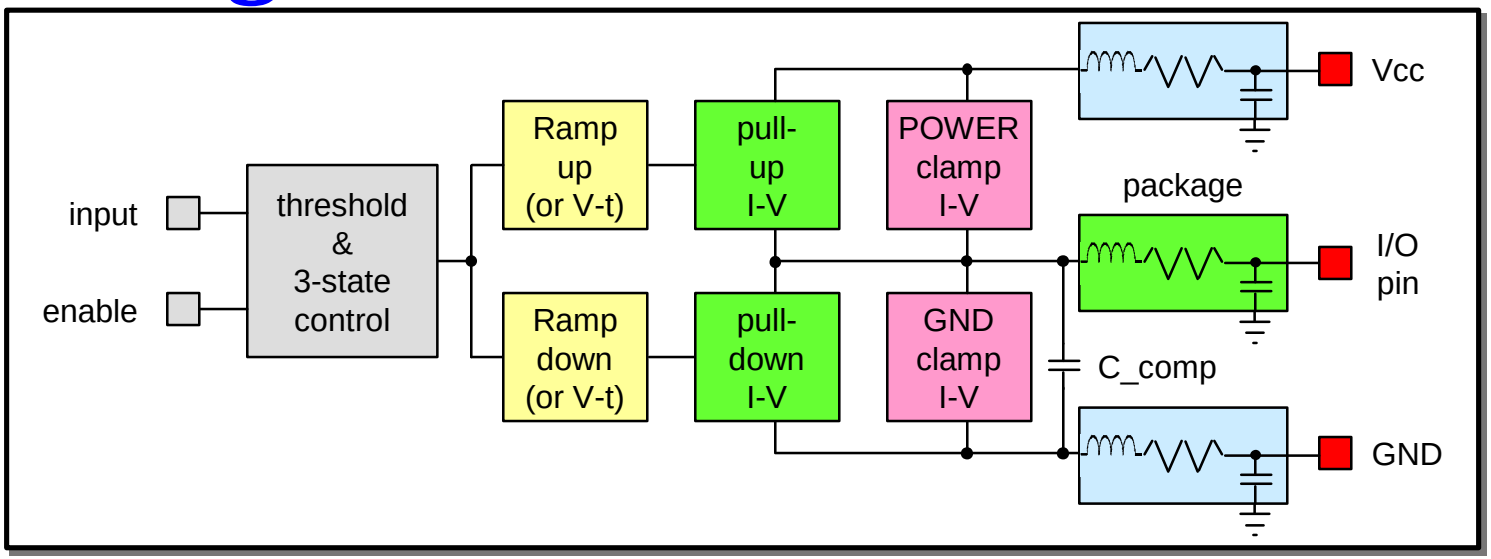
Platform Components GROUP

Basic structure of an IBIS file

component

- **Header**
  - file name, date, version, source, notes, disclaimer, copyright, etc.
  - default package data (L_pkg, R_pkg, C_pkg)
  - pin list (pin name, signal name, buffer name, and optional L_pkg, R_pkg, C_pkg)
  - advanced items (differential pin associations, buffer selector, etc.)
- **Model data**
  - all models found on the chip are included in this section
  - each flavor of a programmable buffer is described under a separate [Model] section
- **One IBIS file can describe several components (chips)**
  - a new [Component] keyword can be used for each part within the same .IBS file
  - models from multiple vendors can be combined to form a “system model”

Building blocks of a basic IBIS model



Block diagram of CMOS buffer

A basic IBIS model consists of:

four I-V curves:	- pullup & POWER clamp
	- pulldown & GND clamp
two ramps:	- dV/dt_rise
	- dV/dt_fall
die capacitance:	- C_comp
packaging:	- RLC values

for each buffer on a chip

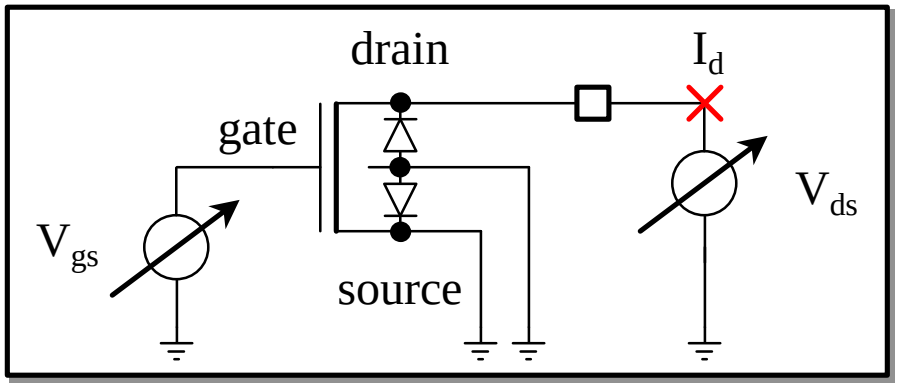
Advanced / new features in IBIS

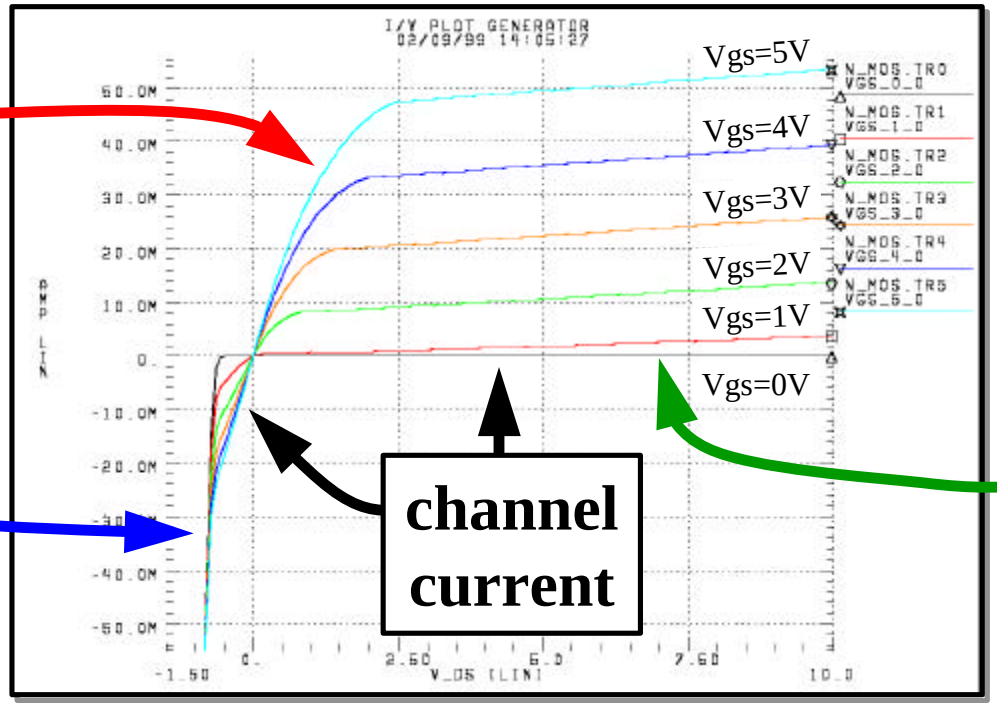
- **Multi-section uncoupled package description**
  - transmission line, package stubs
- **Single-section (lumped) full matrix coupled package description**
- **Electrical board description (EBD)**
- **Multi-stage buffer [Driver Schedule]**
- **Dynamic clamping and bus hold capabilities (on-die termination)**
- **Series pin to pin and FET bus switch modeling**
- **Model selector for programmable buffers**
- **Extended model specifications and simulation hooks**
  - ringback and hysteresis specifications

IBIS model as buffer specification

- **Signal Integrity group defines the correct buffer characteristics**
  - behavioral models lend themselves to easy “what if” analysis to find a system level solution space
  - results can be communicated to design engineering through the I-V and V-t curves and other parameters
- **Circuit designers can use these requirements as a specification for the buffer to be designed**
  - matching SPICE simulated I-V and V-t curves to IBIS model
- **Intel has used this methodology since the early 90’s on several bus interfaces**
  - PCI specification
  - AGP specification
  - 100 MHz SDRAM (PC100) specification
  - numerous internal projects

I-V curves of N-channel pulldowns





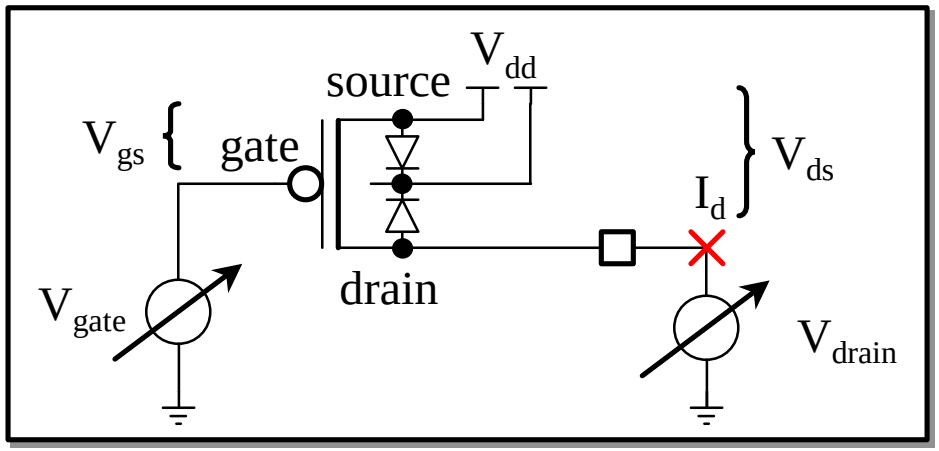
Fully ON

diode current

channel current

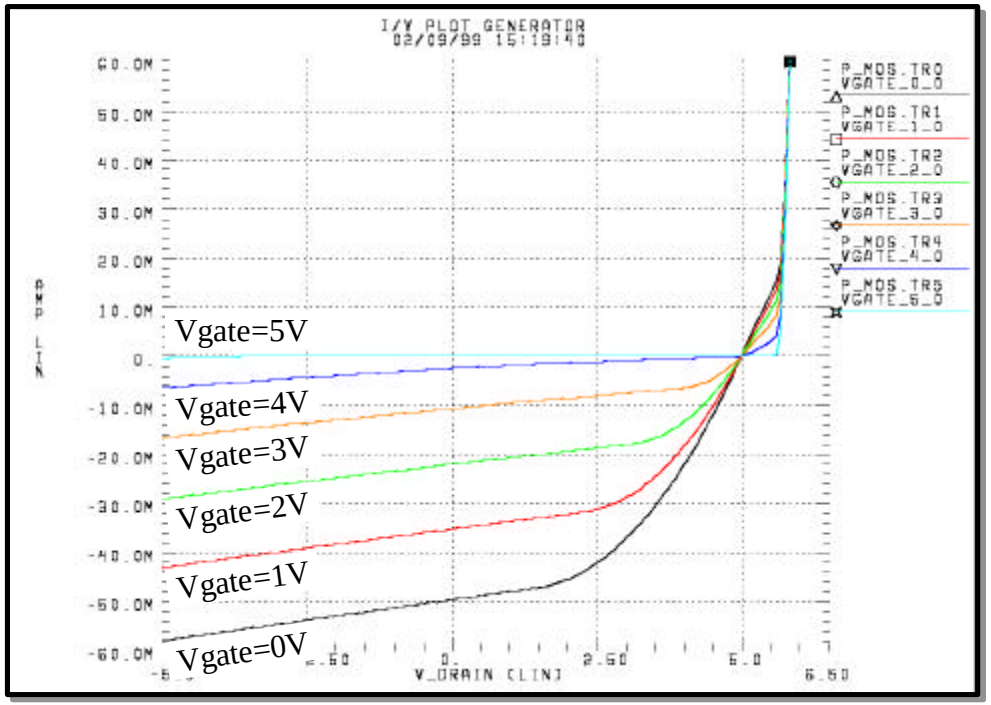
OFF

I-V curves of P-channel pullups



$$V_{gs}=V_{dd}-V_{gate}$$
$$V_{ds}=V_{dd}-V_{drain}$$

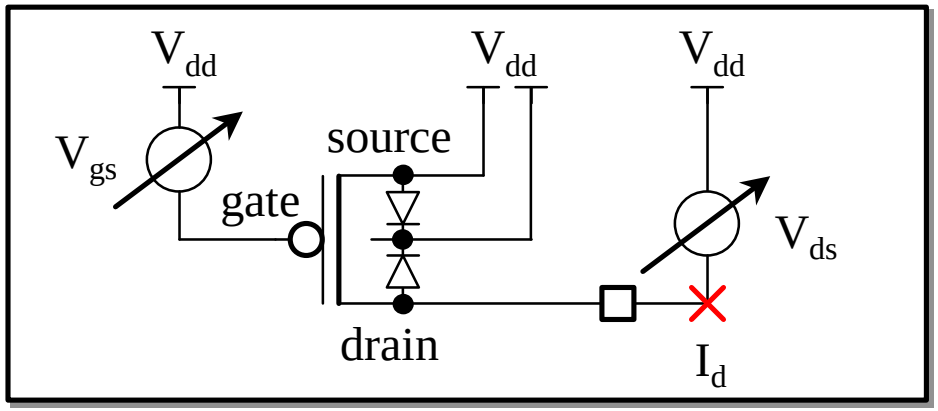
Ground relative measurements don't make sense for pullup structures because V_{gs} and V_{ds} are not related to the GND potential!

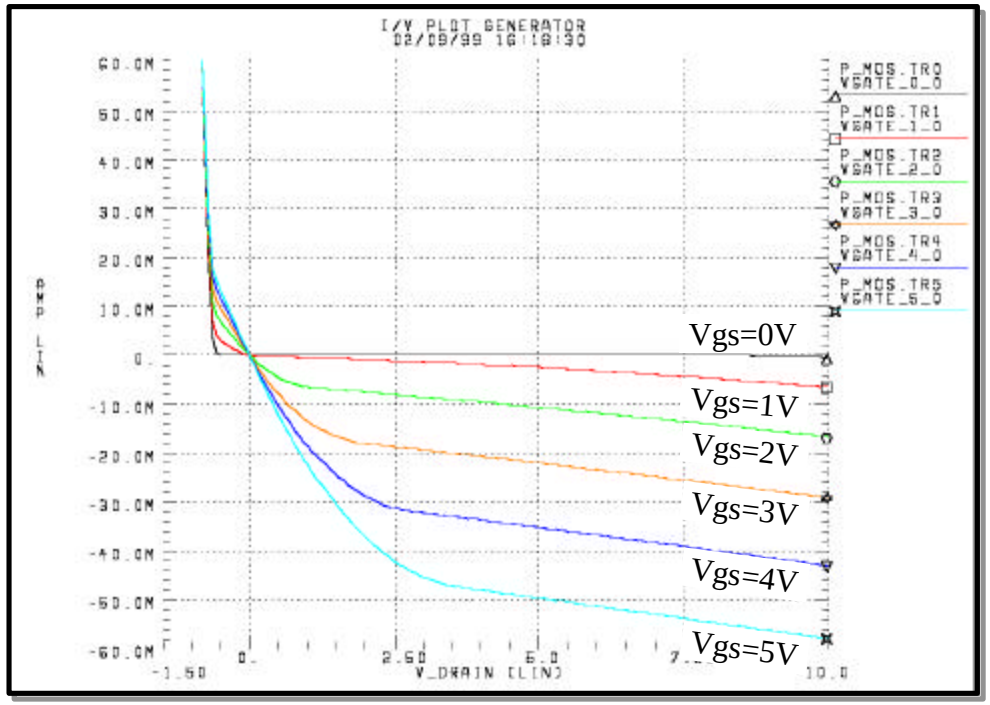


(V_{dd} varies with minimum, typical, and maximum modeling conditions, as well as in GND bounce and V_{dd} droop simulations).

Platform Components GROUP

V_{dd} relative pullup I-V curves

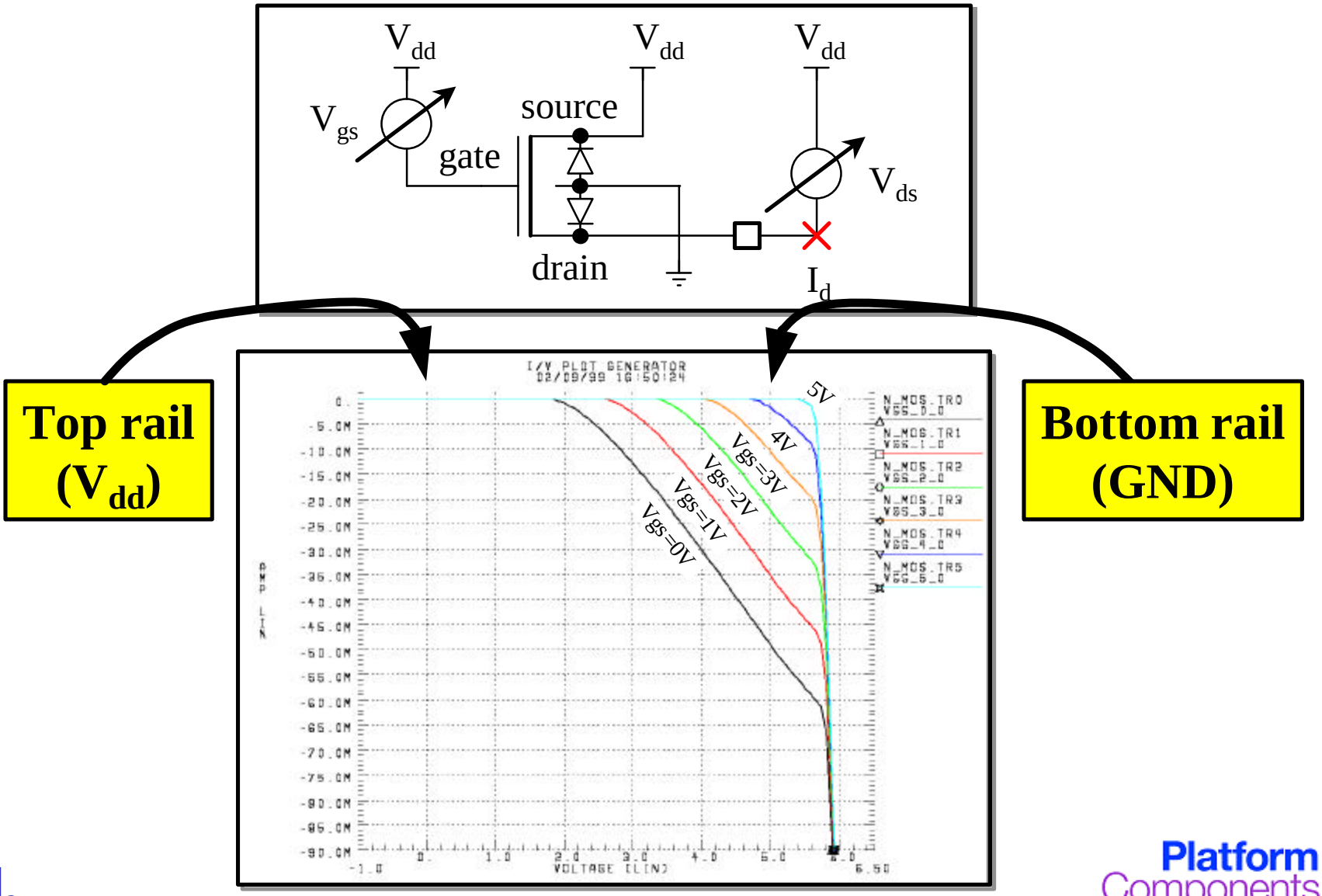




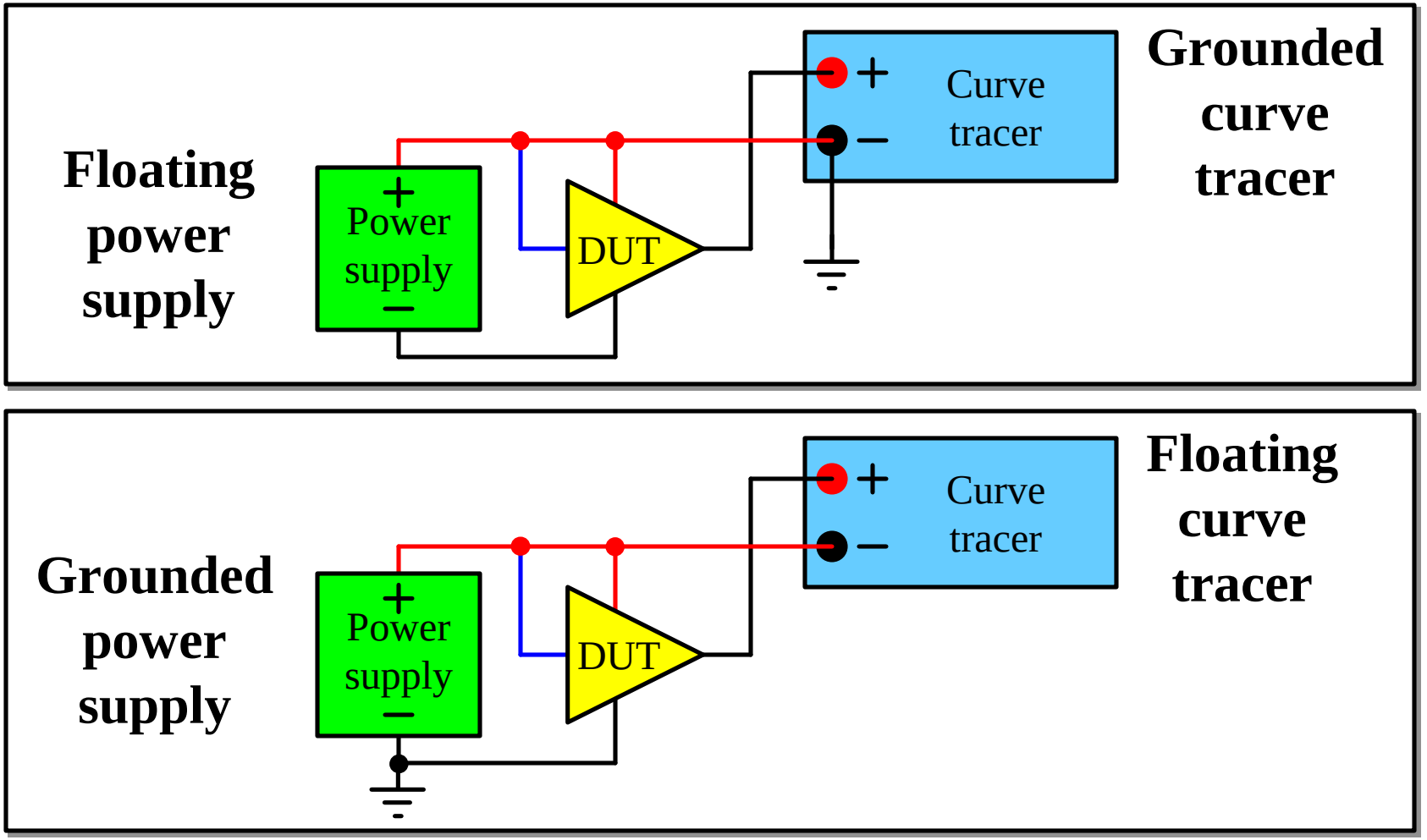
V_{dd} relative characterization of pullup structures are perfectly symmetric with the ground relative characterization of pulldown structures

Measured (or DUT) voltage conditions do not have to be recalculated for each supply voltage.

N-channel pullup I-V curves



V_{dd} relative lab measurements



Don’t do this if supply and curve tracer are both grounded!

What do we get in the I-V curves?

Digital buffers can only be measured in two modes (if not one)

- **Receive or 3-state mode**
  - **allows us to measure the currents of**
    - parasitic diodes (for MOSFETS)
    - ESD protection circuits
    - pullup or pulldown “resistors”
    - static overshoot protection clamps
    - integrated static terminators
- **Drive mode (driving high or low)**
  - **allows us to measure the currents of all of the above, *plus***
    - channel currents for pulldown and/or pullup structures
    - dynamic clamps
    - dynamic bus hold circuits
    - integrated active terminators

Both modes are needed for simulations

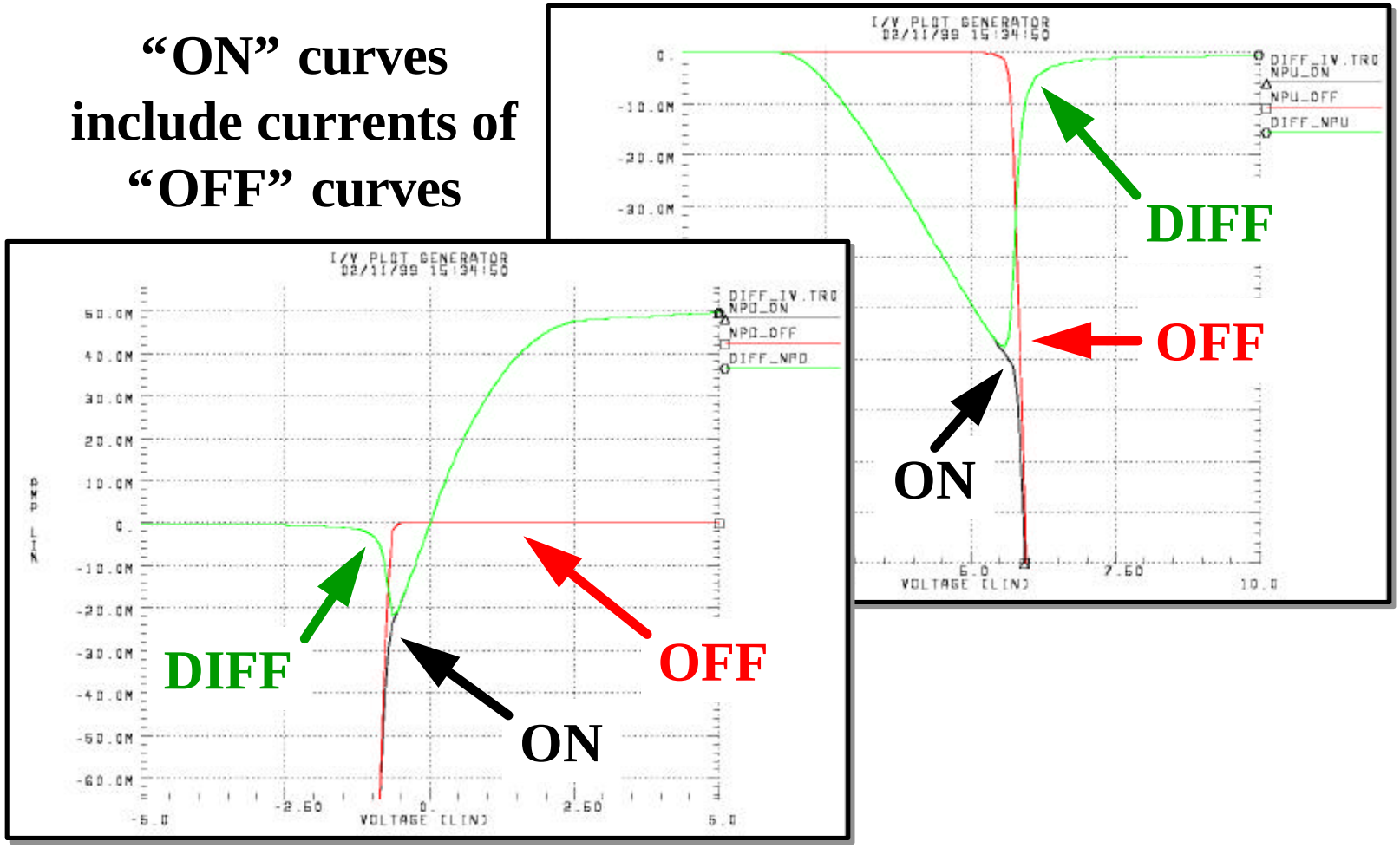
Drive/receive mode transitioning

- **Phase out "receiver model" as the "driver model" is phased in, or vice versa**
  - It may be difficult to come up with an algorithm that can cross fade the two models so that the static currents remain constant during the transitioning process.
- **Keep "receiver model" constantly in the circuit and phase in/out the *difference* between the driver and receiver models**
  - Note that a drive mode model includes the static currents of those circuit or device elements that make up a receive mode model because these are never turned off. These currents would be doubled if the drive mode model would simply be added to a receive mode model.

The second approach is used in IBIS

Difference I-V curve examples

“ON” curves include currents of “OFF” curves



The four I-V curves of an IBIS model

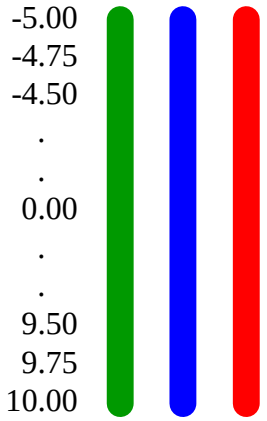
- **[Pulldown], referenced to [Pulldown Reference]**
  - contains the difference of drive and receive (3-state) mode I-V curves for driver driving low
  - the origin of the curve is usually at 0 V (GND) for normal CMOS buffers, but could be at -12 V for RS232 drivers, for example
- **[Pullup], referenced to [Pullup Reference]**
  - contains the difference of drive and receive (3-state) mode I-V curves for driver driving high
  - the origin of the curve is usually at the supply voltage (Vcc or Vdd)
- **[GND Clamp], referenced to [GND Clamp Reference]**
  - contains the receive (3-state) mode I-V curves
  - the origin of the curve is usually at 0 V (GND) for normal CMOS buffers, but could be at -12 V for RS232 drivers, for example
- **[POWER Clamp], referenced to [POWER Clamp Reference]**
  - contains the receive (3-state) mode I-V curves
  - the origin of the curve is usually at the supply voltage (V_{cc} or V_{dd}), but
  - for 5 V safe 3.3 V buffers, for example, this would be referenced to 5 V while the pullup is referenced to 3.3 V

I-V curve ranges

- **The general rule is $-V_{dd}$ to $2*V_{dd}$**
  - for a 5 V buffer this means -5 V to +10 V
- **Why is this necessary?**
  - the theoretical maximum overshoot due to a full reflection is twice the signal swing
  - not all tool extrapolate curves the same way, it is better to define it
- **Exception: clamp curves**
  - [GND Clamp] is required to have $-V_{dd}$ to V_{dd} (-5 V to +5 V)
  - [POWER Clamp] is required to have V_{dd} to $2*V_{dd}$ (+5 V to +10 V)
- **Reason: prevent double counting**
  - the two clamp curve measurements contain the same information
  - the only difference is that one of them is measured with respect to [GND Clamp Reference] (GND relative) and the other one with respect to [POWER Clamp Reference] (V_{cc} or V_{dd} relative)
- **Note that these ranges are defined as minimum required ranges in the IBIS specification, but providing data over a wider range is not prohibited**

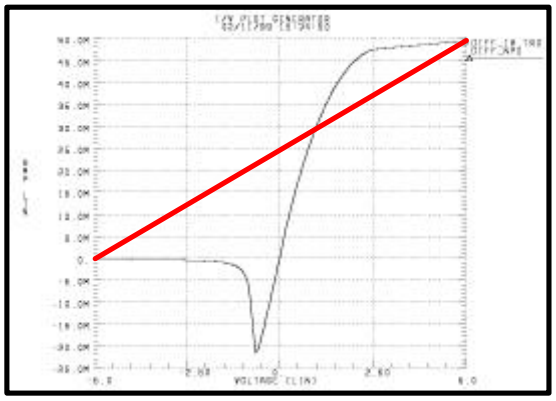
More on I-V curve ranges

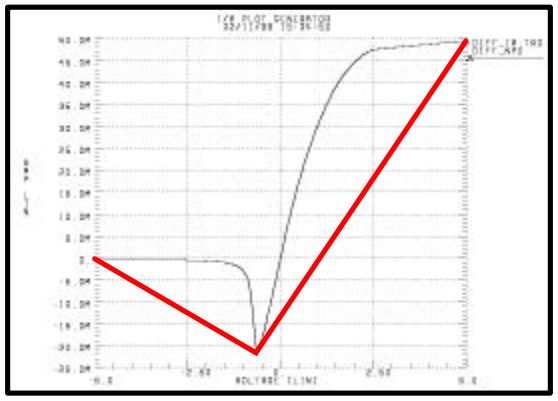
- **Mixed voltage cases**
  - a 5 V safe 3.3 volt buffer should use a -5 V to +10 V range because there is 5 V signaling
  - a 3.3 V GTL+ driver using a V_{ref} of 1.5 V should have a range of -1.5 V to 3.0 V because the signaling is 1.5 V (at most)
- **What about typ., min., and max. supply voltages?**
  - the voltage range should be based on the typical value only because there is one voltage column for each case, and a numerical value is required in the first and last points

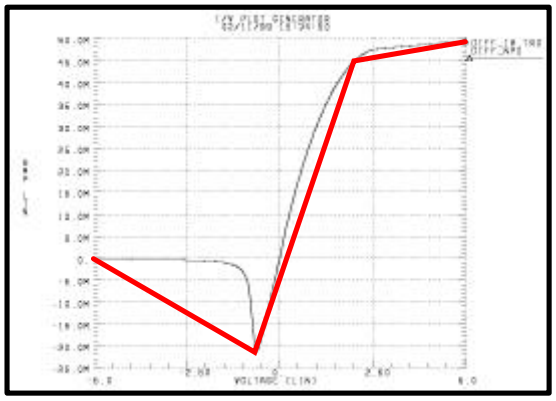


Best 100 points reduction

- **Higher resolution yields more accurate models**
  - it is desirable to have a point at least at every 100 mV, or less
- **The range of a 3.3 V device is -3.3 to 6.6 V, a span of 9.9 V**
  - at 100 mV spacing this will result in 100 points, exactly
- **For 5 V devices or higher resolution a “best points” algorithm must be used to eliminate points from the data where the curve is (almost) straight**
- **V-t curves may even have more points depending on the edge rate and time step used**







Temperature and power supply conditions for I-V and V-t curves

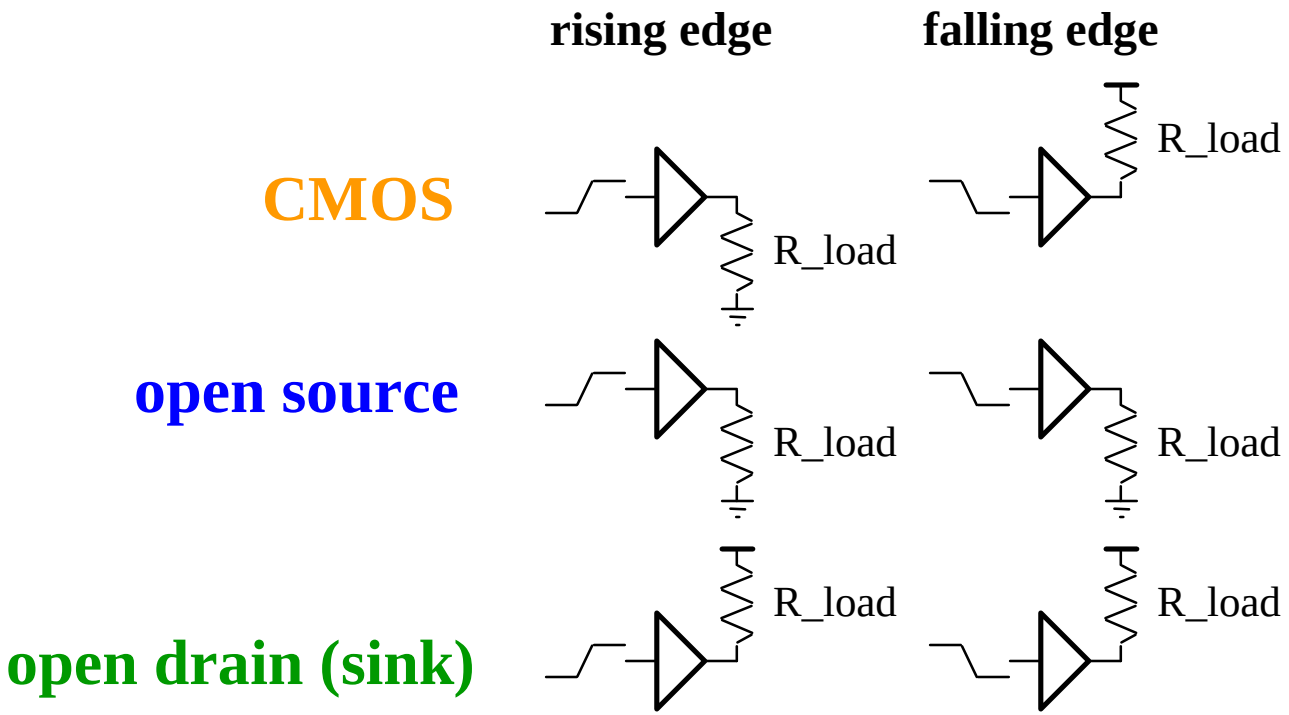
- **IBIS supports three conditions for buffer models**
  - typical values (required)
  - minimum values (optional) - not the same as worst/slow case in general
  - maximum values (optional) - not the same as best/fast case in general
- **For CMOS I-V curves**
  - minimum I-V curve conditions (worst case) require high temperature and low supply voltage
  - maximum I-V curve conditions (best case) require low temperature and high supply voltage
- **For bipolar I-V curves**
  - minimum I-V curve conditions (worst case) require low temperature and low supply voltage
  - maximum I-V curve conditions (best case) require high temperature and high supply voltage

Ramp and V-t curve measurements

- **The [Ramp] and [*** Waveform] keywords describe the transient characteristics of a buffer**
  - these keywords contain information on how fast the pullup and pulldown transistors turn on/off with respect to time
- **The effects of the package must be eliminated from these measurements**
  - remove package from SPICE circuit
  - obtain measurements from the die-pad (possibly without the bond wire connected)
  - reverse engineer a package-less waveform from packaged waveform measurement through numerical methods
- **The effects of die capacitance (C_comp) are included in the shape of the V-t curves**
  - parasitic capacitance cannot be separated from the circuit elements
  - simulation tool should have a special algorithm to avoid double counting C_comp, i.e. make it invisible from the inside out, but visible from the outside in

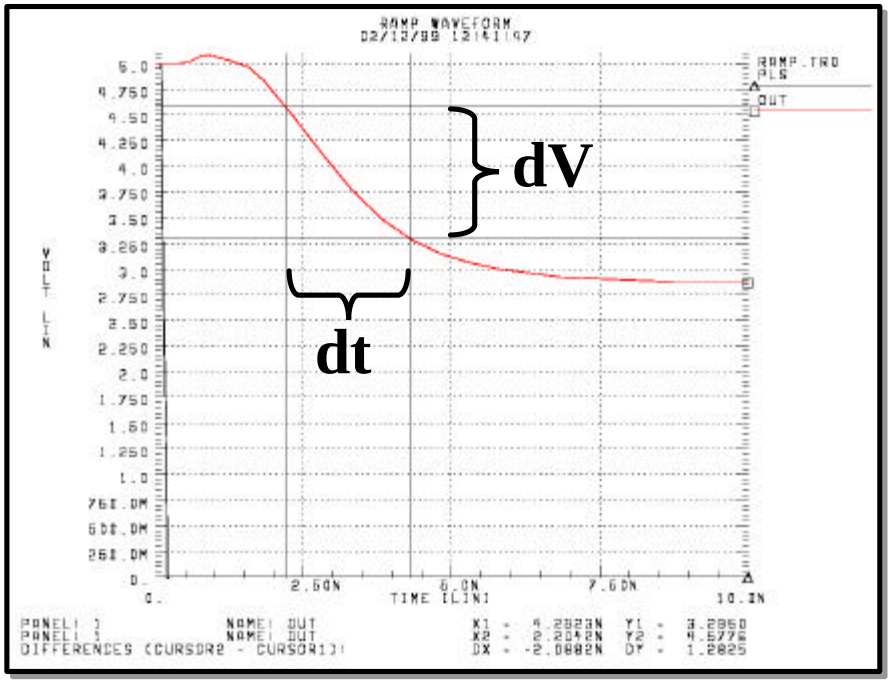
Ramp measurement setup for IBIS 1.1

Make sure R-load is connected to the appropriate supply!



Getting dV and dt from a waveform

- **Find the 20 - 80 % points of the signal swing**
- **Read the voltage difference between these points**
  - this is your dV value
- **Read the time difference between these points**
  - this is your dt value
- **Do NOT divide out these numbers!**



A note on ~0 pF loads

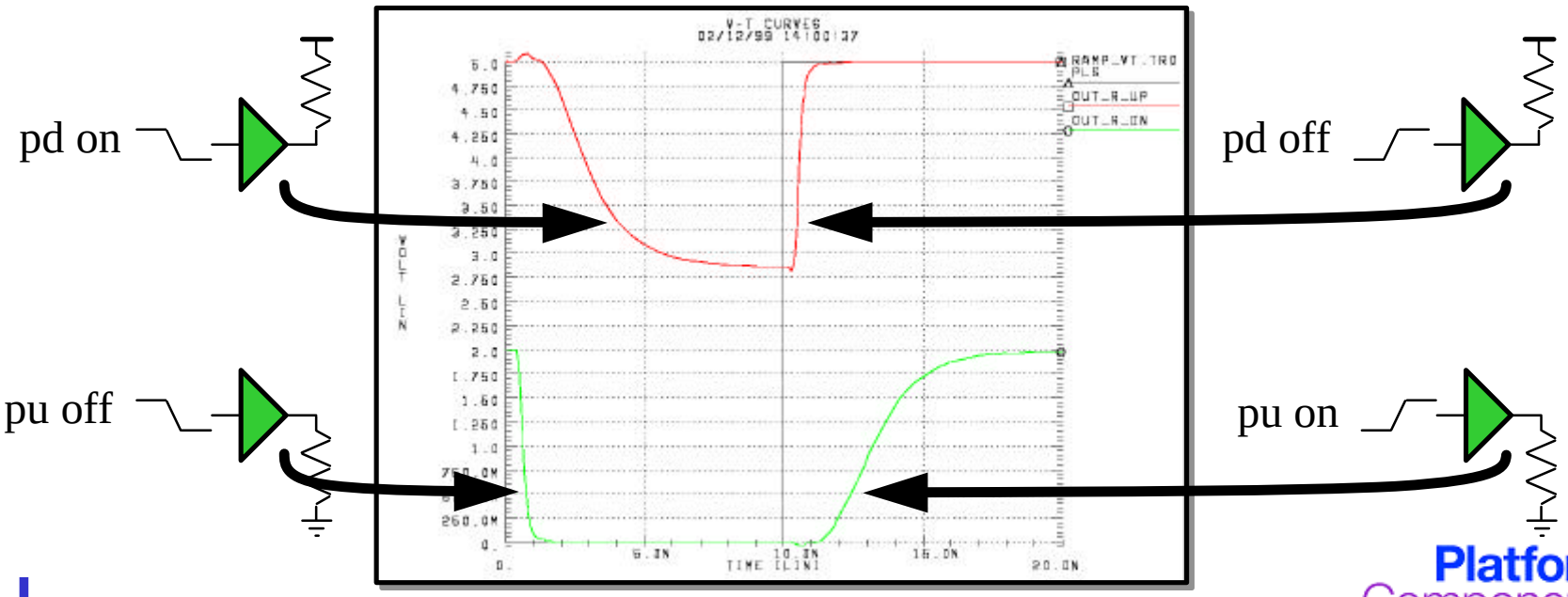
A capacitive load in buffer characterization combines the effects of speed and strength

We need a 0 pF test load!

- **For fast edges transmission lines do not behave as parallel plate capacitors**
  - the driver does not see a lumped RLGC
  - a perfect (distributed) model would have infinite lumps (n=∞)
  - the RLGC values of one lump is the total RLGC divided by n
  - as n goes to infinity, the RLGC values go to zero
- **A transmission line looks like a resistor to the buffer until the reflections return**
- **Specifying the edge rates into ~0 pF load can be achieved with a transmission line or resistor**

The V-t curves of IBIS 2.1

- **V-t curves describe the transient characteristics of a buffer more accurately than ramps**
  - a waveform holds more detail than a straight line slope
  - slew rate controlled and multi-stage buffers may have edges which do not look like straight lines (staggered turn-on/off)
- **A minimum of four V-t curves are needed to adequately describe a CMOS buffer**



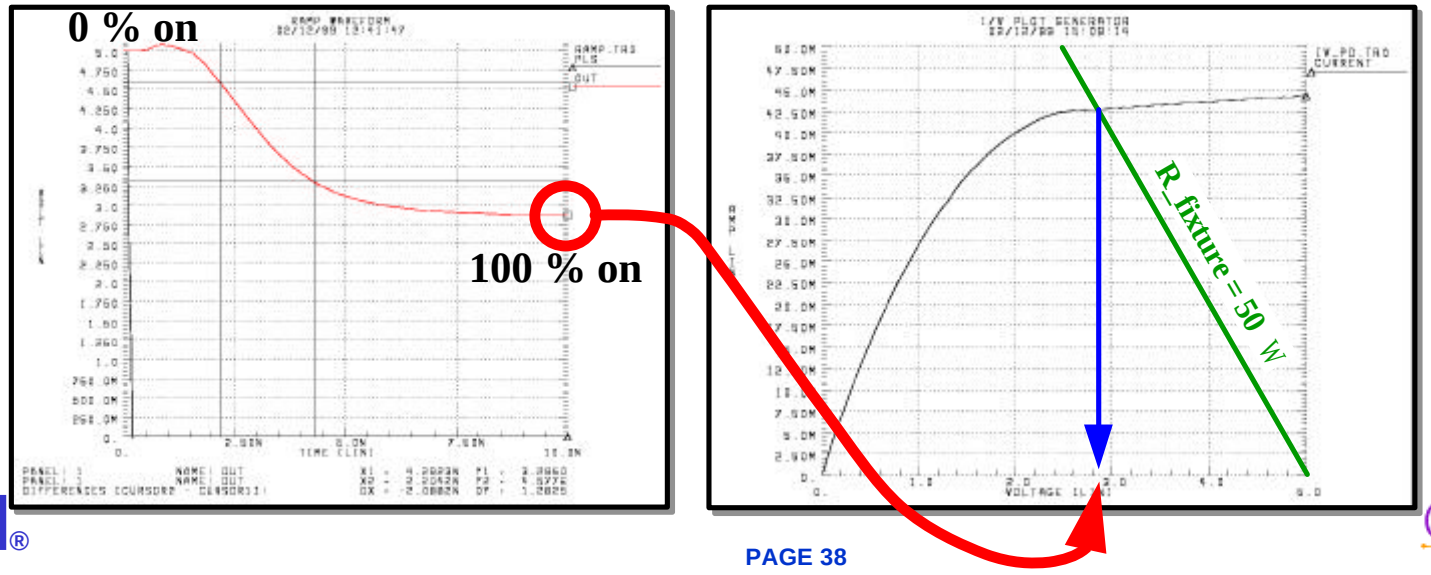
Platform Components GROUP

V-t curves must be time correlated!

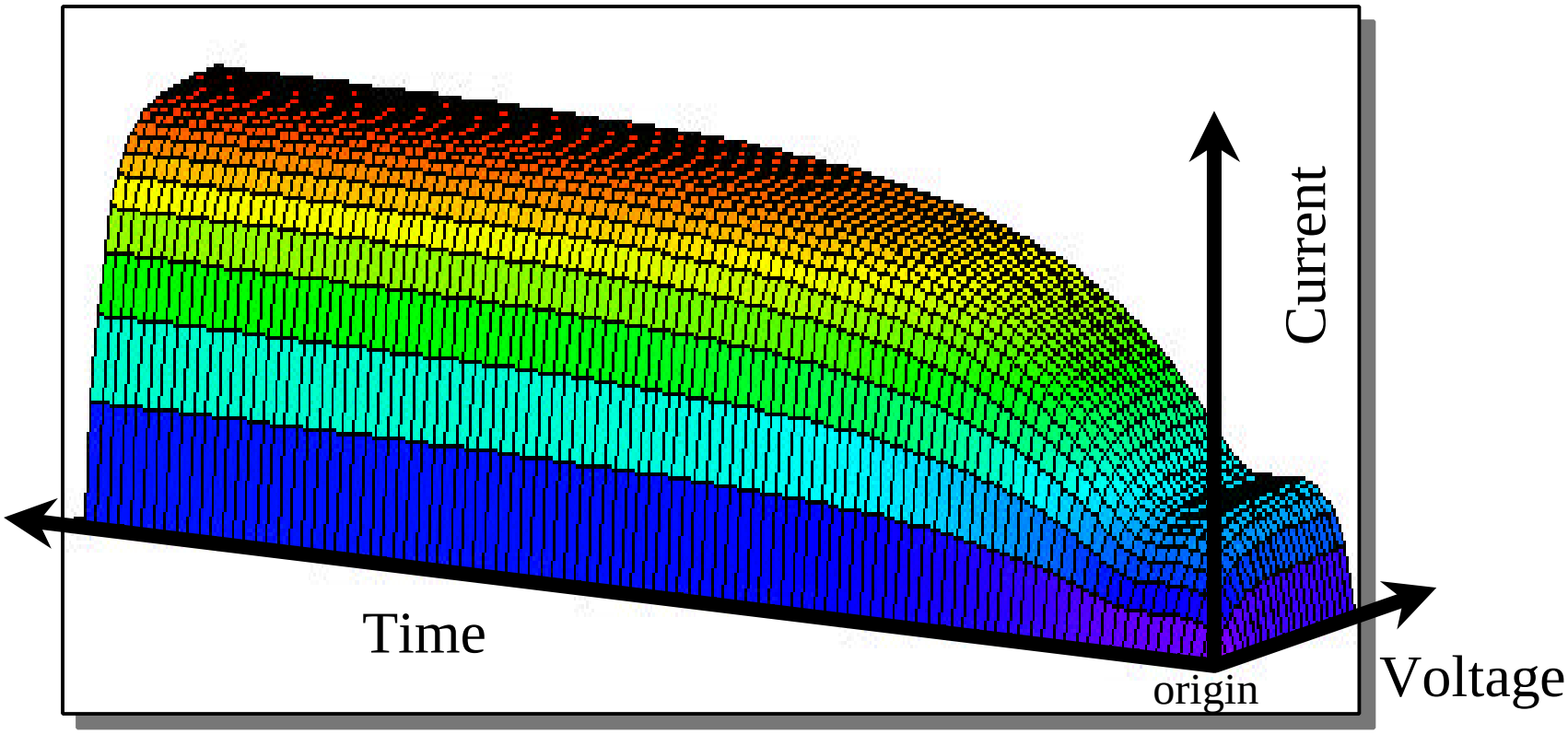
- **Each V-t curve must have t=0 where the pulse crosses the input threshold**
  - if the simulation pulse is not at t=0 of the simulation, the V-t curve(s) must be normalized
  - using a fast edge for the pulse helps to reduce threshold uncertainties
- **V-t curves incorporate the clock-to-out delay**
  - the accuracy of t_{co} depends on the accuracy of the SPICE model (don't trust it, most often they are not made for this)
  - you may need to tweak the leading horizontal part of the V-t curve(s) if more accurate t_{co} is known, BUT be aware that the time relationship between V-t curves must be kept right
- **The length of the V-t curve should correspond to the clock speed at which the buffer is used**
  - you may not get any output if the clock period is shorter than the length of the V-t curve (over clocking)

V-t curves describe the transients

- **I-V curves provide a fully on DC characterization of the transistor**
- **V-t curves can be used to scale the I-V curves**
  - the first point of a “turn-on” V-t curve represents a driver turned OFF (0 %)
  - the last point of a “turn-on” V-t curve represents a fully ON driver (100 %)
  - anything between the end points represent a partially turned ON transistor
- **Model quality check:**
  - solving for an operating point using R_fixture across the I-V curve must yield the same voltage as the last point of the corresponding V-t curve!



3D representation of a transition



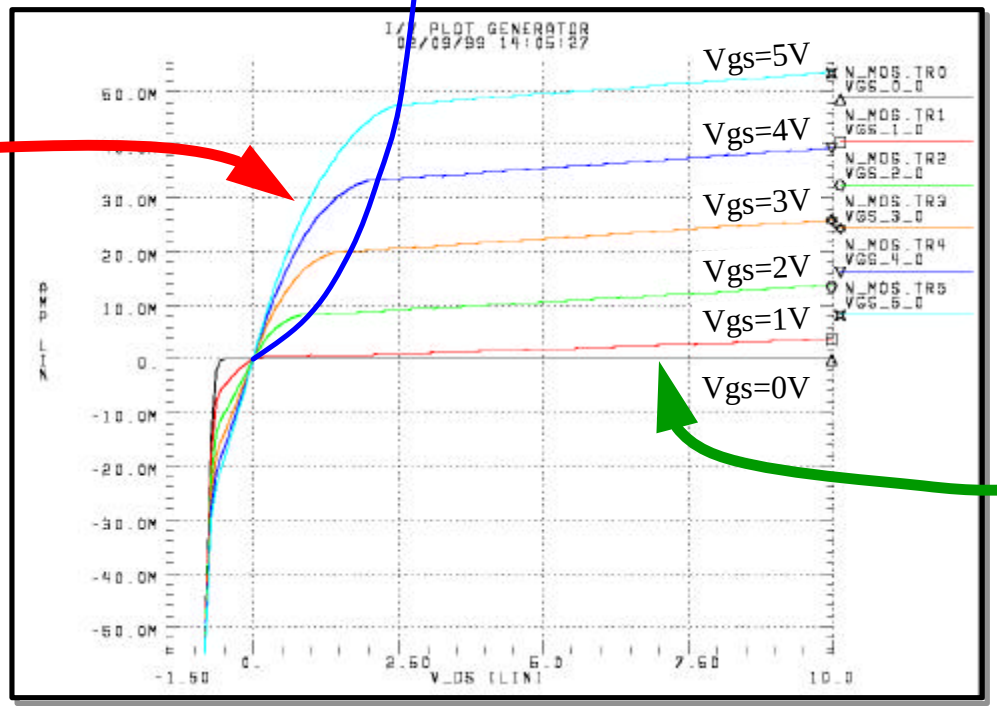
Two-stage, slew rate controlled I/O buffer

I-V curve scaling - an accuracy detail

Linear (vertical) scaling of I-V curves would keep $V_{pinchoff}$ at the same voltage. Is that a problem?

$V_{pinchoff}$ follows "square law"

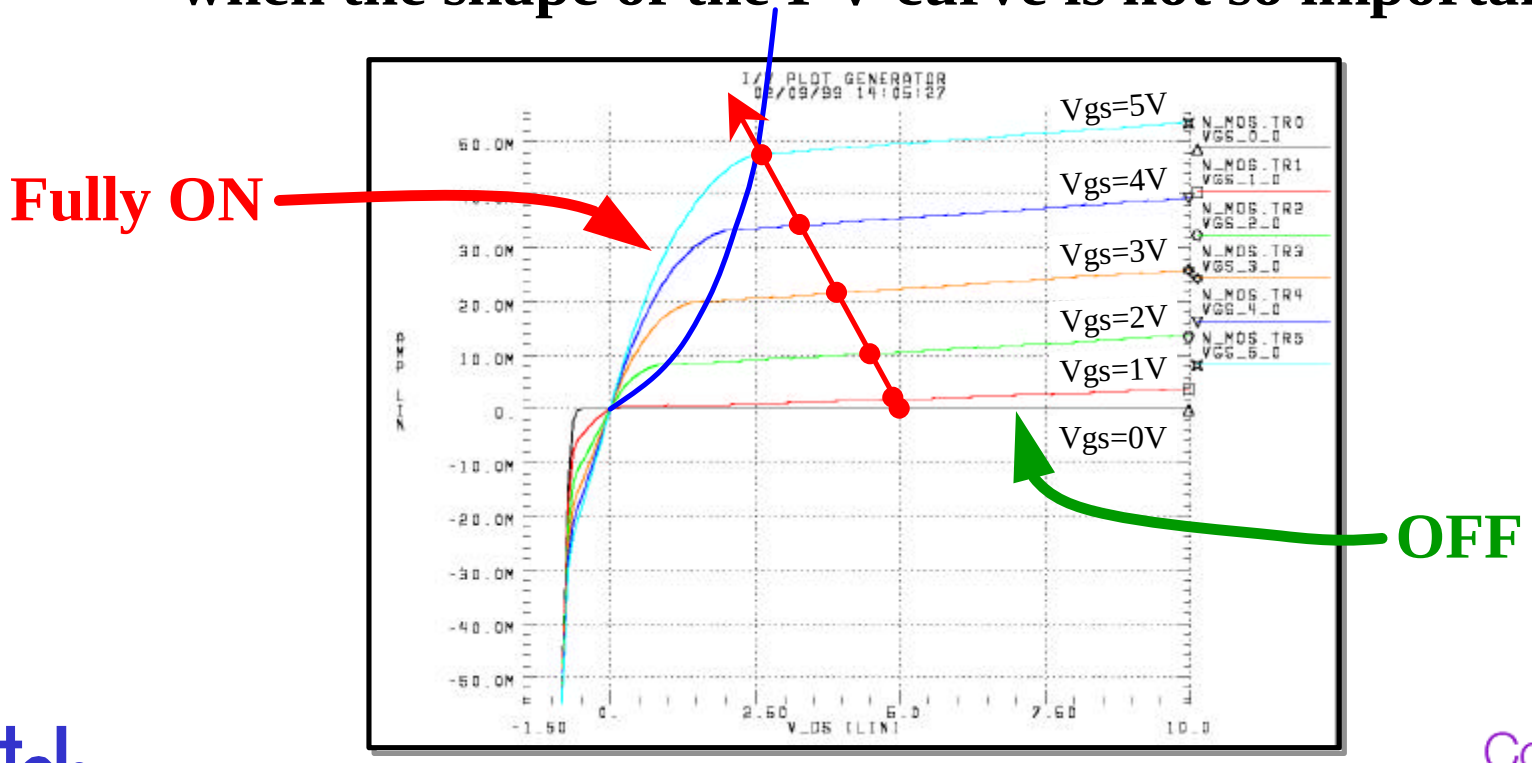
Fully ON



OFF

Linear scaling is usually not a problem

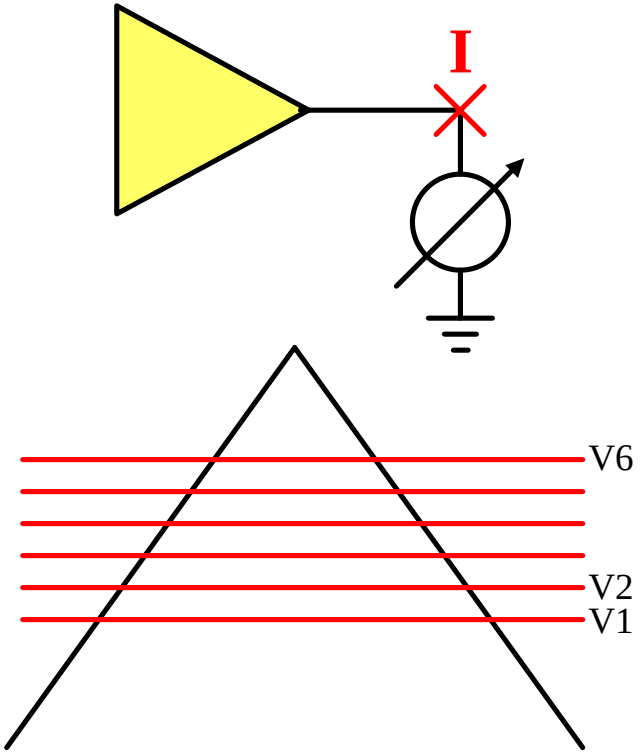
- Linear (vertical) scaling is most inaccurate when the buffer is barely turned on
- Walking along the load line shows that the buffer operates in the saturation region during this time when the shape of the I-V curve is not so important

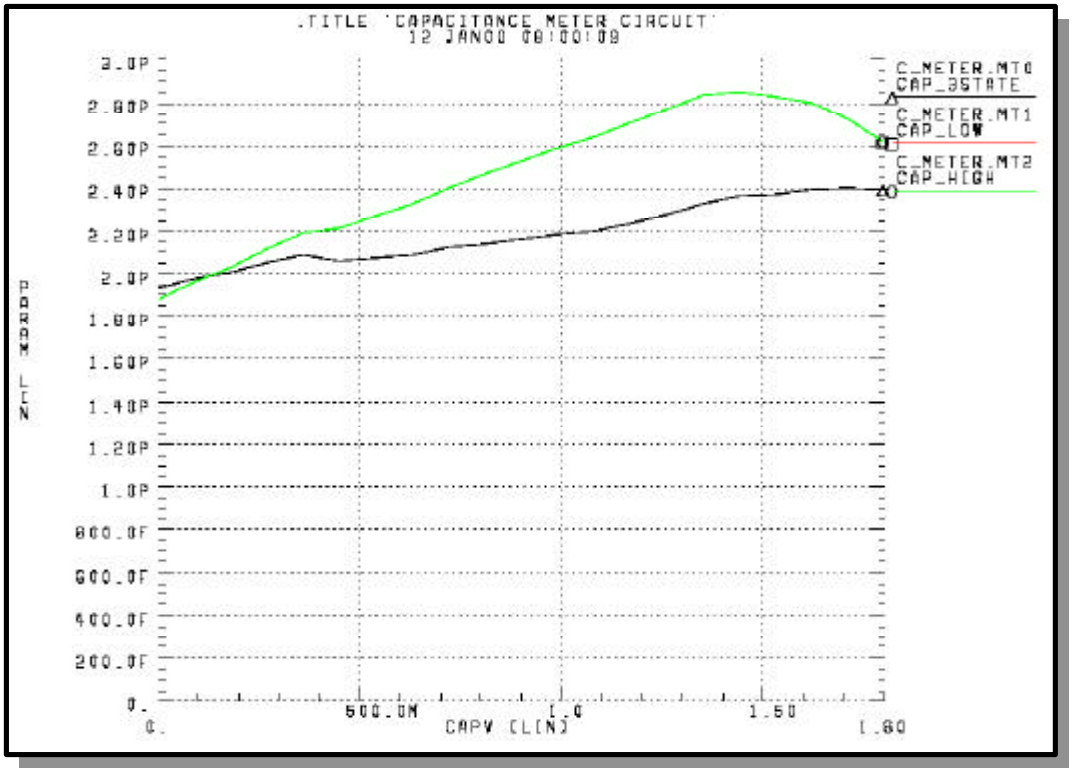


The meaning and importance of C_comp

- **C_comp is the total die capacitance as seen at the die pad**
  - parasitic capacitance of transistors and circuit elements (usually included in the process file)
  - metal capacitance connecting transistors with die pad (not necessarily included in netlist)
  - die pad capacitance (not necessarily included in netlist)
- **Do NOT include package capacitance in C_comp**
  - it is a common mistake to think of C_comp as the pin or I/O capacitance that is used in the data book which includes all capacitance as seen at the pin
- **C_comp is important even for driver models**
  - C_comp plays an important role in shaping the reflected waveforms at the driver
  - make sure C_comp matches what is in the SPICE model when correlating SPICE and IBIS models

Measuring C_comp with simulation





Using the same dV/dt for rise and fall, measure current on both slopes

(I1-I2)/2 = C*dV/dt

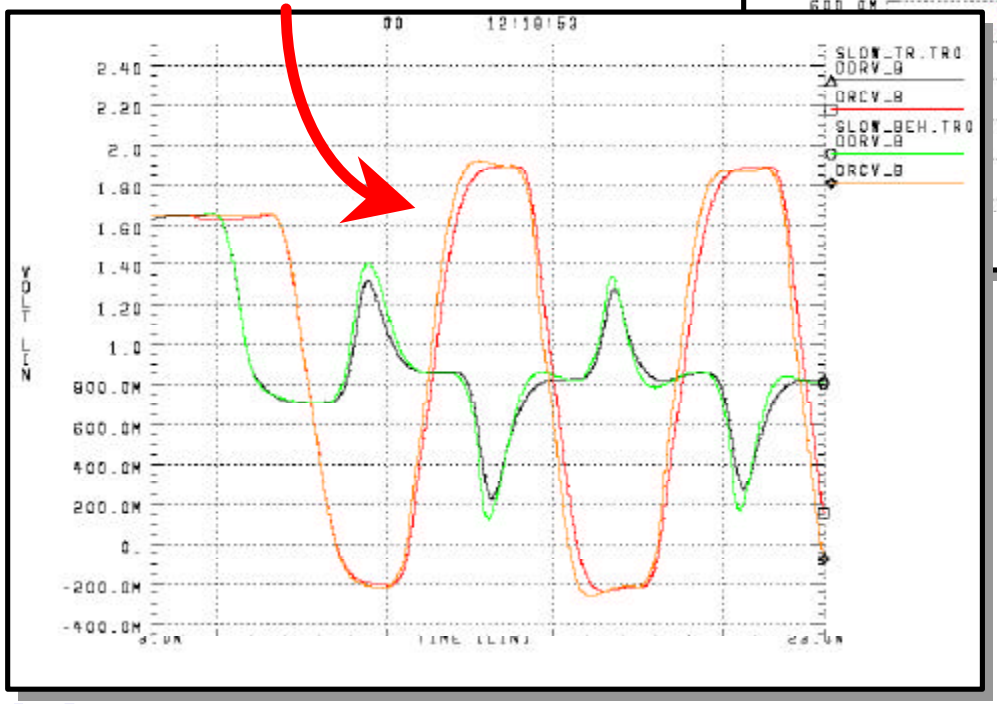
This technique works whether the buffer is 3-stated or not!

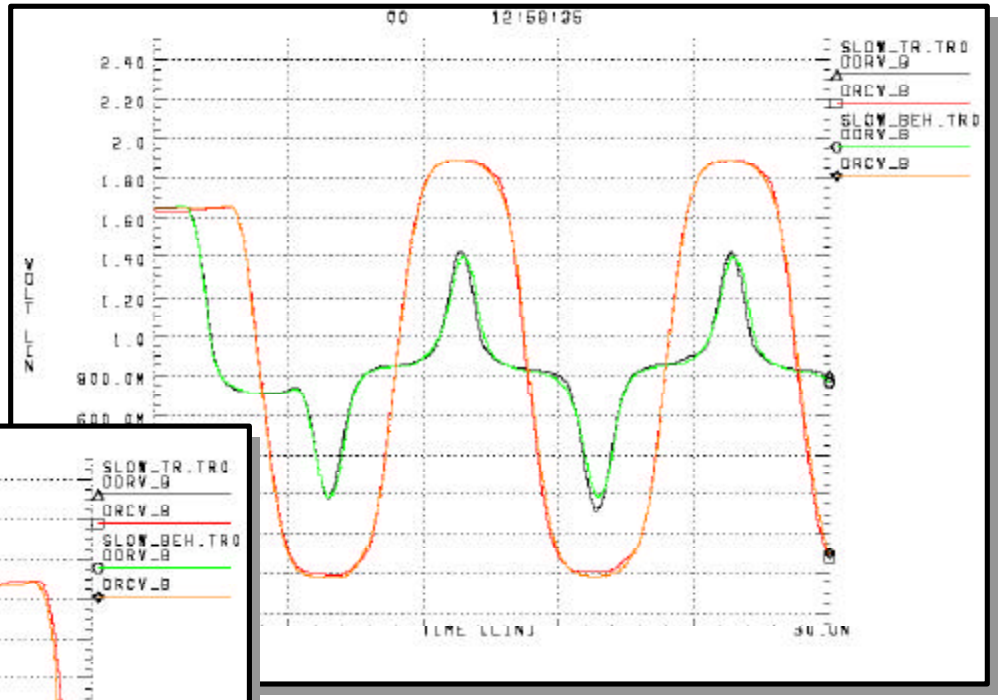
intel®

Platform Components GROUP

Results of a correlation study

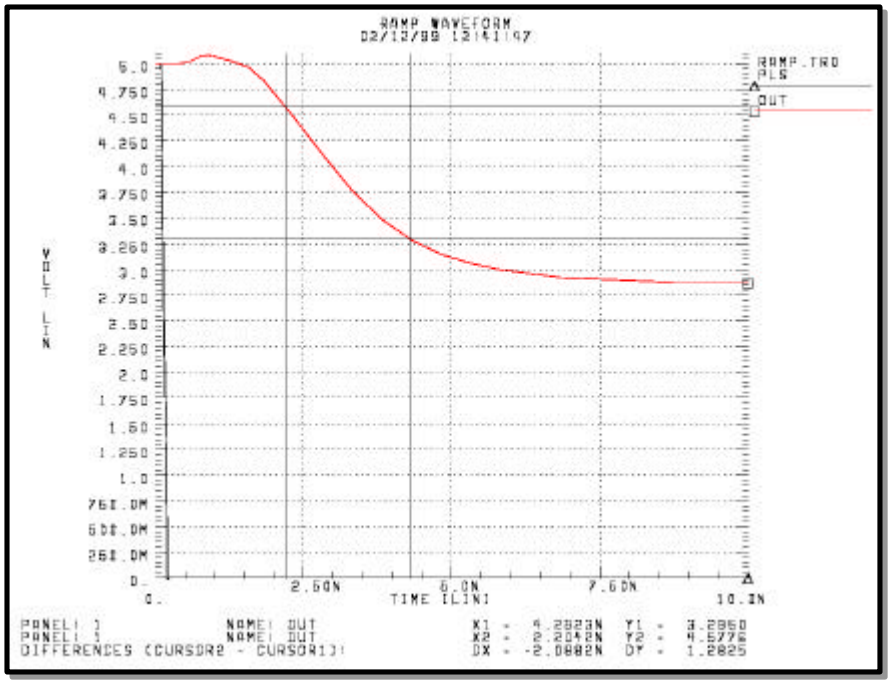
C_comp too large in driver model

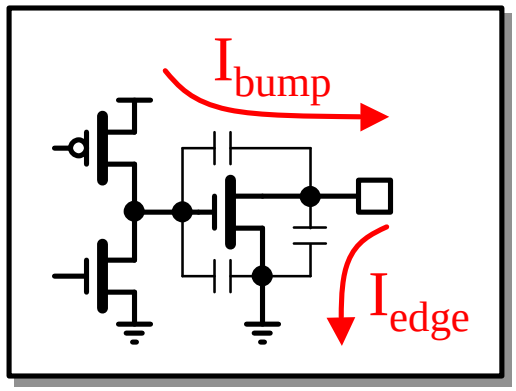




C_comp correct

Limitations of C_comp as is today





- **The bump preceding the falling edge is a capacitive coupling of the gate voltage to the output pad**
  - ⇨ if this V-t curve is used as a scaling coefficient to the pulldown I-V curve, the coefficient will go negative for the duration of the bump (to push up)
  - ⇨ in reality, the current for the gate to pad capacitor comes from the supply rail (for the falling edge and for an inverting transistor)
  - ⇨ if the predriver is connected to a different supply rail, this current may bounce another power pin

SPICE to IBIS checklist

1) Prepare a pin list for the component (.PIN file)
2) Prepare a clean SPICE netlist of the buffer for the I-V and V-t curve simulations
3) Run simulations for each buffer (.LIS files)
4) Convert each buffer's simulation output to IBIS format (.MRx files)
5) Combine individual buffer's IBIS models (.MRx) files into one IBIS model
6) Run IBISchk3 to verify the new IBIS model
7) Run SPICE vs. IBIS simulations to correlate new model

Preparing SPICE models

- **Prepare a clean SPICE netlist of the buffer for the I-V and V-t curve simulations**
  - make sure that you have the SPICE netlist and the process files for the buffer
  - remove all packaging, stimulus sources, transmission lines, test loads, etc. from “netlist” if there are any in it
  - make sure you understand the way the buffer needs to be connected to input, enable, output power, GND, reference voltages, and anything else it may need (strength selector, etc.)
  - find out the operating conditions of the buffer
    temperature (0 - 100 C, or anything else?)
    supply voltage (5.0 V, 3.3 V)
    supply voltage tolerance (± 5, or ±10 %)

Prepare the SPICE simulations (I-V)

- **I-V curve simulations**
  - could be done as a .DC sweep, or .TRAN simulation
  - .TRAN may work better in some cases (flip-flops, etc.)
  - if running in .TRAN mode use a slow sweep ramp
    parasitic capacitance of buffer can alter the actual currents if swept too fast (I=C*dV/dt)
  - set appropriate temperatures, supply voltages, sweep voltages, and time step for typical, minimum, and maximum curves
  - make sure you run the buffer in each mode
    driving high/low and 3-stated
  - measure pullup and power clamp curves relative to their supply rail
    (GND relative curves can be converted later if desired)
  - generate difference curves if necessary
    (this step could be done later also if so desired)

Prepare the SPICE simulations (V-t)

- **V-t curve simulations**
  - must be done with .TRAN simulation
  - select a small enough time step to get enough detail
  - make sure the length of the simulation is long enough to arrive to a steady state
    - last point must match with the I-V curve / load line operating point solution
  - set the *same* temperatures, supply voltages, for typical, minimum, and maximum curves as for the I-V curves
  - select a proper value for R_fixture
    - use the transmission line impedance value the buffer was designed to drive, or
    - the voltage swing of the buffer loaded with R_fixture should be about 1/2 to 2/3 of the full swing without the load
  - make sure you run the rising / falling edges with R_fixture connected to each rail once (for complementary buffers)
    - a minimum of four V-t curves per buffer are highly desirable
  - make sure that each V-t curve has a common time reference

Convert simulated data to IBIS format

- **Format simulation data to accommodate post processing tool (if any)**
  - this step may or may not be necessary
  - the HSPICE templates of this course are read directly by IBIS Center
- **Post process simulated data**
  - I-V curve subtraction
  - clamp I-V curve adjustments
  - reduce number of points to 100 per table
  - guardband, derate if necessary
- **Convert data to IBIS syntax**
- **Concatenate individual buffer models to form an IBIS model for a complete component**
  - you will need additional information about the buffer for the last two steps that was not available from simulation

Process files and typ. min., max.

- **Process files cover a very wide range too often**
  - six sigma coverage is usual
  - design engineers do need this for their work
- **It may be more useful to make IBIS models with a smaller range**
  - system designer may never find a solution with such variations
  - customers may never get a part that falls outside a 3-4 sigma range due to testing, sorting and QA
- **Use "realistic" fast / slow process files, or**
- **Use typical process files with derating factors**
  - run minimum (worst case) and maximum (best case) simulations with temperature and supply voltage variations and apply a "fudge factor" during post processing the data to arrive to a 3-4 sigma "realistic" range

Word of caution when derating curves

- **I-V curves can be scaled easily to adjust min., and max. curves**
- **V-t curves must match the adjusted I-V curves!**
  - new swing amplitude must be calculated from new I-V curve
- **Edge rate adjustment on V-t curves means horizontal shrinking / stretching of the curve**
  - be careful with normalizing the “lead in” part of the V-t curves
- **Don’t forget to adjust the [Ramp] numbers to match the new V-t curves**

HSPICE buffer model example

```
*********************************************************************************
.SUBCKT IO_buf input output power ground enable
*********************************************************************************
X1  enable  en_b            power  ground  INVERTER
X2  input   en_b    pre_p1  power  ground  NAND2
X3  input   enable  pre_n1  power  ground  NOR2
X4  pre_p1  pre_p2          power  ground  INVERTER   mult_p=2  mult_n=2
X5  pre_n1  pre_n2          power  ground  INVERTER   mult_p=2  mult_n=2
X6  pre_p2  pre_p2  gate_p  power  ground  NAND2      mult_p=4  mult_n=2
X7  pre_n2  pre_n2  gate_n  power  ground  NOR2       mult_p=2  mult_n=4
*
Mp  output gate_p power  power  PMOS  L=0.800U W=43.40U NRD=0.0897 NRS=0.0737
+                              AS=434.0P AD=217.0P PS=106.8U PD=53.40U
+                              M=12
Mn  output gate_n ground ground NMOS  L=0.800U W=43.40U NRD=0.0897 NRS=0.0714
+                              AS=434.0P AD=217.0P PS=106.8U PD=53.40U
+                              M=6
C1  pre_p2  ground  C=0.07pF
C2  pre_n2  ground  C=0.07pF
C3  gate_p  ground  C=0.04pF
C4  gate_n  ground  C=0.03pF
*
R1  input   in_R    R=200
C5  in_R    ground  C=0.2pF
X8  input   out_r           power  ground  INVERTER
*
*
*
```

HSPICE template example (part 1)

```
******************** V-t curve simulations *********************
.SUBCKT BUFFER  1      2      3      4      5      6      7
*               in     out    puref  PCLref pdref  GCLref /en
X0 1 2 3 5 7 IO_buf                $ <<<------ Change buffer name here
.ENDS
********************************************************************************
.PROTECT
.TRAN 0.01ns 10ns
********************************************************************************
.TEMP   50                         $ Temperature of typical case
*-------------------------------------------------------------------------------*
.PARAM  PUref_typ  = 5.000V        $ Pullup reference voltage, typ.
.PARAM  PUref_min  = 4.750V        $ Pullup reference voltage, min.
.PARAM  PUref_max  = 5.250V        $ Pullup reference voltage, max.
.PARAM  PCLref_typ = PUref_typ     $ Power clamp reference voltage, typ.
.PARAM  PCLref_min = PUref_min     $ Power clamp reference voltage, min.
.PARAM  PCLref_max = PUref_max     $ Power clamp reference voltage, max.
*-------------------------------------------------------------------------------*
.PARAM  PDref_typ  = 0.000V        $ Pulldown reference voltage, typ.
.PARAM  PDref_min  = 0.000V        $ Pulldown reference voltage, min.
.PARAM  PDref_max  = 0.000V        $ Pulldown reference voltage, max.
.PARAM  GCLref_typ = 0.000V        $ GND clamp reference voltage, typ.
.PARAM  GCLref_min = 0.000V        $ GND clamp reference voltage, min.
.PARAM  GCLref_max = 0.000V        $ GND clamp reference voltage, max.
********************************************************************************
.PARAM  Vpdref     = PDref_typ     $ Reference voltages for typical case
.PARAM  VGNDclref  = GCLref_typ
.PARAM  Vpuref     = PUref_typ
.PARAM  VPOWclref  = PCLref_typ
*-------------------------------------------------------------------------------*
.PARAM  Ven        = Vpdref        $ Active-low enable
*.PARAM  Ven        = Vpuref       $ Active-high enable
********************************************************************************
```

HSPICE template example (part 2)

```
*******************************************************************************
.PARAM  Vfx_pd_on  = Vpuref
.PARAM  Vfx_pd_off = Vpuref
.PARAM  Vfx_pu_on  = Vpdref
.PARAM  Vfx_pu_off = Vpdref
*
.PARAM  Rfx_pd_on  = 50
.PARAM  Rfx_pd_off = 50
.PARAM  Rfx_pu_on  = 50
.PARAM  Rfx_pu_off = 50
*
.PARAM  Cfx_pd_on  = 0.0pF
.PARAM  Cfx_pd_off = 0.0pF
.PARAM  Cfx_pu_on  = 0.0pF
.PARAM  Cfx_pu_off = 0.0pF
*******************************************************************************
.MEASURE TRAN  Vpower           PARAM = 'Vpuref-Vpdref'
.MEASURE TRAN  Vfixture_pd_on   PARAM = Vfx_pd_on
.MEASURE TRAN  Vfixture_pd_off  PARAM = Vfx_pd_off
.MEASURE TRAN  Vfixture_pu_on   PARAM = Vfx_pu_on
.MEASURE TRAN  Vfixture_pu_off  PARAM = Vfx_pu_off
.MEASURE TRAN  Rfixture_pd_on   PARAM = Rfx_pd_on
.MEASURE TRAN  Rfixture_pd_off  PARAM = Rfx_pd_off
.MEASURE TRAN  Rfixture_pu_on   PARAM = Rfx_pu_on
.MEASURE TRAN  Rfixture_pu_off  PARAM = Rfx_pu_off
.MEASURE TRAN  Cfixture_pd_on   PARAM = Cfx_pd_on
.MEASURE TRAN  Cfixture_pd_off  PARAM = Cfx_pd_off
.MEASURE TRAN  Cfixture_pu_on   PARAM = Cfx_pu_on
.MEASURE TRAN  Cfixture_pu_off  PARAM = Cfx_pu_off
*******************************************************************************
.OPTIONS ACCURATE BRIEF INGOLD NUMDGT=8 CO=132 ACCT=0 NOWARN
*.OPTIONS POST=1 PROBE
.PRINT TRAN Pulldown_on  = V(Out_pd_on)
+           Pulldown_off = V(Out_pd_off)
+           Pullup_on    = V(Out_pu_on)
+           Pullup_off   = V(Out_pu_off)
*******************************************************************************
```

HSPICE template example (part 3)

```
*******************************************************************************
Vrefpd      pdref    0  DC = Vpdref
VrefGNDcl   GCLref   0  DC = VGNDclref
Vrefpu      puref    0  DC = Vpuref
VrefPOWcl   PCLref   0  DC = VPOWclref
*
Von         Von      0  DC = Ven
Voff        Voff     0  DC = 'Vpuref-(Ven-Vpdref)'
*
Vpls_r  Pls_r  0  PWL  0.0ns Vpdref 0.1ns Vpuref 100ns Vpuref
Vpls_f  Pls_f  0  PWL  0.0ns Vpuref 0.1ns Vpdref 100ns Vpdref
*************************************************************** pull-down on
Xdrv_pd_on    Pls_f      Out_pd_on    puref  PCLref  pdref  GCLref   Von  BUFFER
Rfxt_pd_on    V_pd_on    Out_pd_on    Rfx_pd_on
Cfxt_pd_on    V_pd_on    Out_pd_on    Cfx_pd_on
Vfxt_pd_on    V_pd_on    0            Vfx_pd_on
*------------------------------------------------------------- pull-down off
Xdrv_pd_off   Pls_r      Out_pd_off   puref  PCLref  pdref  GCLref   Von  BUFFER
Rfxt_pd_off   V_pd_off   Out_pd_off   Rfx_pd_off
Cfxt_pd_off   V_pd_off   Out_pd_off   Cfx_pd_off
Vfxt_pd_off   V_pd_off   0            Vfx_pd_off
*--------------------------------------------------------------- pull-up on
Xdrv_pu_on    Pls_r      Out_pu_on    puref  PCLref  pdref  GCLref   Von  BUFFER
Rfxt_pu_on    V_pu_on    Out_pu_on    Rfx_pu_on
Cfxt_pu_on    V_pu_on    Out_pu_on    Cfx_pu_on
Vfxt_pu_on    V_pu_on    0            Vfx_pu_on
*------------------------------------------------------------- pull-up off
Xdrv_pu_off   Pls_f      Out_pu_off   puref  PCLref  pdref  GCLref   Von  BUFFER
Rfxt_pu_off   V_pu_off   Out_pu_off   Rfx_pu_off
Cfxt_pu_off   V_pu_off   Out_pu_off   Cfx_pu_off
Vfxt_pu_off   V_pu_off   0            Vfx_pu_off
*******************************************************************************
```

HSPICE template example (part 4)

```
*******************************************************************************
.UNPROTECT
*******************************************************************************
.ALTER                              $ Minimum case
.PROTECT
.TEMP   100                         $ Temperature for minimum case
.PARAM  Vpdref      = PDref_min     $ Reference voltages for minimum case
.PARAM  VGNDclref   = GCLref_min
.PARAM  Vpuref      = PUref_min
.PARAM  VPOWclref   = PCLref_min
.UNPROTECT
*******************************************************************************
.ALTER                              $ Maximum case
.PROTECT
.TEMP   0                           $ Temperature for maximum case
.PARAM  Vpdref      = PDref_max     $ Reference voltages for maximum case
.PARAM  VGNDclref   = GCLref_max
.PARAM  Vpuref      = PUref_max
.PARAM  VPOWclref   = PCLref_max
.UNPROTECT
*******************************************************************************
.END
```

HSPICE template example (part 5)

```
********************* I-V curve simulations *********************
.SUBCKT BUFFER  1      2      3      4      5      6      7
*               in     out    puref  PCLref pdref  GCLref /en
X0 1 2 3 5 7 IO_buf                $ <<<------ Change buffer name here
.ENDS
********************************************************************************
.PROTECT
.TRAN 1.0ms 300.0001ms START=0.1us
********************************************************************************
.PARAM  Sweep_t    = 300.0001ms    $ Match this with .TRAN end time above!!!
.PARAM  Sweep_d    = 0.1us         $ Delay before sweep begins
.TEMP   50                         $ Temperature of typical case
*-------------------------------------------------------------------------------*
.PARAM  PUref_typ  = 5.000V        $ Pullup reference voltage, typ.
.PARAM  PUref_min  = 4.750V        $ Pullup reference voltage, min.
.PARAM  PUref_max  = 5.250V        $ Pullup reference voltage, max.
.PARAM  PCLref_typ = PUref_typ     $ Power clamp reference voltage, typ.
.PARAM  PCLref_min = PUref_min     $ Power clamp reference voltage, min.
.PARAM  PCLref_max = PUref_max     $ Power clamp reference voltage, max.
*-------------------------------------------------------------------------------*
.PARAM  PDref_typ  = 0.000V        $ Pulldown reference voltage, typ.
.PARAM  PDref_min  = 0.000V        $ Pulldown reference voltage, min.
.PARAM  PDref_max  = 0.000V        $ Pulldown reference voltage, max.
.PARAM  GCLref_typ = 0.000V        $ GND clamp reference voltage, typ.
.PARAM  GCLref_min = 0.000V        $ GND clamp reference voltage, min.
.PARAM  GCLref_max = 0.000V        $ GND clamp reference voltage, max.
********************************************************************************
.PARAM  Vpdref     = PDref_typ     $ Reference voltages for typical case
.PARAM  VGNDclref  = GCLref_typ
.PARAM  Vpuref     = PUref_typ
.PARAM  VPOWclref  = PCLref_typ
*-------------------------------------------------------------------------------*
.PARAM  Ven        = Vpdref        $ Active-low enable
*.PARAM  Ven        = Vpuref       $ Active-high enable
********************************************************************************
```

HSPICE template example (part 6)

```
*********************************************************************************
.MEASURE TRAN  Vpower        PARAM = 'Vpuref-Vpdref'
.MEASURE TRAN  Pulldown_ref  PARAM = Vpdref
.MEASURE TRAN  GND_cl_ref    PARAM = VGNDclref
.MEASURE TRAN  Pullup_ref    PARAM = Vpuref
.MEASURE TRAN  POWER_cl_ref  PARAM = VPOWclref
*********************************************************************************
.OPTIONS BRIEF INGOLD NUMDGT=8 CO=132 ACCT=0 NOWARN $ ACCURATE
*.OPTIONS POST=1 PROBE
.PRINT TRAN
+ V_sweep      = PAR('(PCLref_typ-GCLref_typ)*(3*(TIME-Sweep_d)/(Sweep_t-Sweep_d)-1)')
+ I_pulldown   = I(Vpulldown)
+ I_gndclamp   = I(Vgnd_clamp)
+ I_pullup     = I(Vpullup)
+ I_powerclamp = I(Vpower_clamp)
*********************************************************************************
Vrefpd     pdref   0  DC = Vpdref
VrefGNDcl  GCLref  0  DC = VGNDclref
Vrefpu     puref   0  DC = Vpuref
VrefPOWcl  PCLref  0  DC = VPOWclref
*
Von        Von     0  DC = Ven
Voff       Voff    0  DC = 'Vpuref-(Ven-Vpdref)'
*
Vpulldown     pdref   pulldown     PWL
+  0.0us                                  0
+  Sweep_d                                'PUref_typ-PDref_typ'
+  Sweep_t                                '-2*(PUref_typ-PDref_typ)'
Vgnd_clamp    GCLref  gnd_clamp    PWL
+  0.0us                                  0
+  Sweep_d                                'PCLref_typ-GCLref_typ'
+  Sweep_t                                '-2*(PCLref_typ-GCLref_typ)'
```

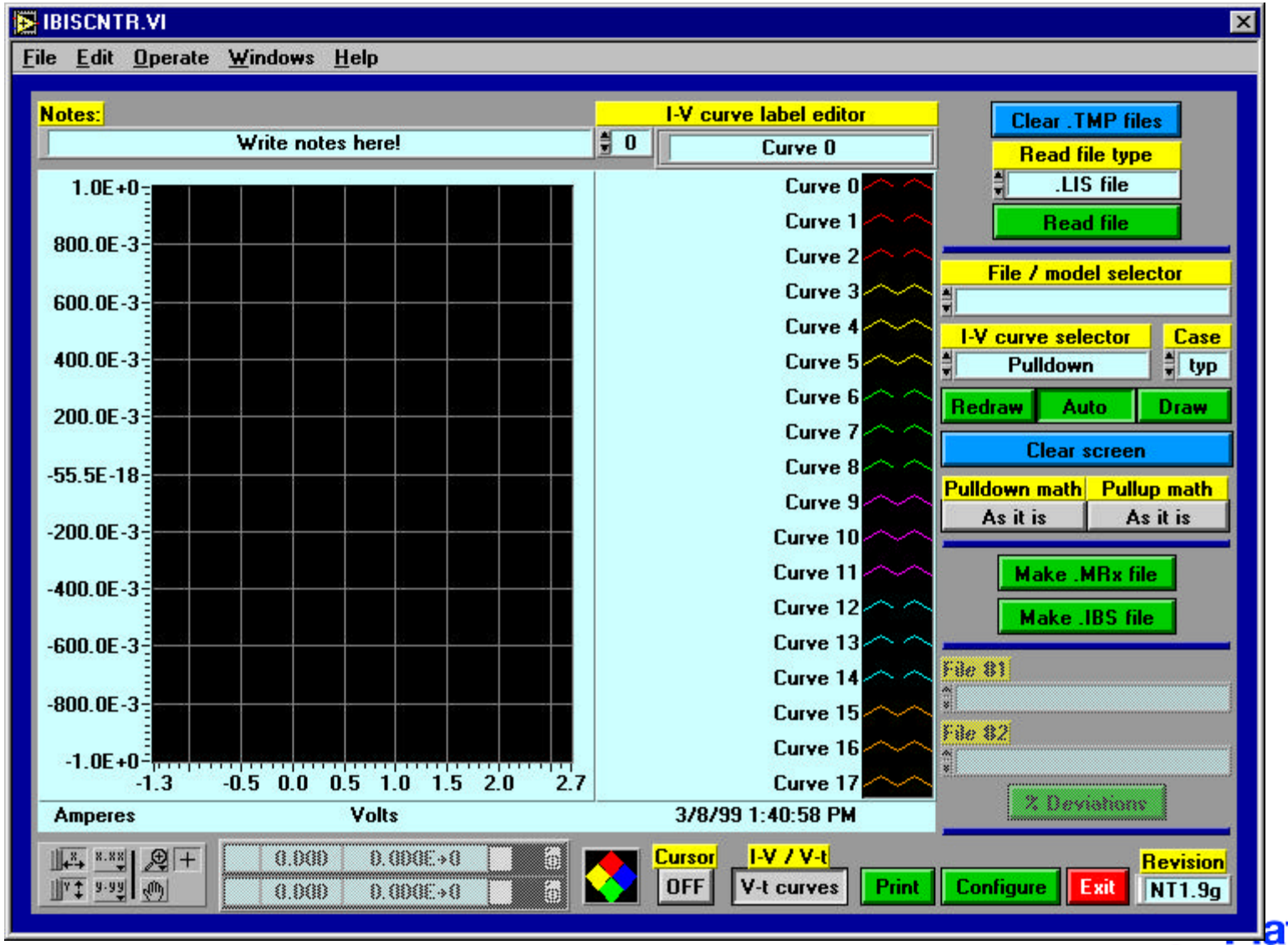
HSPICE template example (part 7)

```
Vpullup       puref   pullup       PWL
+  0.0us                                        0
+  Sweep_d                                      '-1*( PUref_typ-PDref_typ)'
+  Sweep_t                                      '2*( PUref_typ-PDref_typ)'
Vpower_clamp  PCLref  power_clamp  PWL
+  0.0us                                        0
+  Sweep_d                                      '-1*( PCLref_typ-GCLref_typ)'
+  Sweep_t                                      '2*( PCLref_typ-GCLref_typ)'
*******************************************************************************
Xpulldown   pdref  pulldown     puref  PCLref  pdref  GCLref   Von   BUFFER
Xgnd_clamp  pdref  gnd_clamp    puref  PCLref  pdref  GCLref  Voff   BUFFER
*-------------------------------------------------------------------------------*
Xpullup     puref  pullup       puref  PCLref  pdref  GCLref   Von   BUFFER
Xpow_clamp  puref  power_clamp  puref  PCLref  pdref  GCLref  Voff   BUFFER
.UNPROTECT
*******************************************************************************
.ALTER                              $ Minimum case
.PROTECT
.TEMP   100                         $ Temperature for minimum case
.PARAM  Vpdref     = PDref_min      $ Reference voltages for minimum case
.PARAM  VGNDclref  = GCLref_min
.PARAM  Vpuref     = PUref_min
.PARAM  VPOWclref  = PCLref_min
.UNPROTECT
*******************************************************************************
.ALTER                              $ Maximum case
.PROTECT
.TEMP   0                           $ Temperature for maximum case
.PARAM  Vpdref     = PDref_max      $ Reference voltages for maximum case
.PARAM  VGNDclref  = GCLref_max
.PARAM  Vpuref     = PUref_max
.PARAM  VPOWclref  = PCLref_max
.UNPROTECT
*******************************************************************************
.END
```

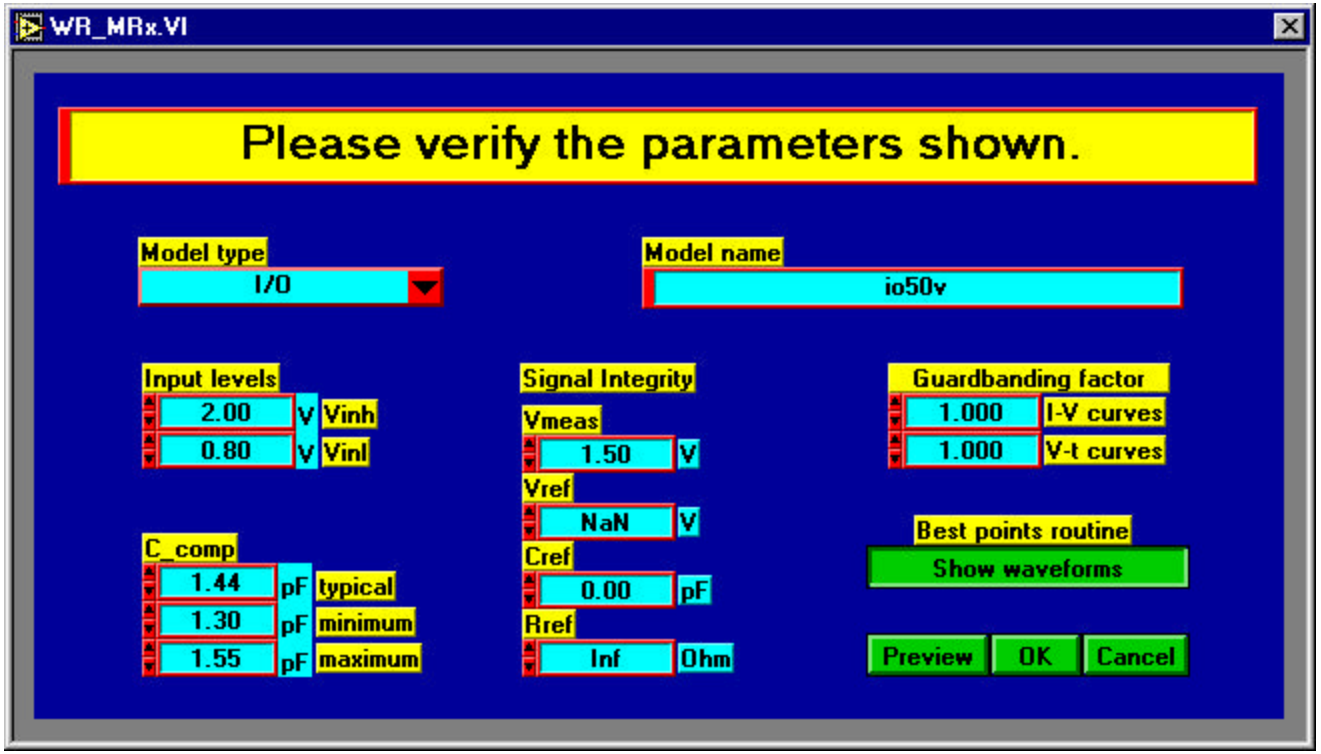
IBIS Center

- **Privately written tool (using LabVIEW)**
- **Supports only up to IBIS 2.1 “buffer” features**
  - [Buffer Selector] is one exception (comes in pin list file)
  - does not support package or EBD modeling
- **Not available to the public - no support available**
- **Post processes simulation data**
  - it reads .LIS files generated by the HSPICE templates
  - .LIS files can be generated by any other SPICE tool
- **Converts SPICE output to IBIS format**
  - subtracts I-V curves appropriately
  - adjusts clamp curves if necessary (for constantly on pu/pd)
  - reduces I-V and V-t data to best 100 points
  - checks for I-V and V-t curve mismatches
- **Builds full IBIS model for a component**
  - concatenates individual buffer models (.MRx files) based on a pin list file
  - checks for electrical equivalency so identical models would not repeat

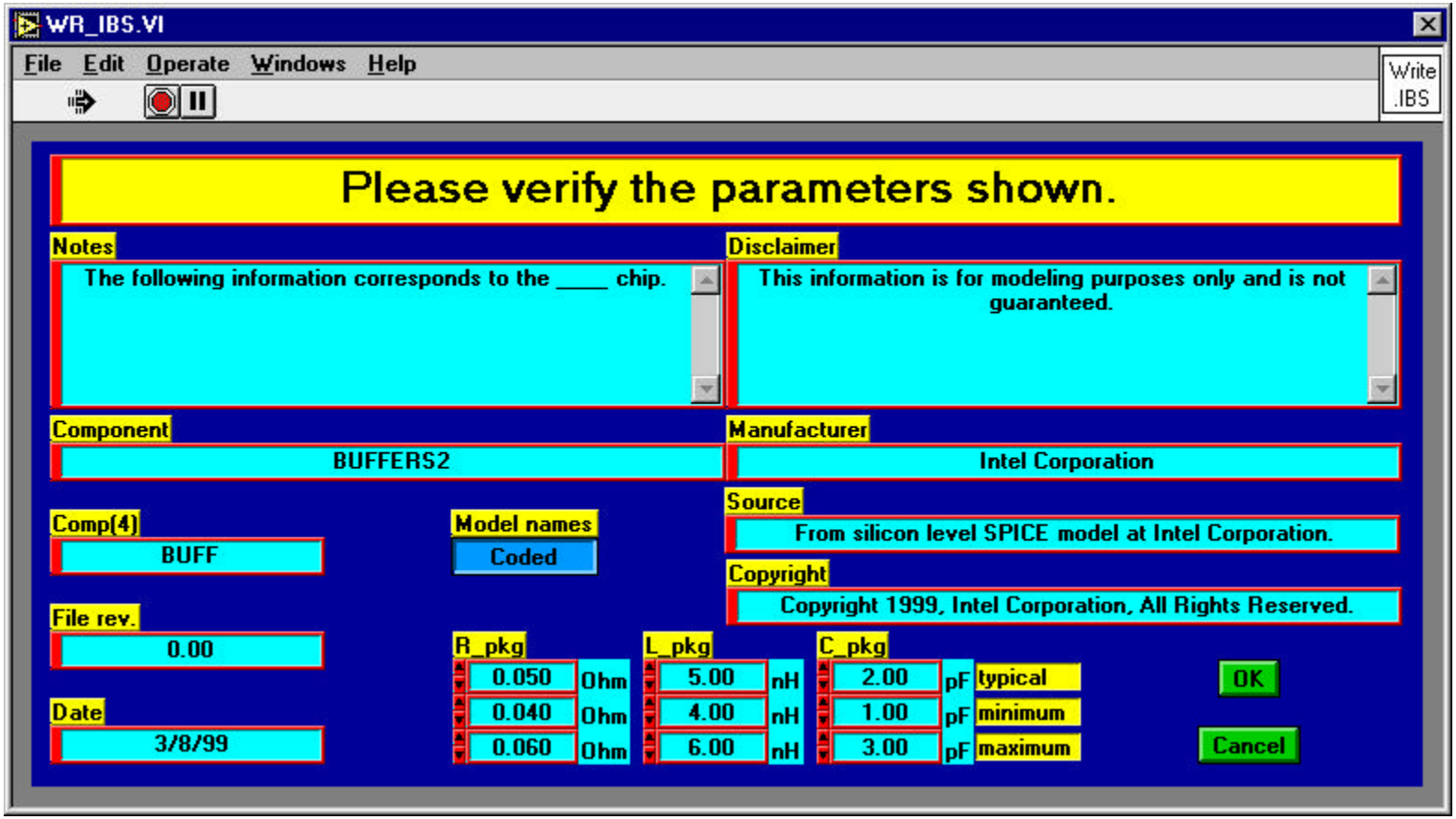
Front panel of IBIS Center



Make .MRx file panel of IBIS Center



Make .IBS file panel of IBIS Center



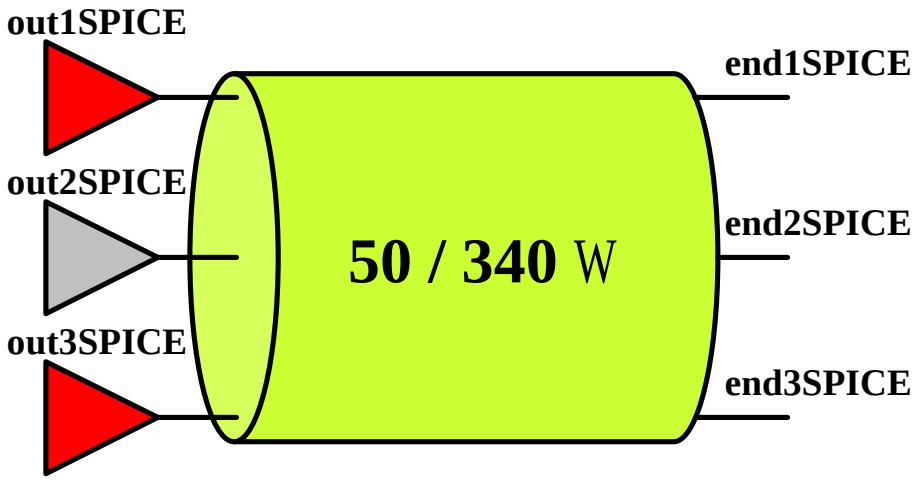
Additional information

- **Collect data for those parameters which are not available directly from SPICE simulations**
  - model type (I/O can be used as input or output only)
  - Vinh, Vinl
  - C_comp (may need to run another simulation for this)
  - package L, R, C
  - Vmeas, Vref, Rref, Cref
  - component name
  - copyright
  - source of the model (measured, simulated)
  - disclaimers

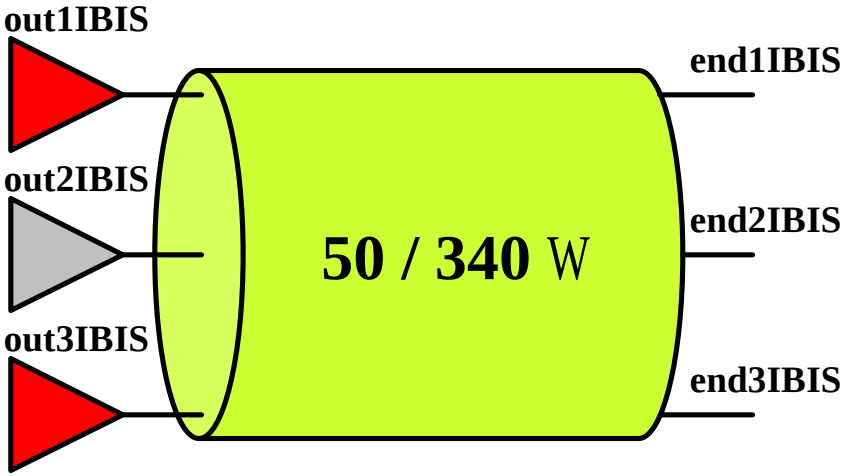
Lab exercise #1

- **Run the HSPICE template for each buffer**
  - open the templates and the SPICE models of the buffers to make sure the buffer name is correct and it is connected properly
  - hspice -i io50v.sp -o io50v.lis
- **Make a .MR1 file from each .LIS file using IBIS Center**
  - read in each .LIS file
  - while making .MR1 file for each, provide appropriate information that is not available in the .LIS file
  - observe “best 100 points” routine (it is fun to watch it)
- **Make a .IBS file from the .MR1 files using IBIS Center**
  - open the .PIN file and look at the pin/signal name/buffer name list
  - press “Make .IBS file” button and fill out the appropriate boxes
- **Verify the new IBIS model**
  - run an HSPICE simulation with the original SPICE model and the new IBIS model side by side and compare the waveforms (compare.sp)

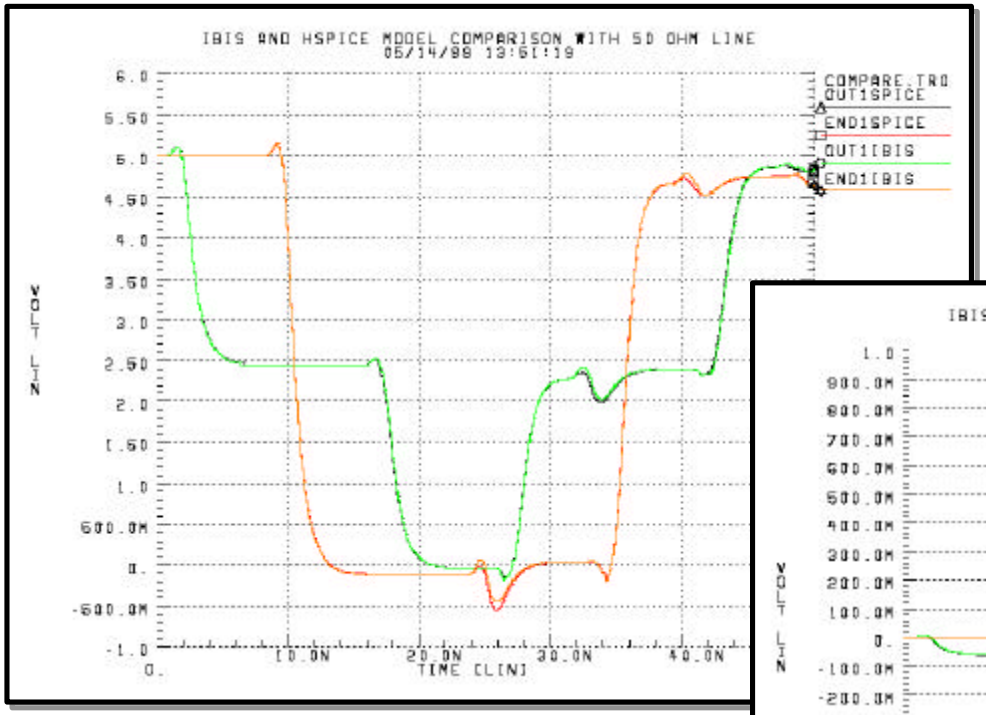
SPICE and IBIS simulation comparison



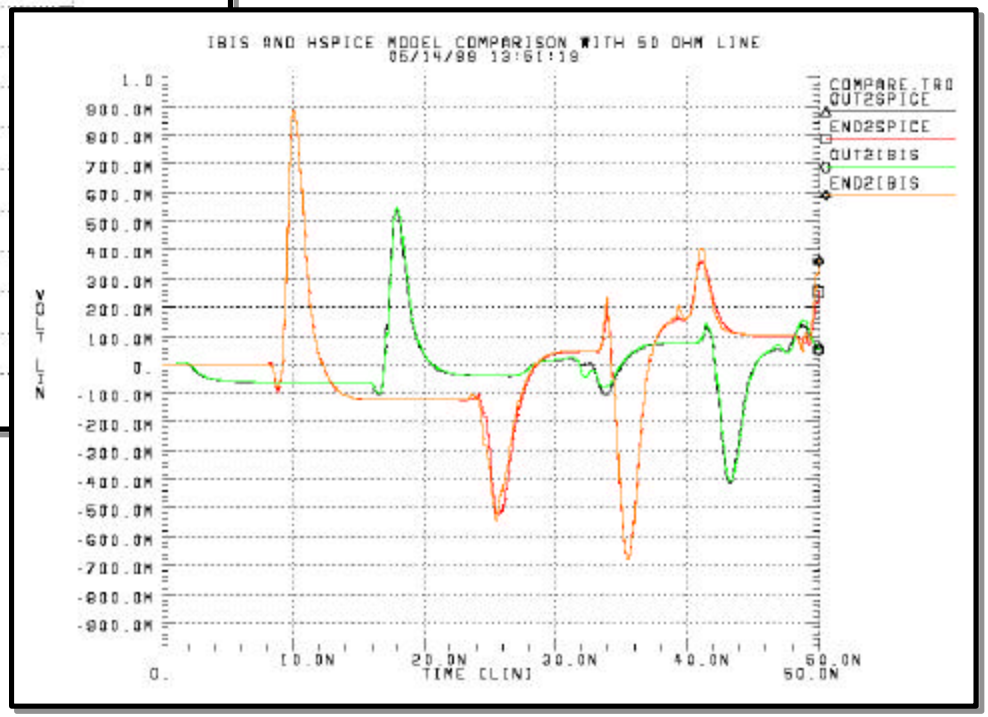
The IBIS model's waveforms were generated with a 50 W load for these buffers



Waveforms with 50 Ω Line

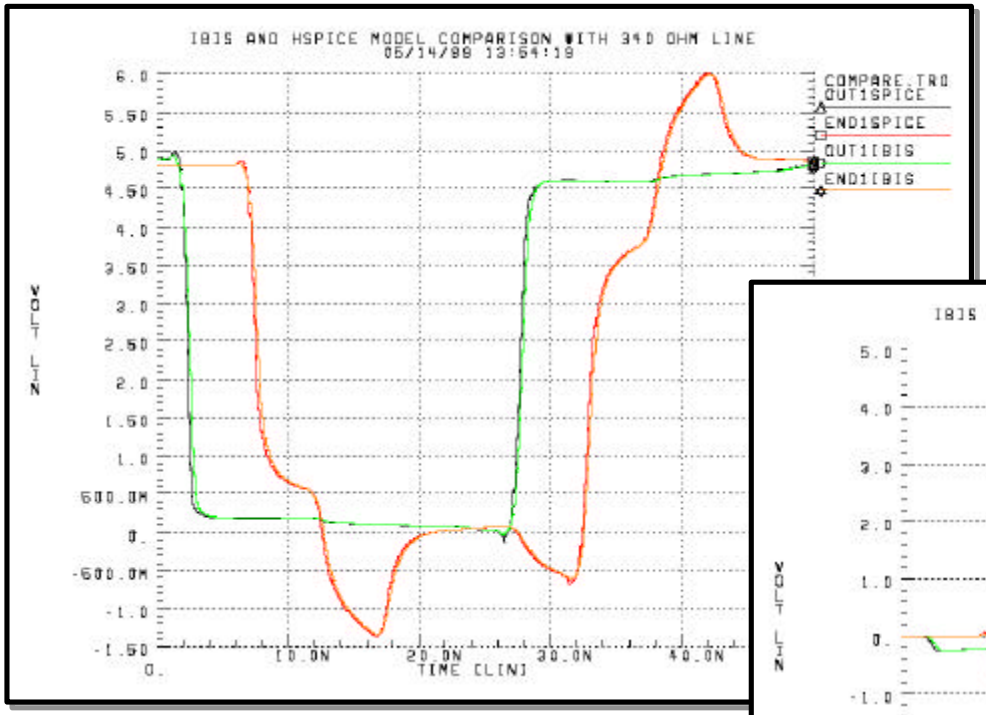


Driven line

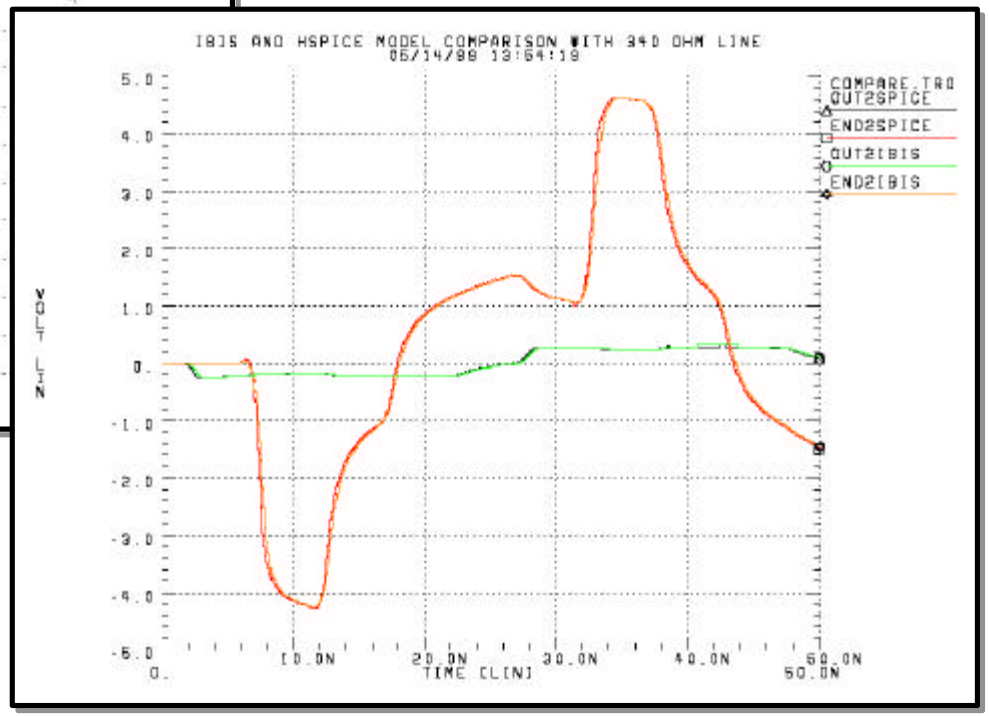


Cross talk on quiet line

Waveforms with 340 W Line



Driven line



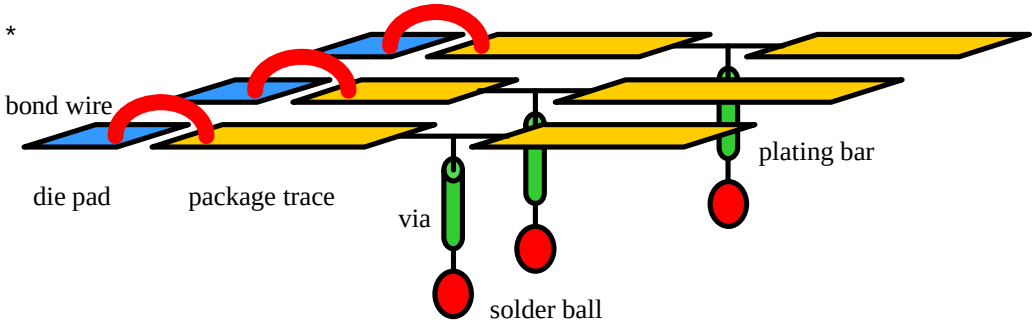
Cross talk on quiet line

Package modeling in IBIS

- **[Package] keyword**
  - this is a required section for each component
  - contains typical, minimum and maximum values for R_pkg, L_pkg and C_pkg
- **[Pin] keyword**
  - each pin can have a distinct R_pin, L_pin or C_pin value
  - these override the values under the [Package] keyword
  - only a single value can be used (no typ., min., max.)
- **[Package Model] keyword**
  - this can reference an external file or [Define Package Model] within the same .IBS file
  - overrides the values under the [Package] keyword
  - two methods possible currently
    - coupled single lump RLC matrix description (full, banded, and sparse matrix formats available)
    - uncoupled multi-section description using RLC, length

BGA package examples

```
|*****************************************
|
[Define Package Model] BGA example
[Manufacturer]          Noname Corp.
[OEM]                   Noname Corp.
[Description]           BGA
[Number Of Sections]    5
[Number Of Pins]        3
[Pin Numbers]
|
A1 Len=0 R=0.160 L=5.0n /       | bond wire
Len=4.00 L=0.170n C=0.250p /    | package trace
Fork
  Len=0.80 L=0.200n C=0.100p / | plating bar
  Endfork
Len=0.010 L=0.010n C=0.020p /   | via
Len=0 C=2.0p /                  | Ball 1
|
A2 Len=0 R=0.160 L=5.0n /       | bond wire
Len=2.50 L=0.170n C=0.250p /    | package trace
Fork
  Len=0.80 L=0.200n C=0.100p / | plating bar
  Endfork
Len=0.010 L=0.010n C=0.020p /   | via
Len=0 C=2.0p /                  | Ball 2
|
A3 Len=0 R=0.160 L=5.0n /       | bond wire
Len=4.00 L=0.170n C=0.250p /    | package trace
Fork
  Len=0.80 L=0.200n C=0.100p / | plating bar
  Endfork
Len=0.010 L=0.010n C=0.020p /   | via
Len=0 C=2.0p /                  | Ball 3
```




```
| The resistance matrix for this package has no coupling
|
[Resistance Matrix]     Banded_matrix
[Bandwidth]             0
[Row]   1
10.0
[Row]   2
15.0
[Row]   3
15.0
|
[Inductance Matrix]     Full_matrix
[Row]   1
3.04859e-07      4.73185e-08      1.3428e-08
[Row]   2
3.04859e-07      4.73185e-08
[Row]   3
3.04859e-07
|
|
[Capacitance Matrix]    Sparse_matrix
[Row]   1
```

Single Line Equivalent Method

- **Multi section package description is uncoupled**
  - even/odd mode model must be made separately to account for the coupling effects

$$Z=\mathrm{Sqrt}(L/C) \quad Z_{even}=\mathrm{Sqrt}(L_{even}/C_{even}) \quad Z_{odd}=\mathrm{Sqrt}(L_{odd}/C_{odd})$$

$$v=1/\mathrm{Sqrt}(L*C) \quad v_{even}=1/\mathrm{Sqrt}(L_{even}*C_{even}) \quad v_{odd}=1/\mathrm{Sqrt}(L_{odd}*C_{odd})$$

$$L_{even}=L_{11}+L_{12} \quad L_{odd}=L_{11}-L_{12}$$

$$C_{even}=C_{11}-C_{12} \quad C_{odd}=C_{11}+C_{12}$$

Where C_{11} is the total capacitance of line 1 ($C_{line1\_to\_GND}$ + C_{12}).

Electrical Board Description (EBD)

- **Even though the main focus of IBIS was buffer modeling, this capability became necessary for memory modules, processor cartridges, multi chip modules, etc.**
  - these kind of devices are often sold as a “canned unit”
  - very difficult to get routing information (Gerber files) for memory modules,
  - it is increasingly more important to simulate traces as transmission lines
- **The EBD syntax is very similar to the uncoupled multi-section package description**
  - uncoupled sections using RLC and length
  - two additional features are “node” and “pin”

Connector modeling in IBIS

- **Not available yet**
  - some tools use the package syntax as a workaround mixed with tool specific features and/or syntax
- **The connector subcommittee turned in a connector specification proposal Feb. 1, 1999**
  - version 0.941 was uploaded to IBIS web site on 3/11/2000
- **This proposal is currently under discussion and is expected to be voted on around mid 2000**
  - supports single/multi line models
  - supports cascaded model matrixes
  - supports swath matrixes
  - supports auto generated models (define number of pins)
  - supports connectors in multiple configurations (mated / unmated, board edge / solder tail / press fit, etc.)

Advanced buffer modeling

- **Pullup or pulldown "resistors"**
  - they prevent 3-stated buses from floating around the threshold voltages
  - usually in the kΩ range (I_{sat} in μA range)
  - usually implemented as a transistor turned on constantly
- **Bus hold circuits (may be dynamic)**
  - similar to pu/pd resistor idea, but usually has a lower impedance
  - could be time, edge or level dependent if dynamic
- **Integrated terminators**
  - static transmission line termination (low impedance)
  - dynamic implementations designed to save power
- **Dynamic clamping mechanisms**
  - strong clamps turn on momentarily to prevent excessive overshoot
- **Staged buffers**
  - mostly used in slew rate controlled drivers
- **Kicker circuits**
  - transition boosters and then turn off
- **Anything else you can invent goes here...**

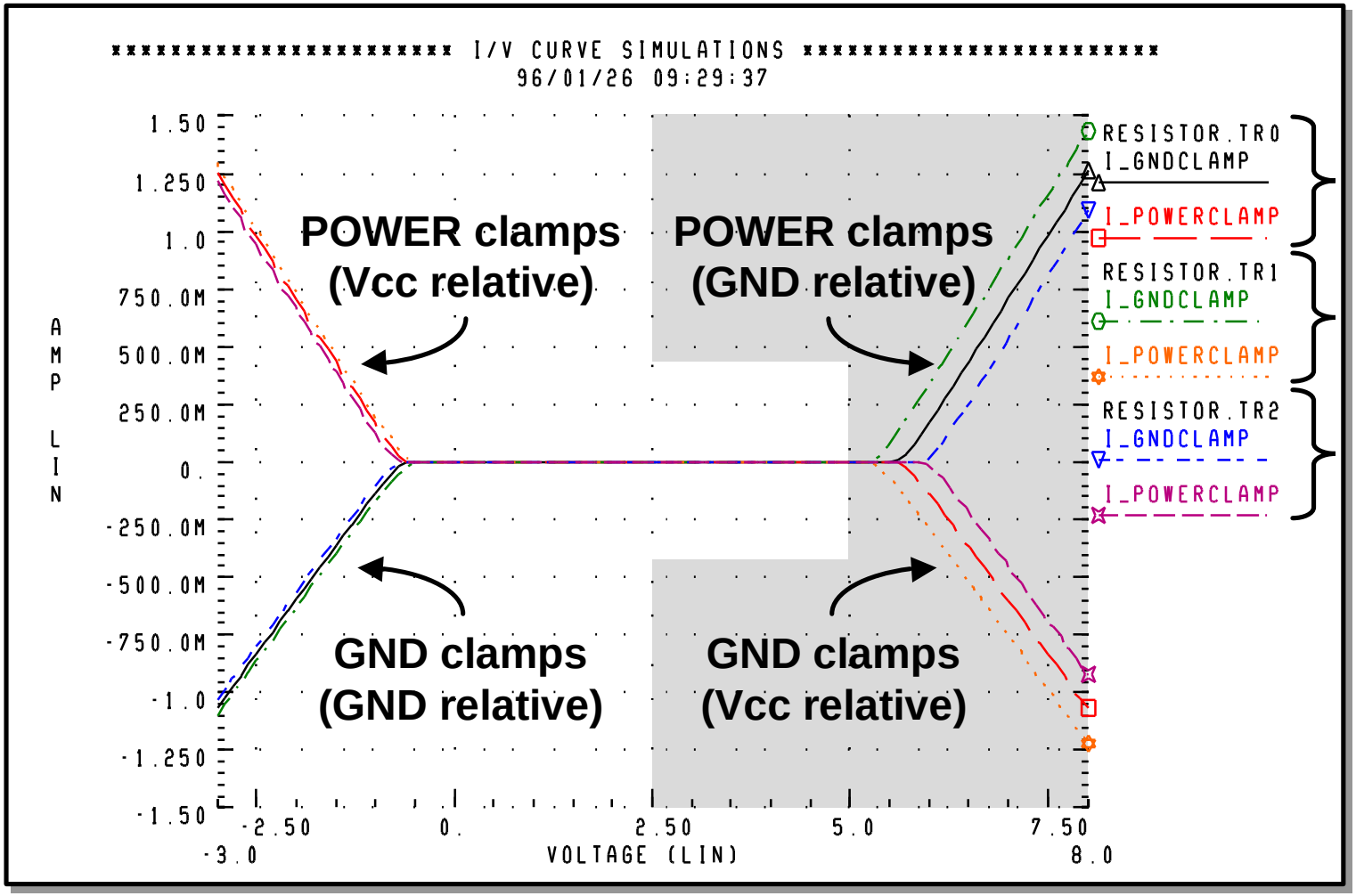
Modeling static advanced features

- **Anything that is ON constantly should be modeled using the [Power Clamp] or [GND Clamp] I-V curves**
  - pullup or pulldown “resistors”
  - static integrated terminators
  - static clamps
  - static bus hold circuits
- **Make sure you are using the appropriate rail for correct power and GND bounce simulation purposes**
  - use [Power Clamp] for pullup resistor
  - [GND Clamp] for pulldown resistor, etc.
- **Some additional post processing may be required to avoid double counting**

Modeling dynamic advanced features

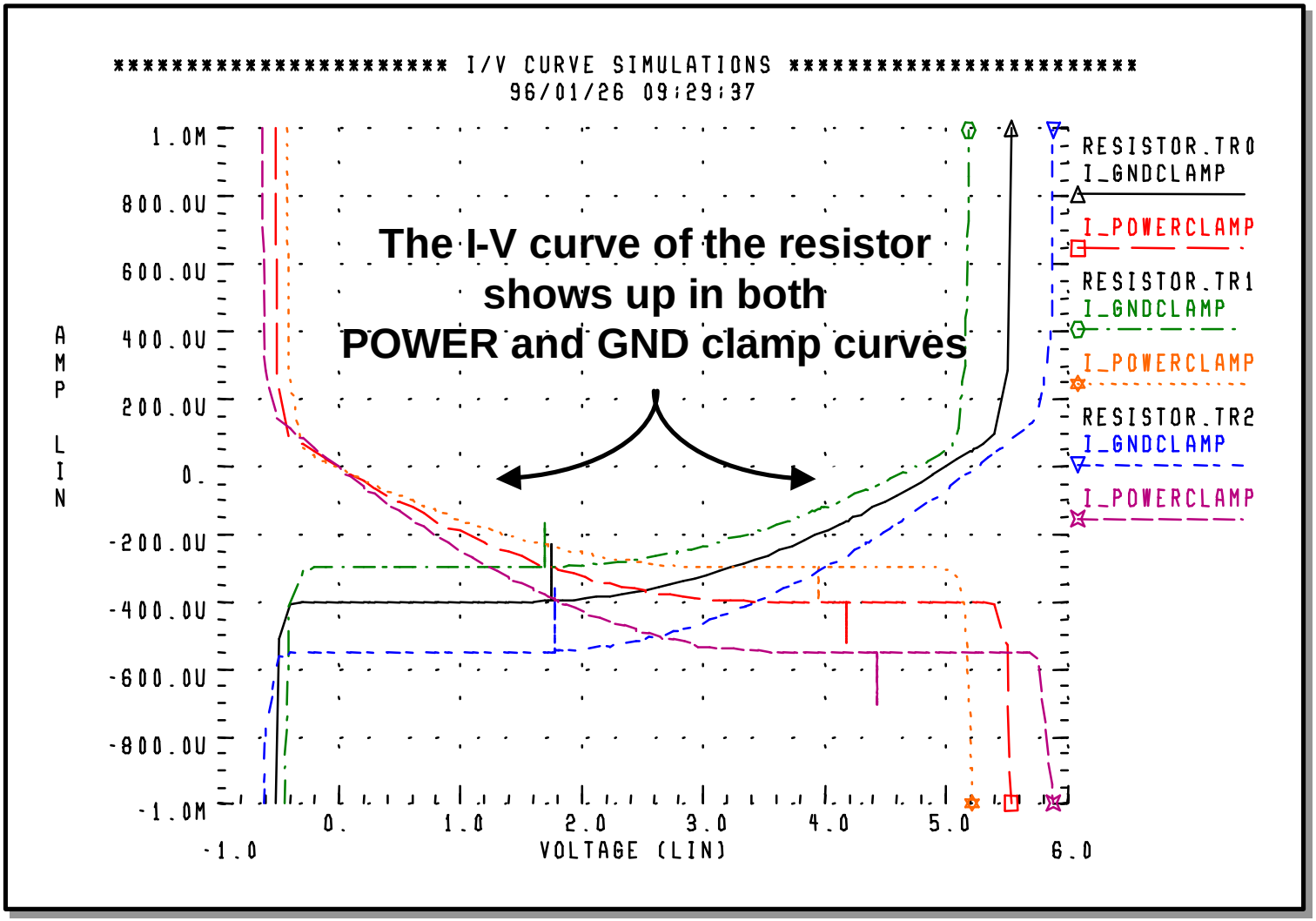
- **Use IBIS version 3.1 or 3.2 features**
  - keywords: [Driver Schedule], [Add Submodel], [Submodel], [Submodel Spec]
  - subparameters: Dynamic_clamp, Bus_hold
- **Detailed knowledge of circuit behavior is required**
- **Familiarity with buffer's SPICE netlist required**
- **May have to dissect or modify SPICE netlist to generate necessary data**
- **It may not be possible to make such models from simple or direct lab measurements**

Pullup resistor example

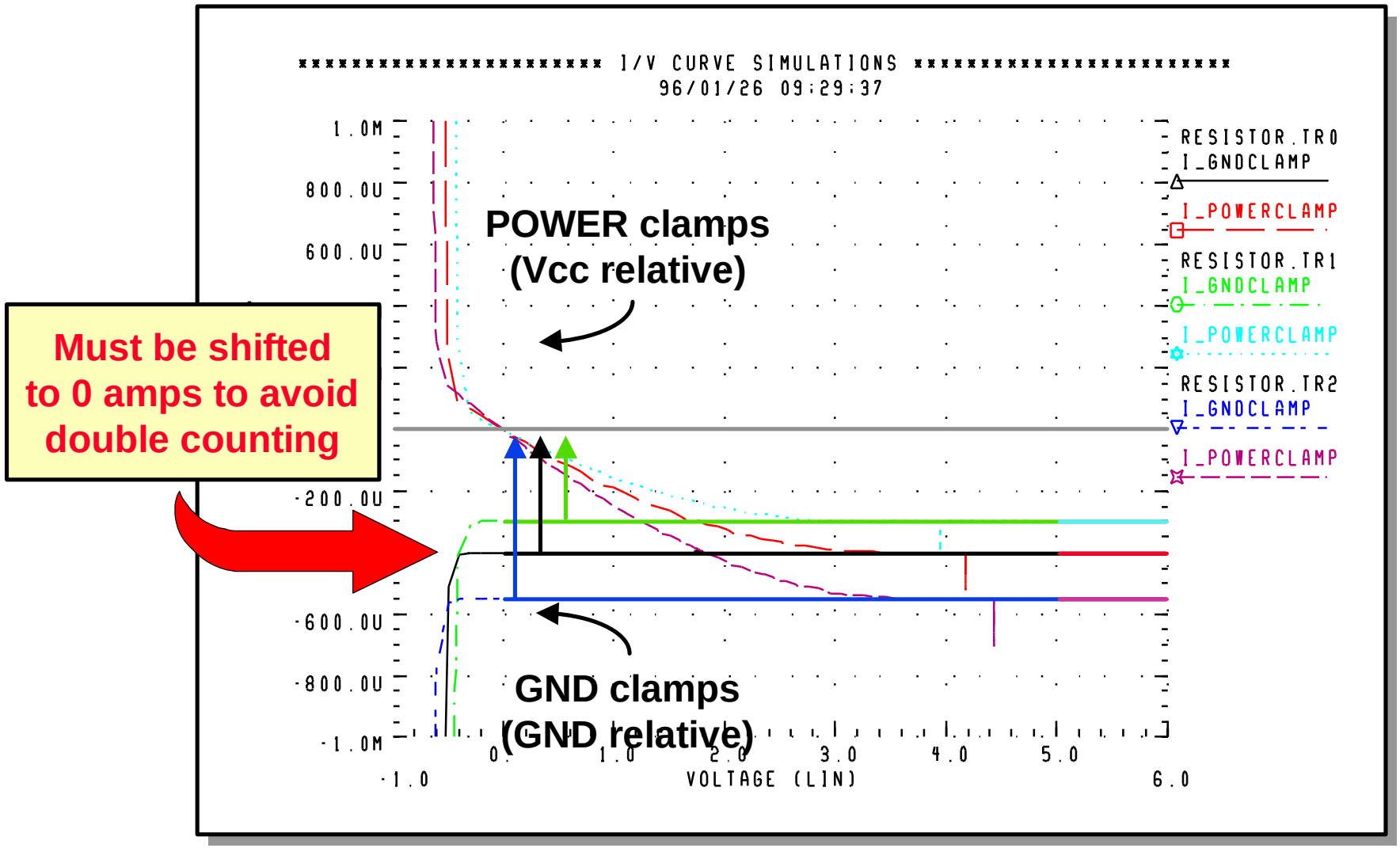


I-V curves of a 3-stated buffer with pullup R

Zooming in on I-V curves



Process to avoid double counting



Algorithm

- **Sweep device from $-V_{cc}$ to $2*V_{cc}$ twice: GND and V_{cc} relative**
- **Cut clamp curve which will include the resistor at V_{cc}**
- **Cut other clamp curve at 0V**
- **Normalize (shift) the clamp curve which will not include the resistor to zero current at 0V**
- **Extrapolate both clamp curves horizontally to $2*V_{cc}$**

Exercise #2a - buffer with pullup R

- **Look at the IO50V_R.INC file and notice the change**
  - there is a P-channel MOSFET element called MpR who's gate is connected to ground (it is always on)
- **Run HSPICE with the IO50V_R.SP file**
- **Open the IO50V_R.LIS file with IBIS Center**
- **Make an .MR1 file from it**
  - you may get a warning about having both pu and pd resistors in the buffer
  - you have to edit the .LIS file and make sure that the last column of the I-V curves have zeroes at 0 volt
  - press the ignore button when you get a warning about the V-t curve's starting point not matching the V_fixture voltage
- **Display the I-V curves of both the .LIS and .MR1 files and observe the differences**
  - you should see the GND clamp curve shifted up to 0 amp
  - the GND clamp curve should be flat above 0 volt
  - the POWER clamp should be extended horizontally above the power supply voltage (5 V)

On-die terminations

- **Series termination**
  - does not require any special work because it is described by the shape of the I-V curve
- **Parallel termination**
  - if the parallel termination is on all the time, use the method described for pullup/pulldown resistors
- **Switched parallel termination**
  - the parallel termination device is turned off while the opposite half of the buffer is driving
  - make a normal complementary model for the driver portion of the buffer
  - make a difference I-V curve for the terminator device and use the **[Add Submodel]** keyword in **non-driving** mode with the **[Submodel]** keyword's **dynamic_clamp** in static mode (without a pulse)

Switched parallel termination example

- **The buffer is a normal CMOS driver, but its pullup is ON in receive mode acting as a parallel terminator**

```
|*******************************************************************************
|
[Add Submodel]
| Submodel name         Mode
ParTerm                 Non-Driving
|
|*******************************************************************************
|
[Submodel]     ParTerm
Submodel_type  Dynamic_clamp
|
|*******************************************************************************
|
[POWER Clamp]
|
|       Voltage            I( typ)             I(min)             I(max)
|
     -1.79999995E+0     14.23263550E-3     17.10075140E-3     12.31312752E-3
|    . . .
|    . . .
|   The I-V curve table of the [Pullup] is repeated here, because the
|   terminator is actually the pullup left on in receive mode.
|    . . .
|    . . .
      3.59999990E+0    -44.34032738E-3    -44.32120919E-3    -48.62782359E-3
|
|*******************************************************************************
```

Differential buffer example

- **Make an independent model for each pin**
  - most differential drivers consist of two independent totem-pole drivers
  - since input currents are usually minimal, defining the clamp currents and differential threshold is sufficient for most differential receivers
- **Associate the drivers and/or receivers as differential using the [Diff Pin] keyword**
  - use the inv_pin, vdiff, tdelay_xxx subparameters to define polarity, differential threshold and differential skews

```
[Diff Pin]  inv_pin  vdiff  tdelay_typ tdelay_min tdelay_max
|
 3           4       150mV    -1ns       0ns       -2ns   | Input or I/O pair
 7           8         0V      1ns        NA         NA   | Output* pin pair
 9          10         NA       NA        NA         NA   | Output* pin pair
16          15       200mV     1ns     | Input or I/O pin pair
20          19         0V       NA     | Output* pin pair,  tdelay = 0
22          21         NA       NA     | Output*,  tdelay = 0
                                       | * Could be Input or I/O with  vdiff = 0
```

Exercise #2b - differential buffer

- **Open the IBIS file you made in Lab #1 and add a differential driver and/or receiver to it manually**
  - add two or more pins to the pin list under the [Pin] keyword
  - fill out the pin number, signal and buffer name columns
  - add a new keyword [Diff Pin]
  - fill in the headers next to the keyword
  - type the pin numbers of those additional pins you entered under the [Pin] keyword
  - fill in the values for the subparameters (use NAs if appropriate)

Exercise #2c - programmable buffer

- **Create a new IBIS model with a programmable buffer**
  - make a copy of one of the SPICE I/O or output models (.INC file) with a different file name
  - edit its netlist to make a different strength buffer from it (change the M=x multiplier for the output transistors)
  - run HSPICE to generate a new .LIS file and convert it to a .MR1 file with IBIS Center as before
  - add another pin to the pin list under the [Pin] keyword
  - put “progbuf_1” (or anything you like) for its buffer name
  - add the keyword [Model Selector] after the [Pin] section
  - give it the same name you used in the pin list above (progbuf_1)
  - type a list of model names (.MR1 file names) under the [Model Selector] keyword and some comments on their right
  - run IBIS Center’s “Make .IBS file” routine to create a new IBIS model from the new pin list

Exercise #2d - checking the model

- **Run the IBIS checker on the IBIS file(s) you made**
  - drive:\path\ibischk3 IBIS_file [> output_file]