Modeling on-die terminations in IBIS
(without double counting)

IBIS Summit at DAC 2003

Marriott Hotel, Anaheim, CA

June 5, 2003

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

Outline

- Summary of advanced buffer features
- General guidelines for making models for buffers with advanced features
- Static parallel termination
  - Algorithms to avoid double counting
- Switched parallel termination

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
- **Integrated terminators**
  - static transmission line termination (low impedance)
  - dynamic implementations designed to save power
- **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
- **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...**

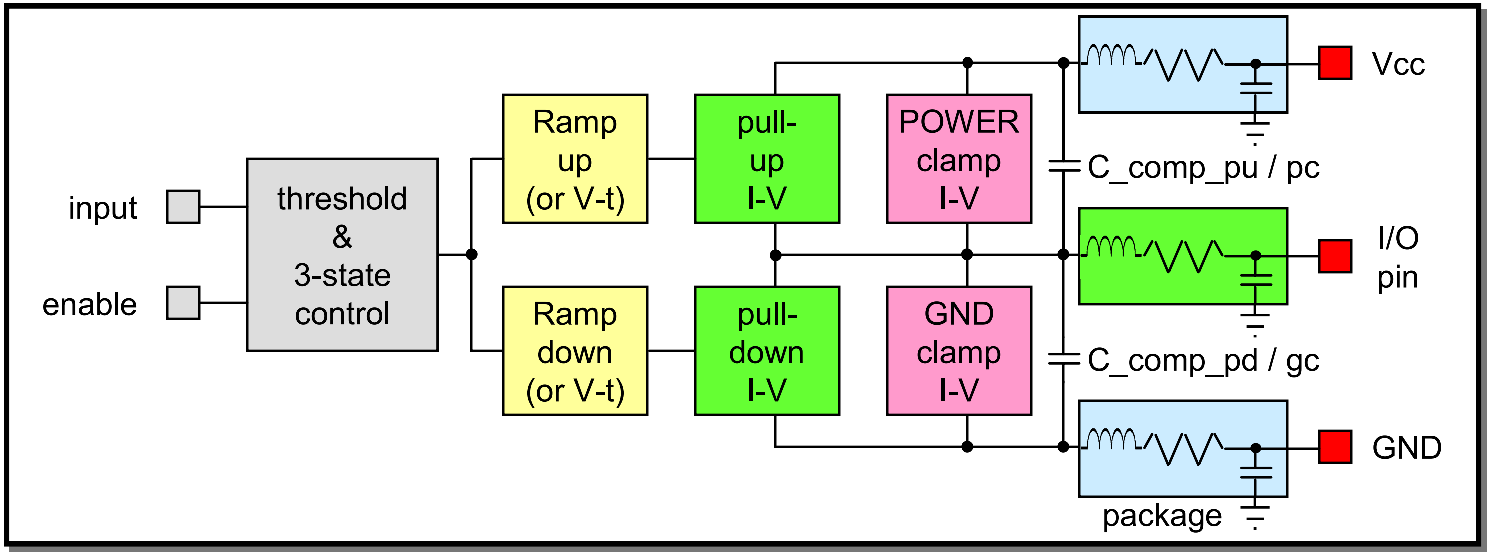
Modeling static advanced features

- **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, ESD circuits
  - 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 3.2 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 in separate steps**
- **It may not be possible to make such models from simple and/or direct lab measurements**

Block diaram of a CMOS IBIS model

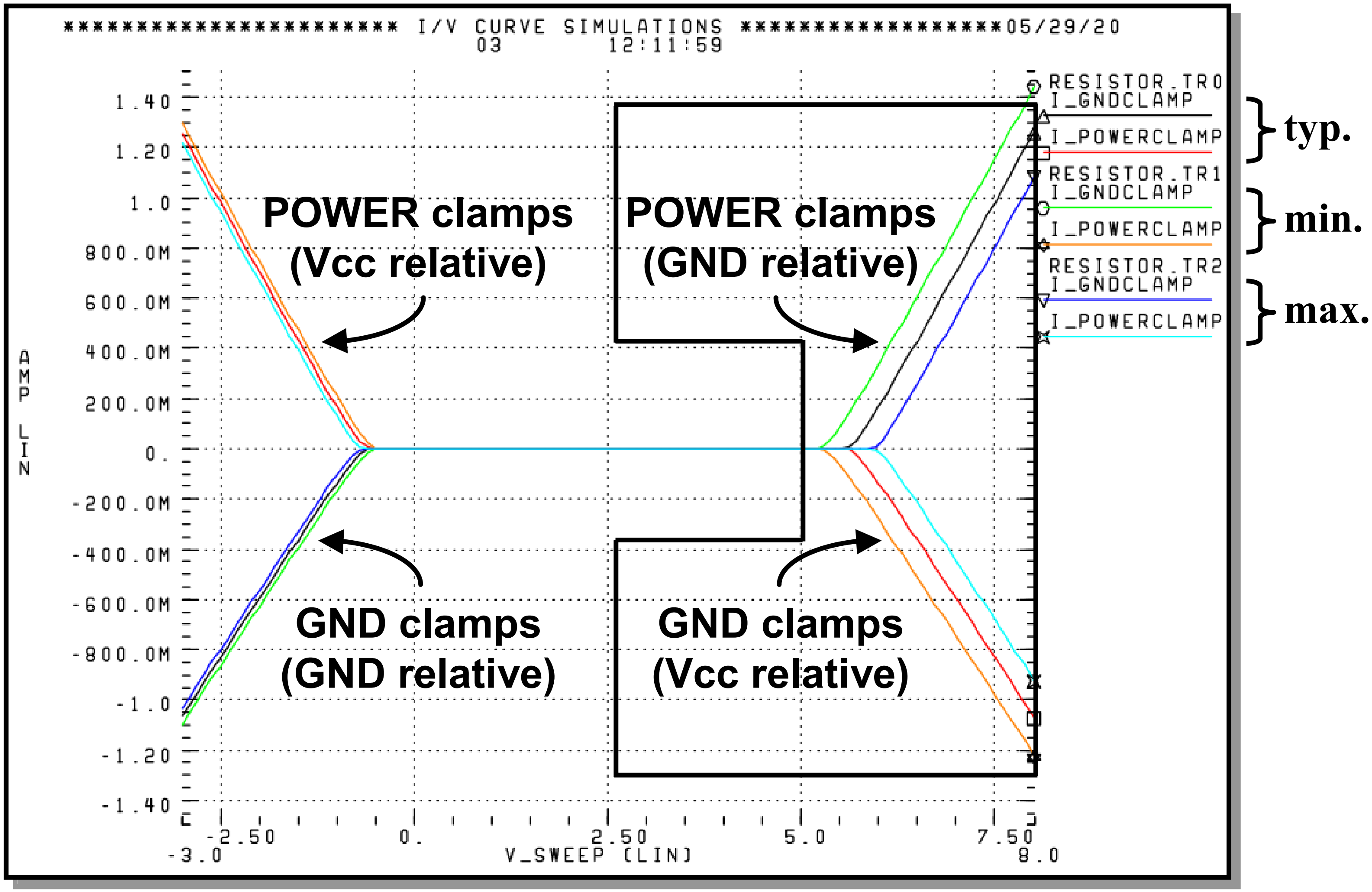


- Power/GND clamp IV curves are always ON
  - Use these for everything that is static
    - Parasitic diodes
    - ESD circuits
    - On-die terminations, etc…
- Pullup/Pulldown IV curves are switched ON/OFF by the Ramps/Vt curves
  - Use these for everything that is switched or dynamic
    - Drivers, “kickers”
    - Dynamic clamps
    - Dynamic on-die terminations, etc…

On-die terminations

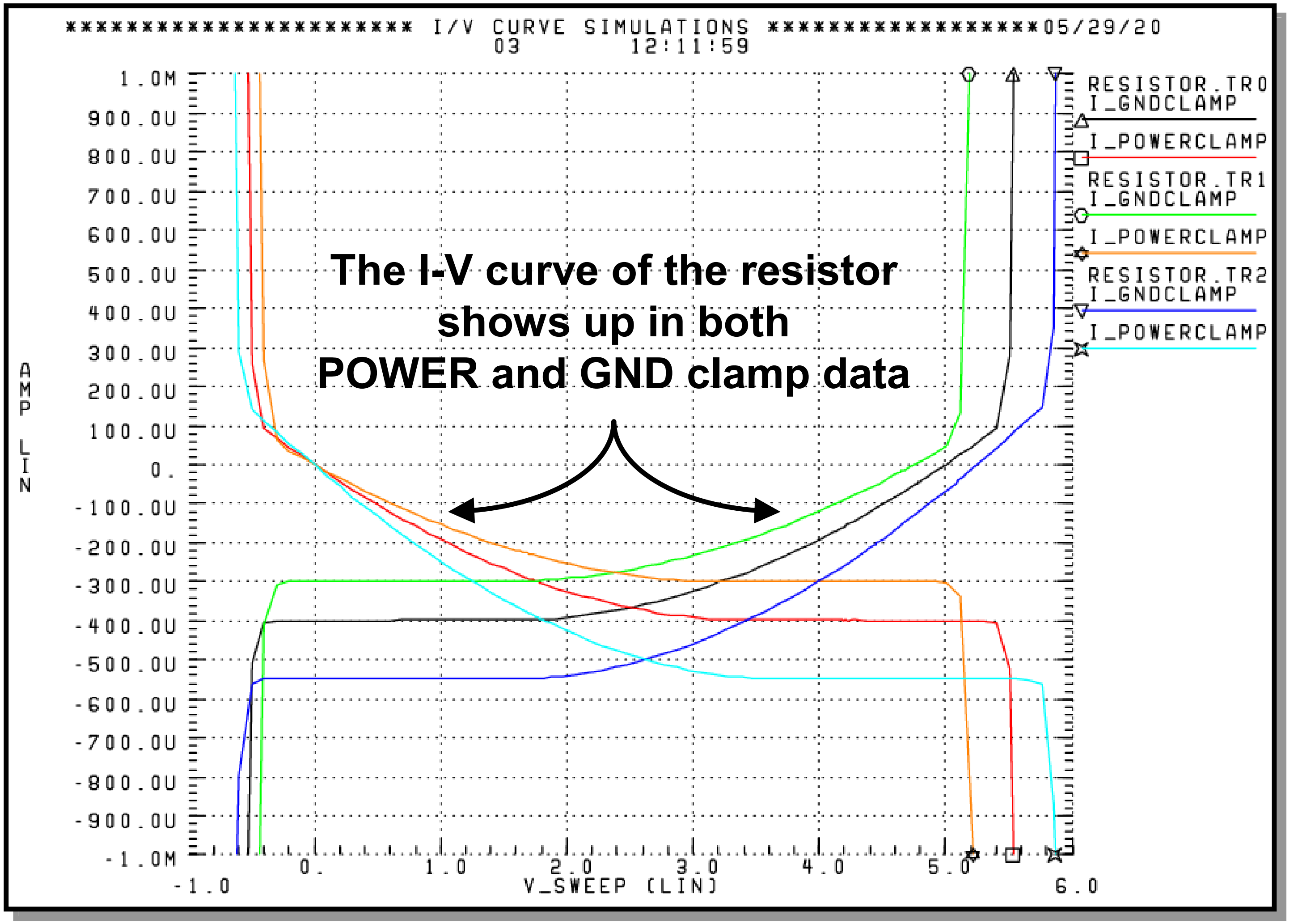
- **Series termination**
  - does not require any special work because it is described by the shape of the I-V curve
- **Parallel termination**
  - if the parallel termination is on all the time, use the method described for pullup/pulldown resistors
- **Switched parallel termination**
  - the parallel 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)

Pullup resistor example

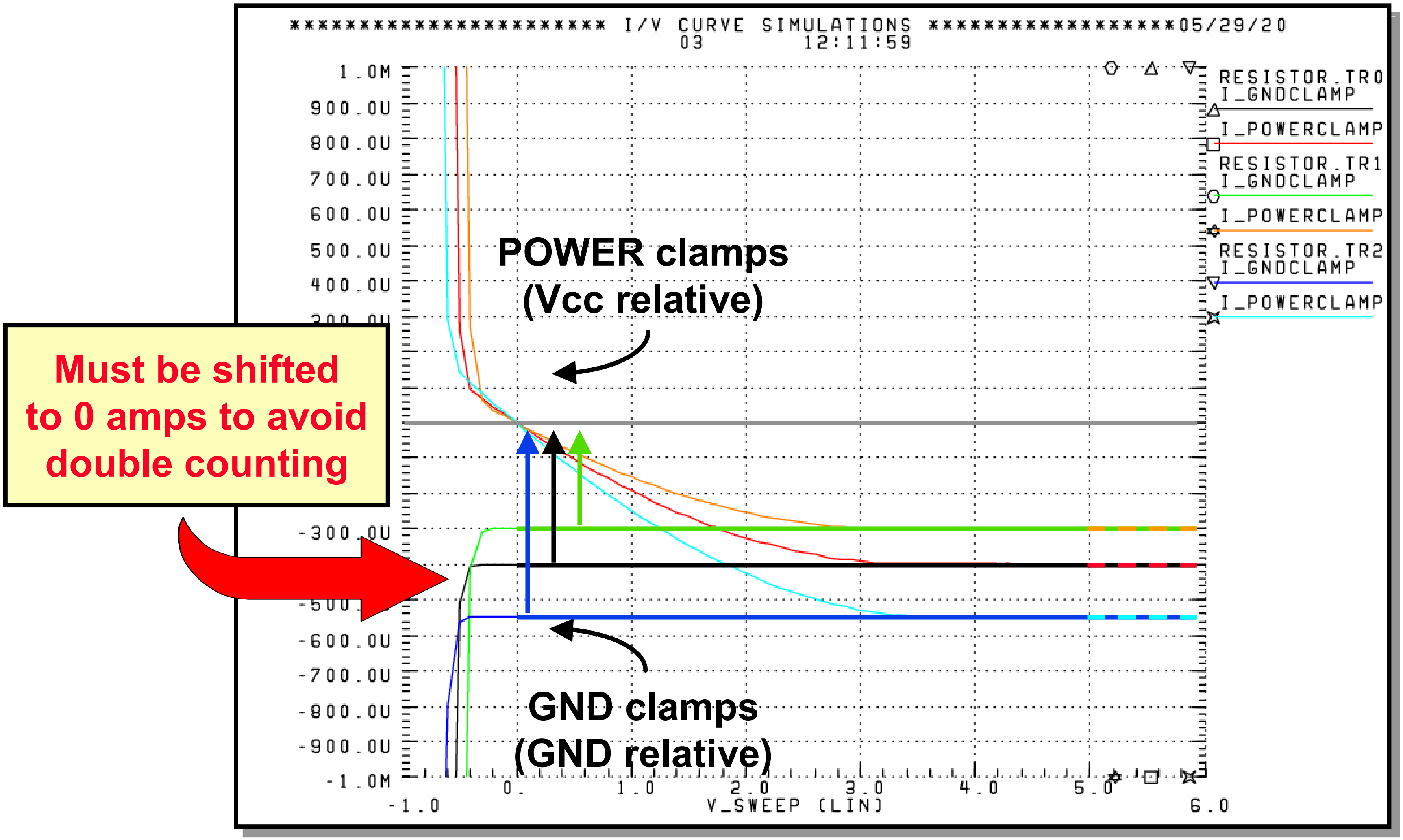


I-V curves of a 3-stated buffer with pullup R

Zooming in on I-V curves



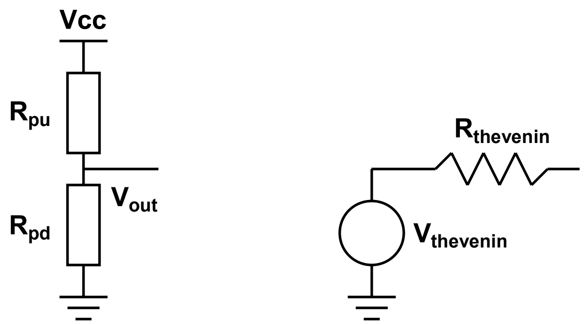
Algorithm in pictures



Algorithm in words

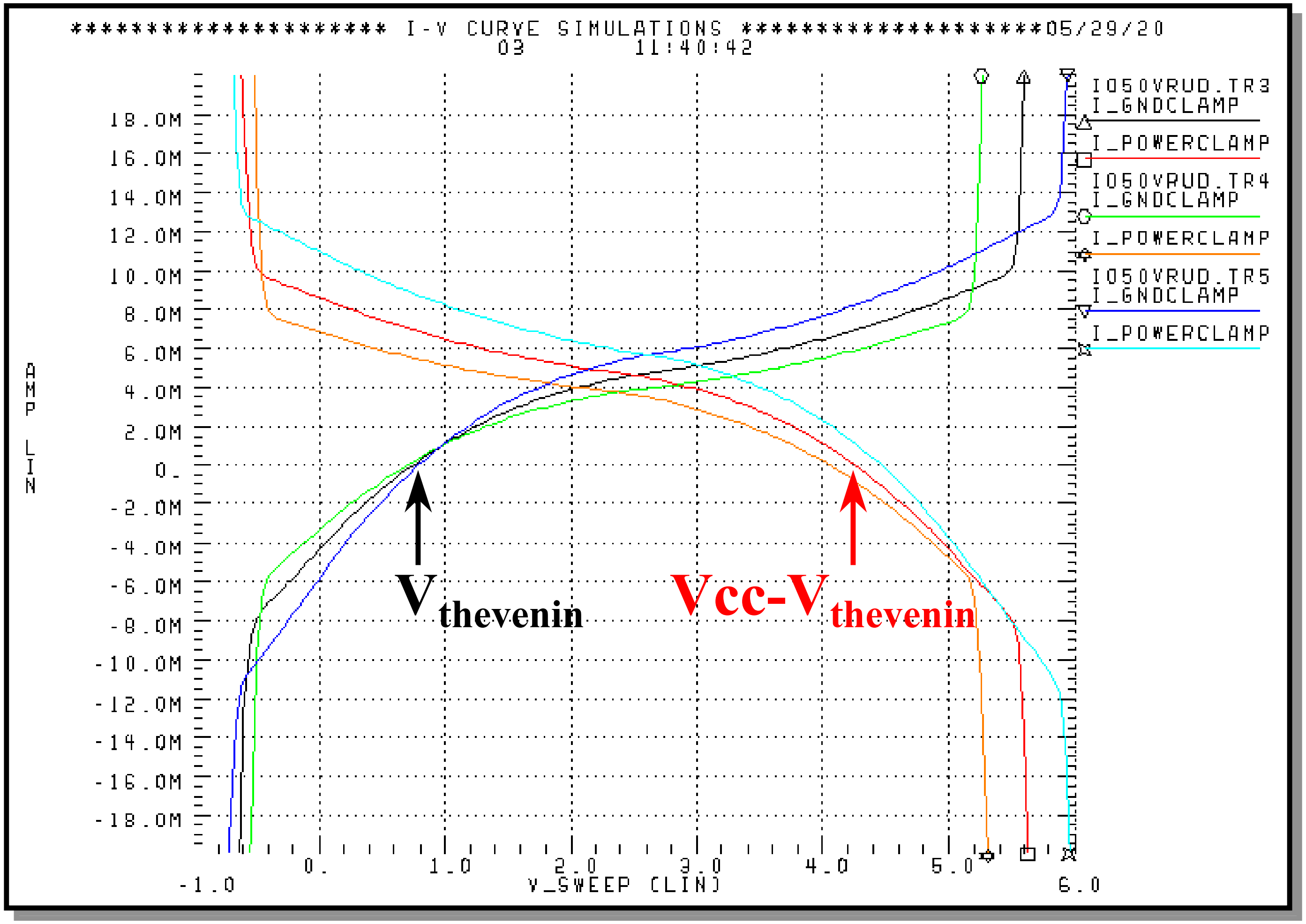
- Sweep device from $-V_{cc}$ to $2*V_{cc}$ twice: GND and V_{cc} relative
- Cut clamp curve which will include the resistor at V_{cc}
  - This can be automated by detecting which group of IV curves goes through the origin
- Cut other clamp curve at 0V
- Normalize (shift) the clamp curve which will not include the resistor to zero current at 0V
- Extrapolate both clamp curves horizontally to $2*V_{cc}$

Pullup and pulldown resistor example



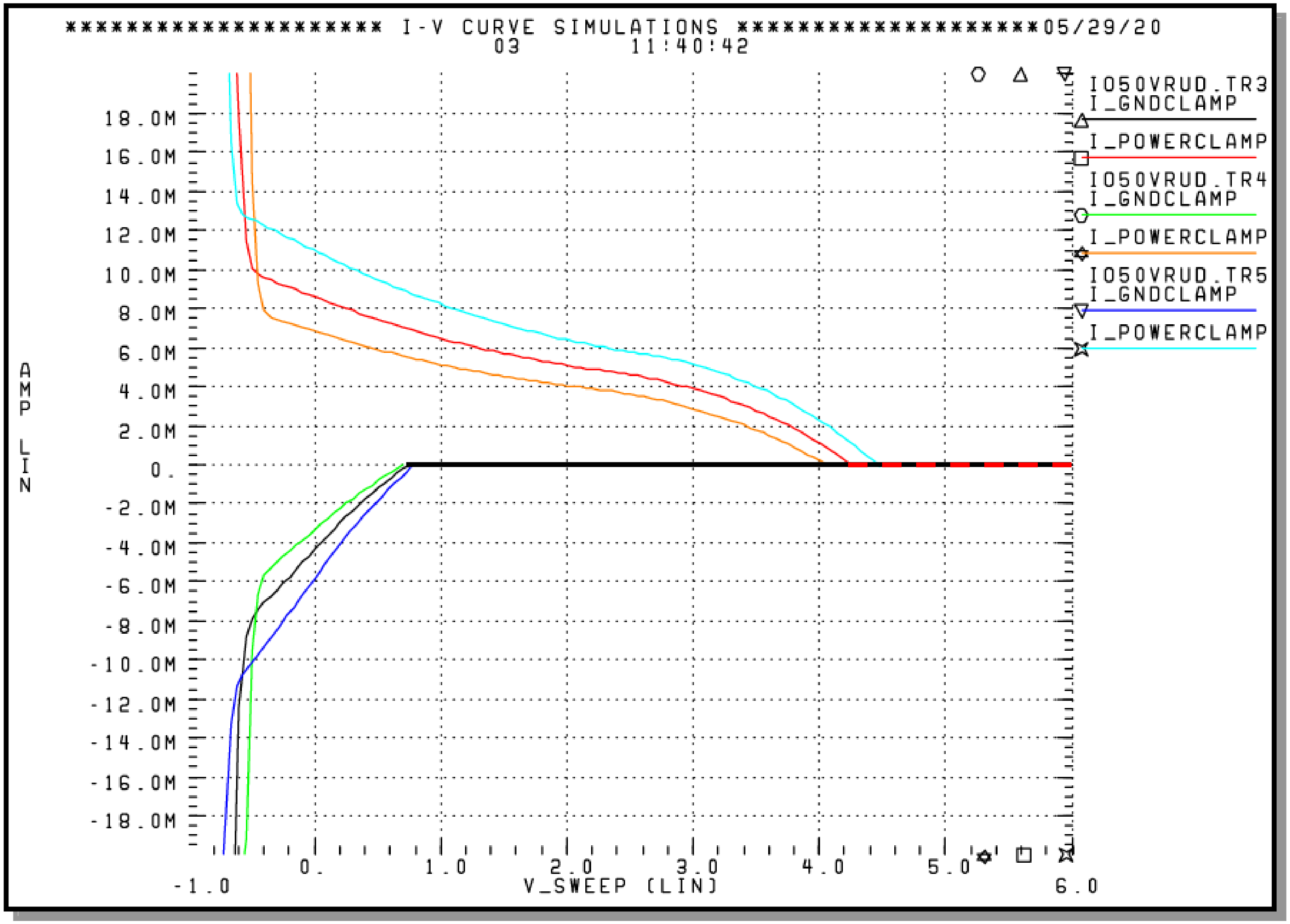
- Looking into the output pad we see $R_{thevenin}$
- It is not possible to separate $R_{thevenin}$ into R_{pu} and R_{pd} from a single measurement at the pad
- The algorithm described on the following pages is only a crude approximation, but it may be better than leaving everything in one IV curve
  - Useful for POWER and GND bounce simulations

IV curves of pu and pd R example



I-V curves of a 3-stated buffer with both pu and pd R

Algorithm in pictures



Algorithm in words

- Sweep device from $-V_{cc}$ to $2*V_{cc}$ twice: GND and V_{cc} relative
- Cut clamp curves where they reach zero current going left to right
- Extrapolate all clamp curves horizontally to $2*V_{cc}$

Switched parallel termination example

- **This 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
|
|**************************************************************************
```