Enabling Full Power-aware Bus Simulation with Non-IBIS Device Model - A Kit using IBIS [External Model]

Skipper Liang
Cadence Design Systems
Asian IBIS Summit, Taipei, Taiwan
November 13, 2015

Agenda

Introduction

Approach 1: Using IBIS [External Model]

Approach 2: Using an IBIS model to drive the SPICE netlist

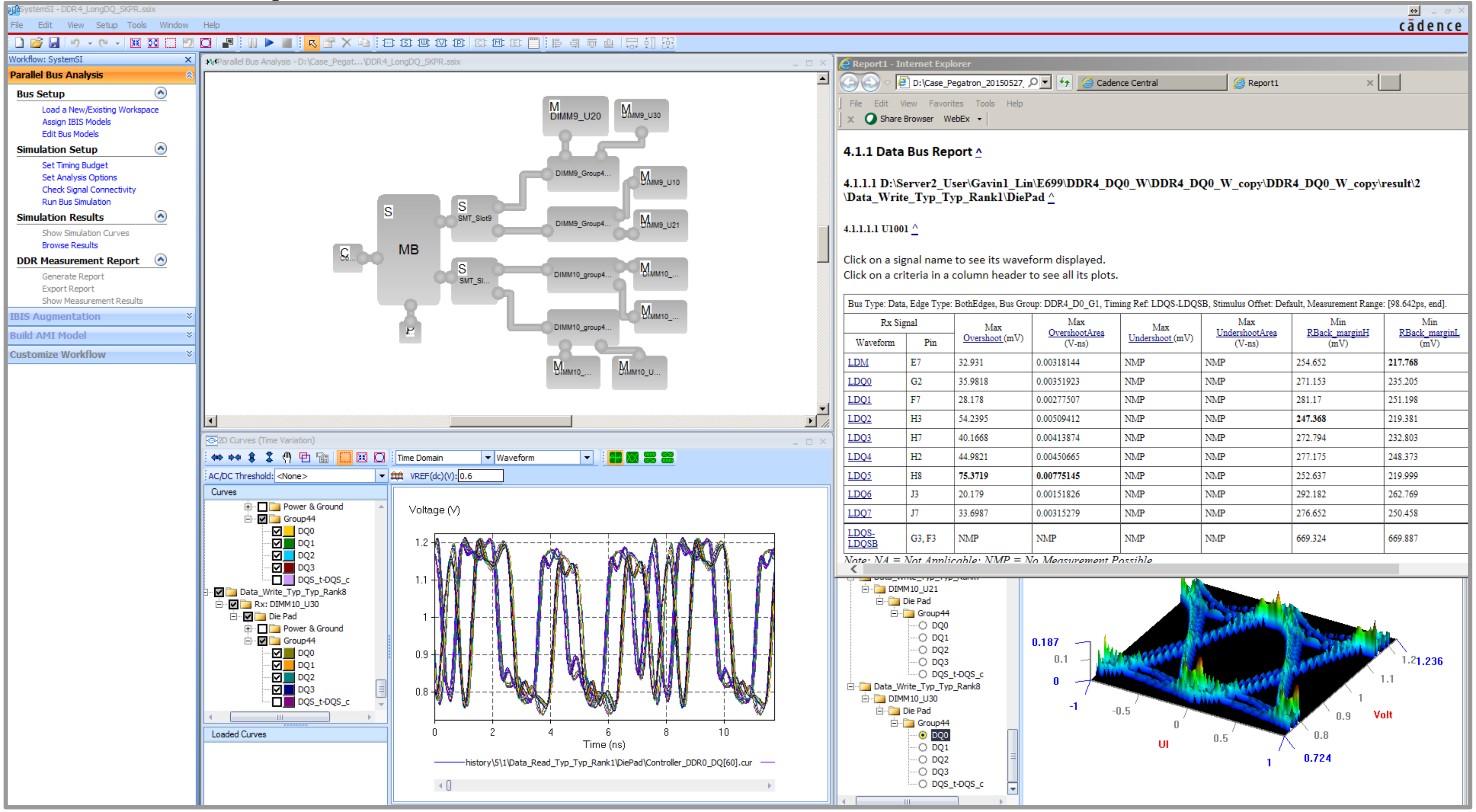
Solution: A dummy IBIS model using [External Model] and additional circuit

Summary

Introduction

Bus simulation

- It refers to an automatic process
  - which includes simulation, measurement and report generation against a Bus, especially the memory interface in an EDA tool



4.1.1 Data Bus Report ^

4.1.1.1 D:\Server2_User\Gavin1_Lin\E699\DDR4_DQ0_W\DDR4_DQ0_W_copy\DDR4_DQ0_W_copy\result\2\Data_Write_Typ_Typ_Rank1\DiePad ^

4.1.1.1.1 U1001 ^

Click on a signal name to see its waveform displayed.
Click on a criteria in a column header to see all its plots.

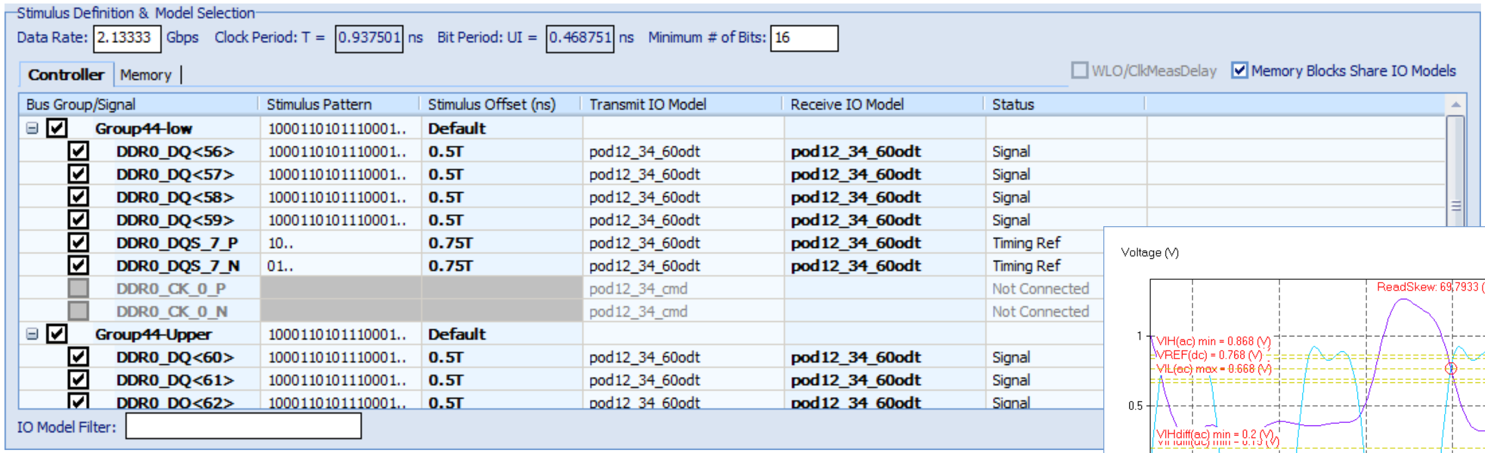
Bus Type: Data, Edge Type: BothEdges, Bus Group: DDR4_D0_G1, Timing Ref: LDQS-LDQSB, Stimulus Offset: Default, Measurement Range: [98.642ps, end].

Rx Signal		Max Overshoot (mV)	Max OvershootArea (V-ns)	Max Undershoot (mV)	Max UndershootArea (V-ns)	Min RBack_marginH (mV)	Min RBack_marginL (mV)
Waveform	Pin						
LDM	E7	32.931	0.00318144	NMP	NMP	254.652	**217.768**
LDQ0	G2	35.9818	0.00351923	NMP	NMP	271.153	235.205
LDQ1	F7	28.178	0.00277507	NMP	NMP	281.17	251.198
LDQ2	H3	54.2395	0.00509412	NMP	NMP	**247.368**	219.381
LDQ3	H7	40.1668	0.00413874	NMP	NMP	272.794	232.803
LDQ4	H2	44.9821	0.00450665	NMP	NMP	277.175	248.373
LDQ5	H8	**75.3719**	**0.00775145**	NMP	NMP	252.637	219.999
LDQ6	J3	20.179	0.00151826	NMP	NMP	292.182	262.769
LDQ7	J7	33.6987	0.00315279	NMP	NMP	276.652	250.458
LDQS-LDQSB	G3, F3	NMP	NMP	NMP	NMP	669.324	669.887

Note: NA = Not Applicable; NMP = No Measurement Possible

IBIS model

- Intrinsic format - the description of [Pin], [Diff Pin] and [Model Selector]
- IBIS makes Bus simulation possible in an EDA tool
  - easy to assign data groups with individual timing reference
  - Easy to change models and settings in batch mode



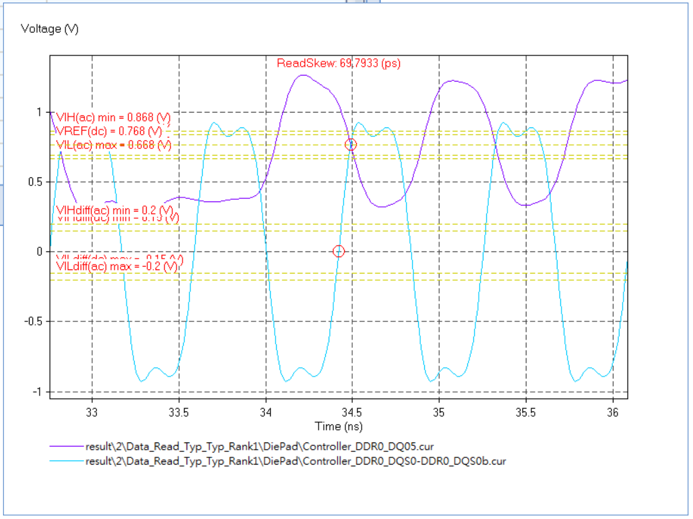
Stimulus Definition & Model Selection

Data Rate: 2.13333 Gbps Clock Period: T = 0.937501 ns Bit Period: UI = 0.468751 ns Minimum # of Bits: 16

Controller | Memory   ☐ WLO/ClkMeasDelay ☑ Memory Blocks Share IO Models

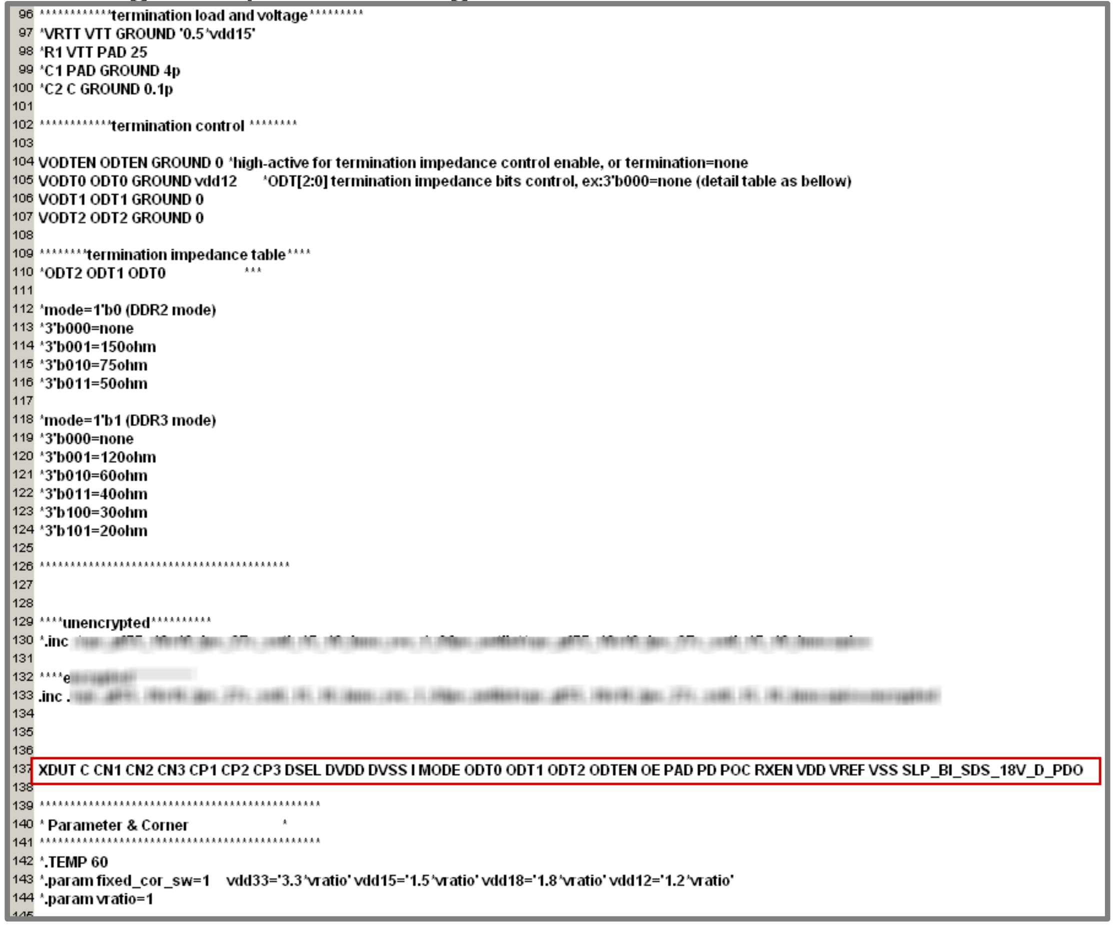
Bus Group/Signal	Stimulus Pattern	Stimulus Offset (ns)	Transmit IO Model	Receive IO Model	Status
☑ Group44-low	100011010110001..	Default			
☑ DDR0_DQ<56>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQ<57>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQ<58>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQ<59>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQS_7_P	10..	0.75T	pod12_34_60odt	pod12_34_60odt	Timing Ref
☑ DDR0_DQS_7_N	01..	0.75T	pod12_34_60odt	pod12_34_60odt	Timing Ref
☐ DDR0_CK_0_P			pod12_34_cmd		Not Connected
☐ DDR0_CK_0_N			pod12_34_cmd		Not Connected
☑ Group44-Upper	100011010110001..	Default			
☑ DDR0_DQ<60>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQ<61>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal
☑ DDR0_DQ<62>	100011010110001..	0.5T	pod12_34_60odt	pod12_34_60odt	Signal

IO Model Filter:



SPICE netlist

- Maximum freedom in the circuit format but hard for Bus simulation to be implemented
- Has detailed characteristics of circuits which can’t be fully described by IBIS model
  - Some IC designers prefer using SPICE netlist model than IBIS model



```
************termination load and voltage*********
*VRTT VTT GROUND '0.5*vdd15'
*R1 VTT PAD 25
*C1 PAD GROUND 4p
*C2 C GROUND 0.1p

************termination control ********

VODTEN ODTEN GROUND 0 *high-active for termination impedance control enable, or termination=none
VODT0 ODT0 GROUND vdd12     *ODT[2:0] termination impedance bits control, ex:3'b000=none (detail table as bellow)
VODT1 ODT1 GROUND 0
VODT2 ODT2 GROUND 0

********termination impedance table****
*ODT2 ODT1 ODT0                  ***

*mode=1'b0 (DDR2 mode)
*3'b000=none
*3'b001=150ohm
*3'b010=75ohm
*3'b011=50ohm

*mode=1'b1 (DDR3 mode)
*3'b000=none
*3'b001=120ohm
*3'b010=60ohm
*3'b011=40ohm
*3'b100=30ohm
*3'b101=20ohm

*******************************************


****unencrypted**********
*.inc [redacted]

****e[redacted]
.inc .[redacted]



XDUT C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I MODE ODT0 ODT1 ODT2 ODTEN OE PAD PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO

********************************************
* Parameter & Corner                       *
********************************************
*.TEMP 60
*.param fixed_cor_sw=1    vdd33='3.3*vratio' vdd15='1.5*vratio' vdd18='1.8*vratio' vdd12='1.2*vratio'
*.param vratio=1
```

IBIS External Model

- Allow a spice netlist to be called by an IBIS model, which means:

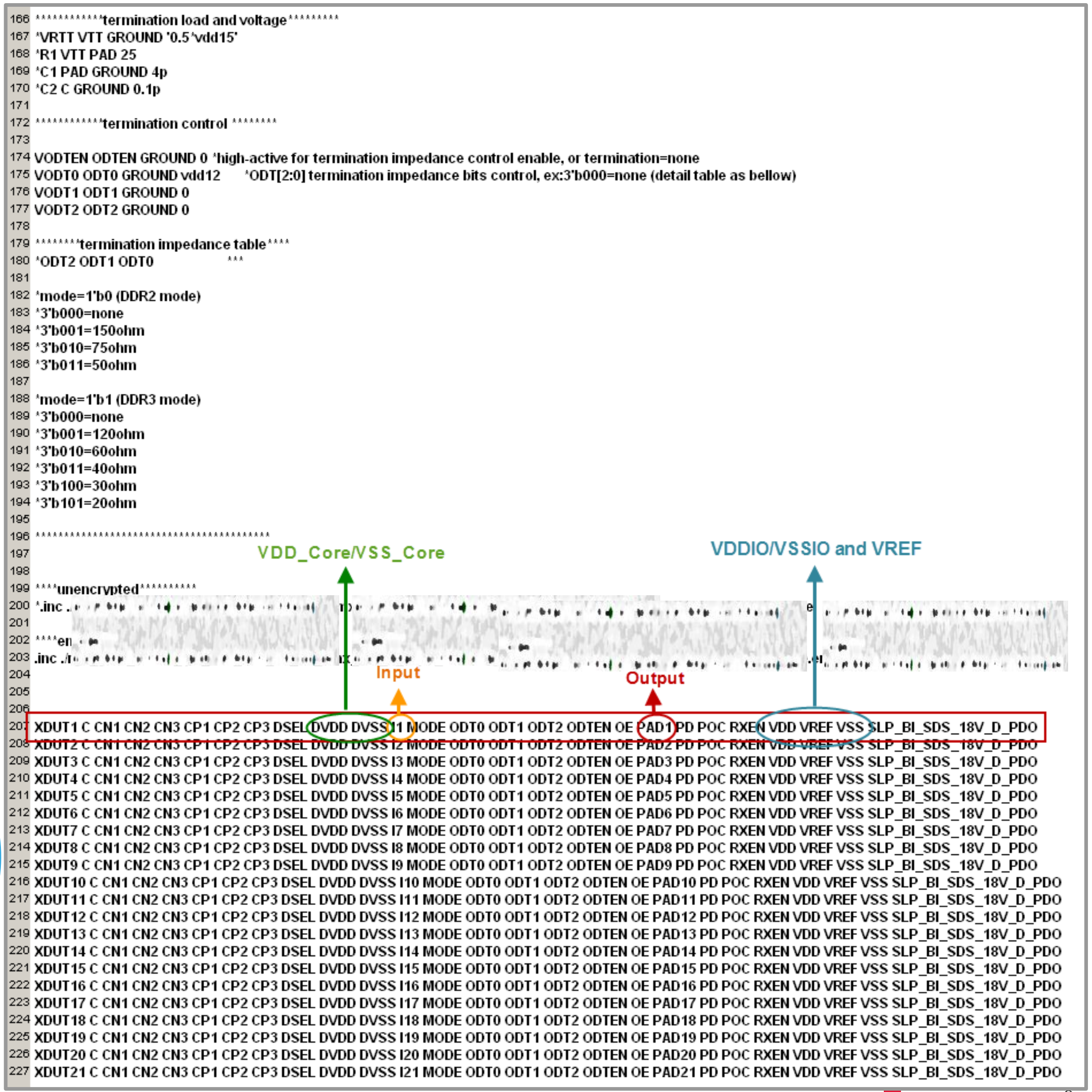
- With IBIS format, easy for Bus-Sim to be implemented
- Include detailed characteristics of circuit

```
[External Model]
Language SPICE
|
| Corner corner_name  file name        circuit_name (.subckt name)
Corner    Typ         buffer_typ.spi   buffer_io_typ
Corner    Min         buffer_min.spi   buffer_io_min
Corner    Max         buffer_max.spi   buffer_io_max
|
| Parameters - Not supported in SPICE
|
| Ports List of port names (in same order as in SPICE)
Ports A_signal my_drive my_enable my_receive my_ref
Ports A_puref A_pdref A_pcref A_gcref A_extref
|
| D_to_A d_port   port1     port2    vlow vhigh trise tfall corner_name
D_to_A   D_drive  my_drive   my_ref   0.0  3.3   0.5n  0.3n   Typ
D_to_A   D_enable my_enable  A_gcref  0.0  3.3   0.5n  0.3n   Typ
|
| A_to_D d_port    port1      port2       vlow  vhigh  corner_name
A_to_D   D_receive  my_receive  my_ref  0.8   2.0    Typ
|
| Note: A_signal might also be used instead of a user-defined interface port
| for measurements taken at the die pads
|
[End External Model]
```

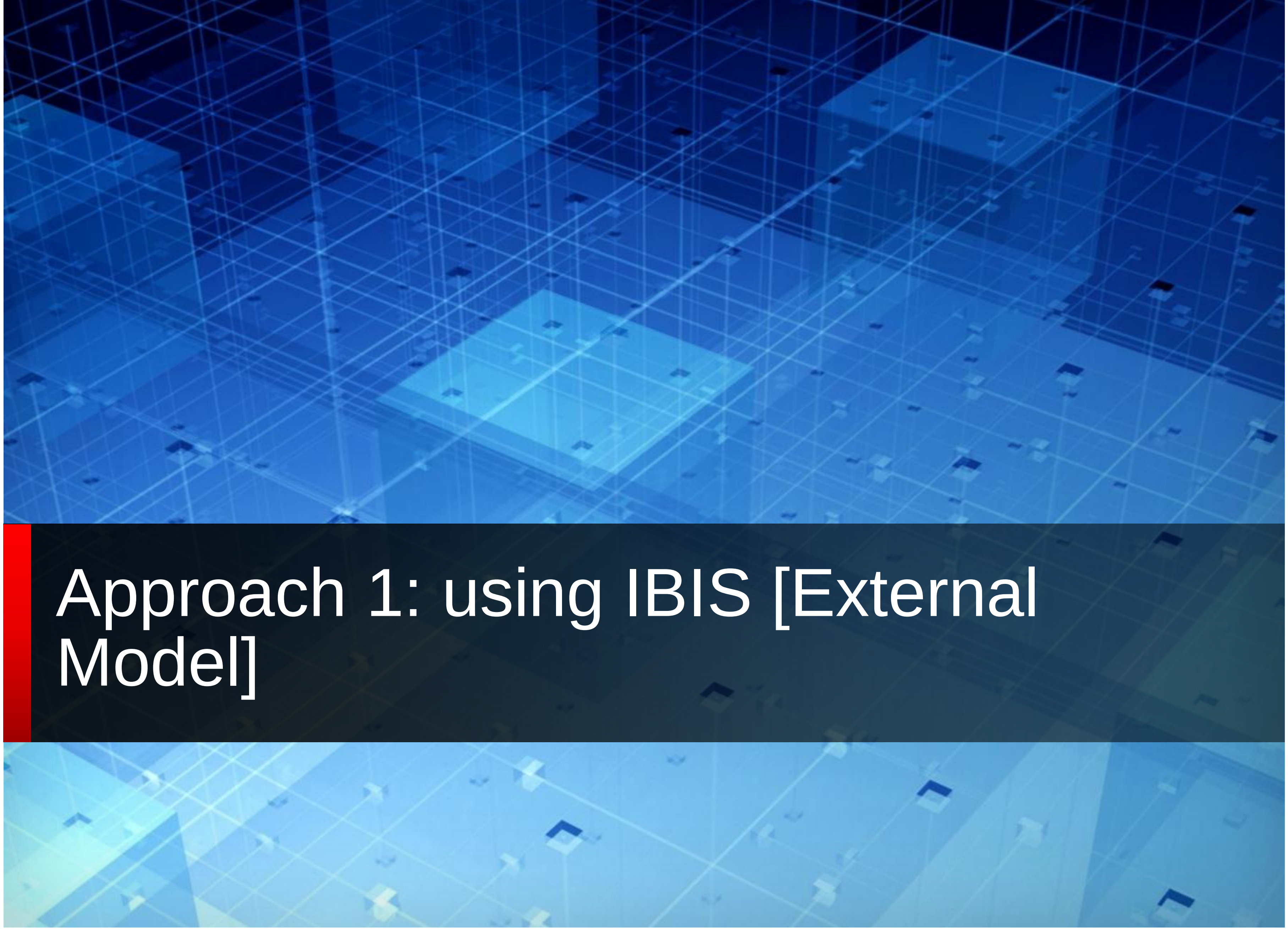
SPICE netlist of transistor based device model

- An example of SPICE netlist for a transistor based device model
- For power supply, there are:
  - VDD_Core/Vss_Core
  - VDDIO/VSSIO
  - VREF

Question: if we can use IBIS [External Model] to call this netlist to execute a fully power-aware Bus-Simulation?

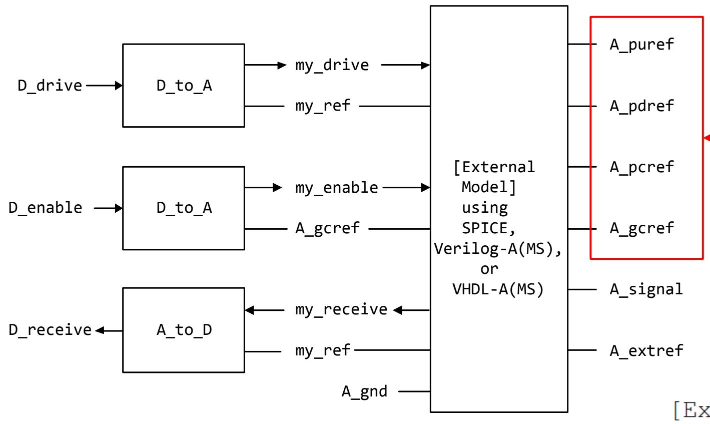


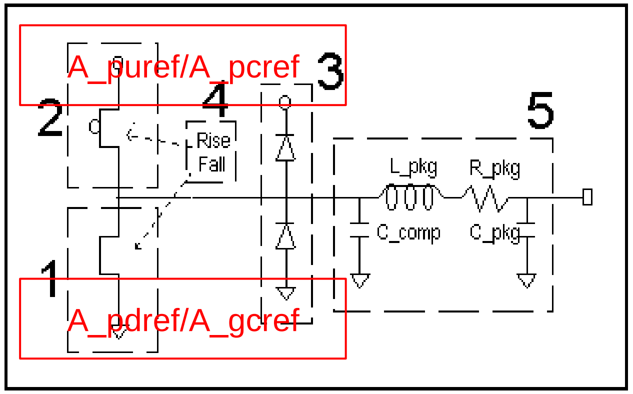
```
************'termination load and voltage*********
*VRTT VTT GROUND '0.5*vdd15'
*R1 VTT PAD 25
*C1 PAD GROUND 4p
*C2 C GROUND 0.1p

************'termination control ********

VODTEN ODTEN GROUND 0 'high-active for termination impedance control enable, or termination=none
VODT0 ODT0 GROUND vdd12      'ODT[2:0] termination impedance bits control, ex:3'b000=none (detail table as bellow)
VODT1 ODT1 GROUND 0
VODT2 ODT2 GROUND 0

********'termination impedance table****
*ODT2 ODT1 ODT0                     ***

*mode=1'b0 (DDR2 mode)
*3'b000=none
*3'b001=150ohm
*3'b010=75ohm
*3'b011=50ohm

*mode=1'b1 (DDR3 mode)
*3'b000=none
*3'b001=120ohm
*3'b010=60ohm
*3'b011=40ohm
*3'b100=30ohm
*3'b101=20ohm

****************************************



****unencrypted**********
*.inc .[redacted]

****en[redacted]
.inc ./[redacted]



XDUT1 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I1 MODE ODT0 ODT1 ODT2 ODTEN OE PAD1 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT2 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I2 MODE ODT0 ODT1 ODT2 ODTEN OE PAD2 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT3 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I3 MODE ODT0 ODT1 ODT2 ODTEN OE PAD3 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT4 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I4 MODE ODT0 ODT1 ODT2 ODTEN OE PAD4 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT5 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I5 MODE ODT0 ODT1 ODT2 ODTEN OE PAD5 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT6 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I6 MODE ODT0 ODT1 ODT2 ODTEN OE PAD6 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT7 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I7 MODE ODT0 ODT1 ODT2 ODTEN OE PAD7 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT8 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I8 MODE ODT0 ODT1 ODT2 ODTEN OE PAD8 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT9 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I9 MODE ODT0 ODT1 ODT2 ODTEN OE PAD9 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT10 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I10 MODE ODT0 ODT1 ODT2 ODTEN OE PAD10 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT11 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I11 MODE ODT0 ODT1 ODT2 ODTEN OE PAD11 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT12 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I12 MODE ODT0 ODT1 ODT2 ODTEN OE PAD12 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT13 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I13 MODE ODT0 ODT1 ODT2 ODTEN OE PAD13 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT14 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I14 MODE ODT0 ODT1 ODT2 ODTEN OE PAD14 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT15 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I15 MODE ODT0 ODT1 ODT2 ODTEN OE PAD15 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT16 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I16 MODE ODT0 ODT1 ODT2 ODTEN OE PAD16 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT17 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I17 MODE ODT0 ODT1 ODT2 ODTEN OE PAD17 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT18 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I18 MODE ODT0 ODT1 ODT2 ODTEN OE PAD18 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT19 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I19 MODE ODT0 ODT1 ODT2 ODTEN OE PAD19 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT20 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I20 MODE ODT0 ODT1 ODT2 ODTEN OE PAD20 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
XDUT21 C CN1 CN2 CN3 CP1 CP2 CP3 DSEL DVDD DVSS I21 MODE ODT0 ODT1 ODT2 ODTEN OE PAD21 PD POC RXEN VDD VREF VSS SLP_BI_SDS_18V_D_PDO
```



Approach 1: using IBIS [External Model]

Multi-power nets in [External Model] Section?





- Same as a normal IBIS model, an IBIS model with [External Model] can connect to only one PWR / GND pair, usually the VDDIO and VSSIO.
- What will we do for the following:
  1. VDD_Core/Vss_Core

     ➡ **A_puref, A_pdref, A_pcref, A_gcref**
  2. VDDIO/VSSIO

     ➡
  3. VREF

     ➡

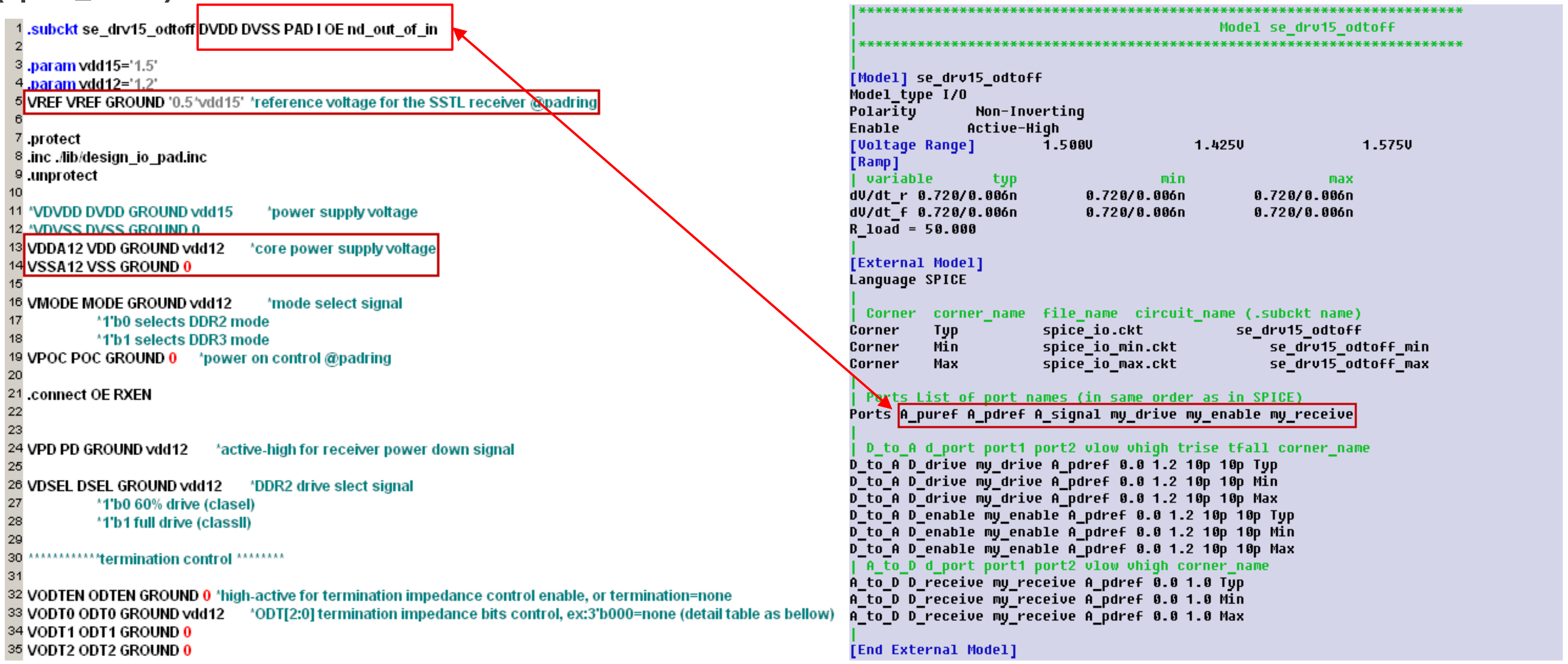
?

```
[External Model]
Language SPICE
|
| Corner corner_name  file_name        circuit_name (.subckt name)
Corner     Typ         buffer_typ.spi   buffer_io_typ
Corner     Min         buffer_min.spi   buffer_io_min
Corner     Max         buffer_max.spi   buffer_io_max
|
| Parameters - Not supported in SPICE
|
| Ports List of port names (in same order as in SPICE)
Ports A_signal my_drive my_enable my_receive my_ref
Ports A_puref A_pdref A_pcref A_gcref A_extref
|
| D_to_A d_port   port1     port2    vlow vhigh trise tfall corner_name
D_to_A   D_drive  my_drive   my_ref   0.0  3.3   0.5n  0.3n   Typ
D_to_A   D_enable my_enable  A_gcref  0.0  3.3   0.5n  0.3n   Typ
|
| A_to_D d_port    port1       port2      vlow  vhigh  corner_name
A_to_D   D_receive  my_receive  my_ref   0.8   2.0    Typ
|
| Note: A_signal might also be used instead of a user-defined interface port
| for measurements taken at the die pads
|
[End External Model]
```

cādence®

Multi-power nets in [External Model] Section? (Cont')

(spice_io.ckt)



```
.subckt se_drv15_odtoff DVDD DVSS PAD I OE nd_out_of_in

.param vdd15='1.5'
.param vdd12='1.2'
VREF VREF GROUND '0.5*vdd15'  *reference voltage for the SSTL receiver @padring

.protect
.inc ./lib/design_io_pad.inc
.unprotect

*VDVDD DVDD GROUND vdd15       *power supply voltage
*VDVSS DVSS GROUND 0
VDDA12 VDD GROUND vdd12      *core power supply voltage
VSSA12 VSS GROUND 0

VMODE MODE GROUND vdd12       *mode select signal
               *1'b0 selects DDR2 mode
               *1'b1 selects DDR3 mode
VPOC POC GROUND 0     *power on control @padring

.connect OE RXEN


VPD PD GROUND vdd12      *active-high for receiver power down signal

VDSEL DSEL GROUND vdd12      *DDR2 drive slect signal
               *1'b0 60% drive (clasel)
               *1'b1 full drive (classII)

***********termination control ********

VODTEN ODTEN GROUND 0 *high-active for termination impedance control enable, or termination=none
VODT0 ODT0 GROUND vdd12      *ODT[2:0] termination impedance bits control, ex:3'b000=none (detail table as bellow)
VODT1 ODT1 GROUND 0
VODT2 ODT2 GROUND 0
```

```
|*************************************************************************
|                                     Model se_drv15_odtoff
|*************************************************************************
|
[Model] se_drv15_odtoff
Model_type I/O
Polarity       Non-Inverting
Enable         Active-High
[Voltage Range]        1.500V             1.425V              1.575V
[Ramp]
| variable       typ              min              max
dV/dt_r 0.720/0.006n        0.720/0.006n        0.720/0.006n
dV/dt_f 0.720/0.006n        0.720/0.006n        0.720/0.006n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner  corner_name  file_name  circuit_name (.subckt name)
Corner    Typ          spice_io.ckt          se_drv15_odtoff
Corner    Min          spice_io_min.ckt          se_drv15_odtoff_min
Corner    Max          spice_io_max.ckt          se_drv15_odtoff_max
|
| Ports List of port names (in same order as in SPICE)
Ports A_puref A_pdref A_signal my_drive my_enable my_receive
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall corner_name
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Typ
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Min
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Max
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Typ
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Min
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Max
| A_to_D d_port port1 port2 vlow vhigh corner_name
A_to_D D_receive my_receive A_pdref 0.0 1.0 Typ
A_to_D D_receive my_receive A_pdref 0.0 1.0 Min
A_to_D D_receive my_receive A_pdref 0.0 1.0 Max
|
[End External Model]
```

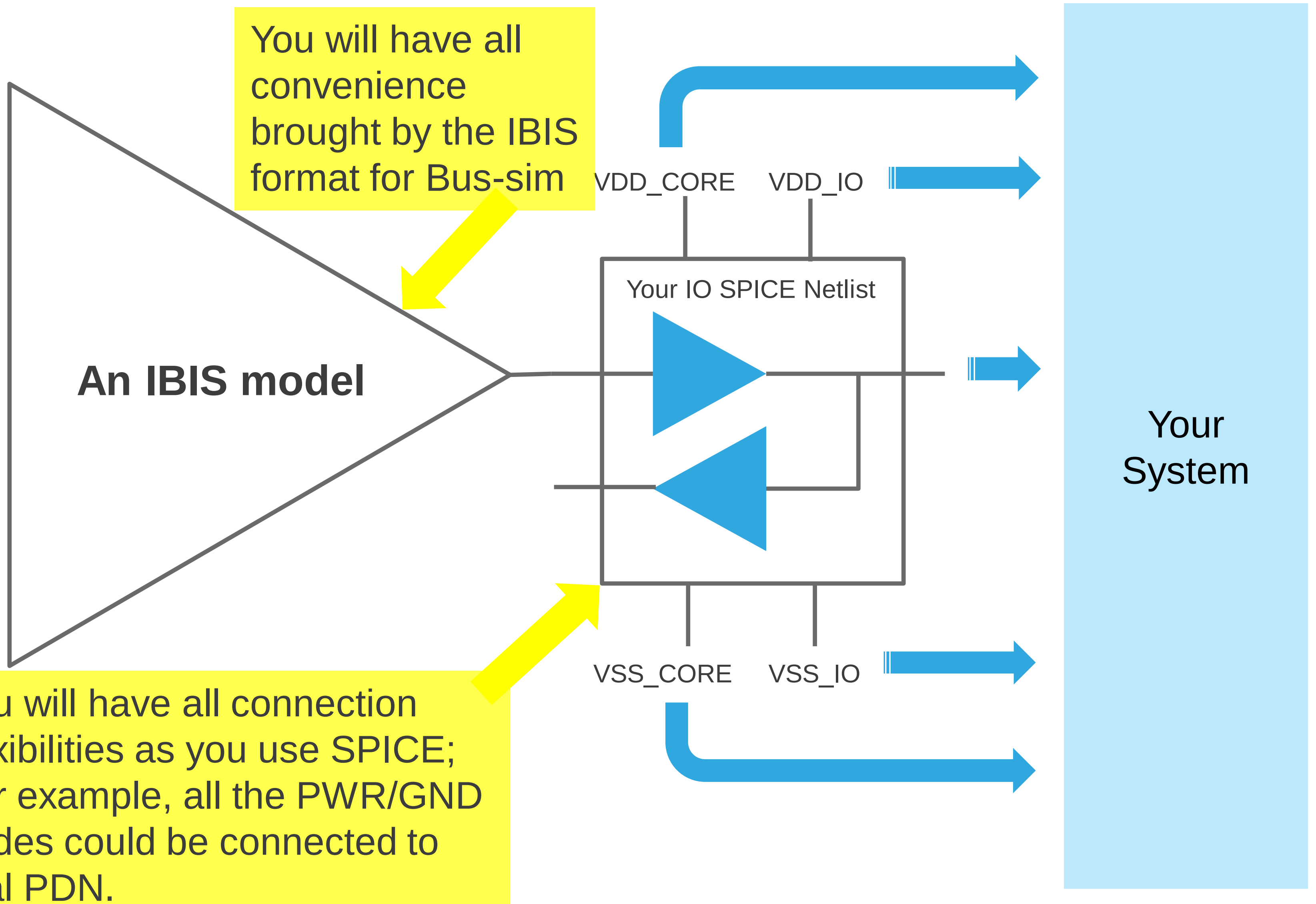
- Except the VDDIO and VSSIO, which will be connected to real power delivery network routing, the other PWR and GND will be connected to a "**IDEAL**" voltage source.

➡ **Not Fully Power-Aware**

Approach 2: Using an IBIS model to drive the SPICE netlist

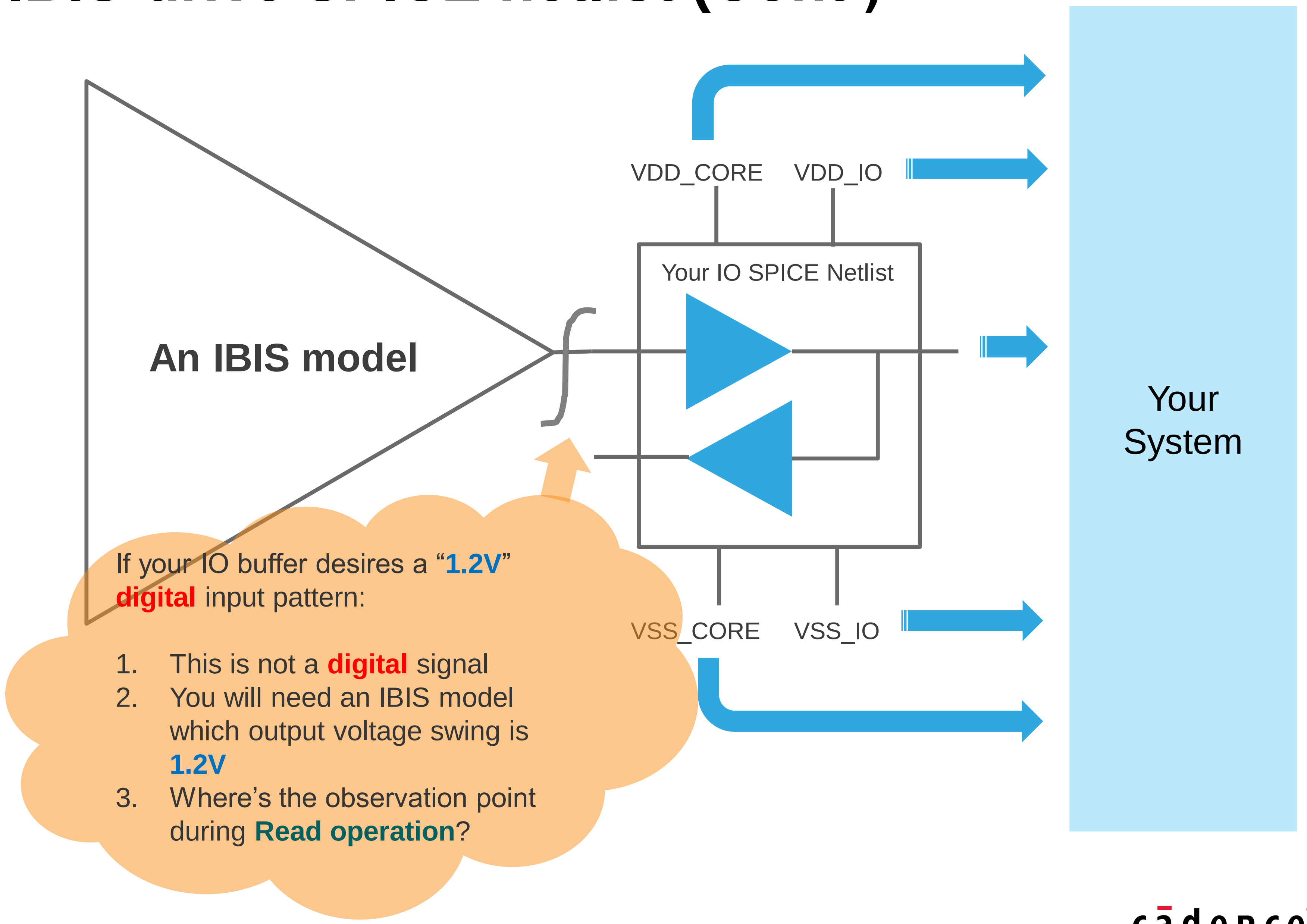
IBIS-drive SPICE netlist

You will have all convenience brought by the IBIS format for Bus-sim



You will have all connection flexibilities as you use SPICE; For example, all the PWR/GND nodes could be connected to real PDN.

IBIS-drive SPICE netlist (Cont')



If your IO buffer desires a "**1.2V**" **digital** input pattern:

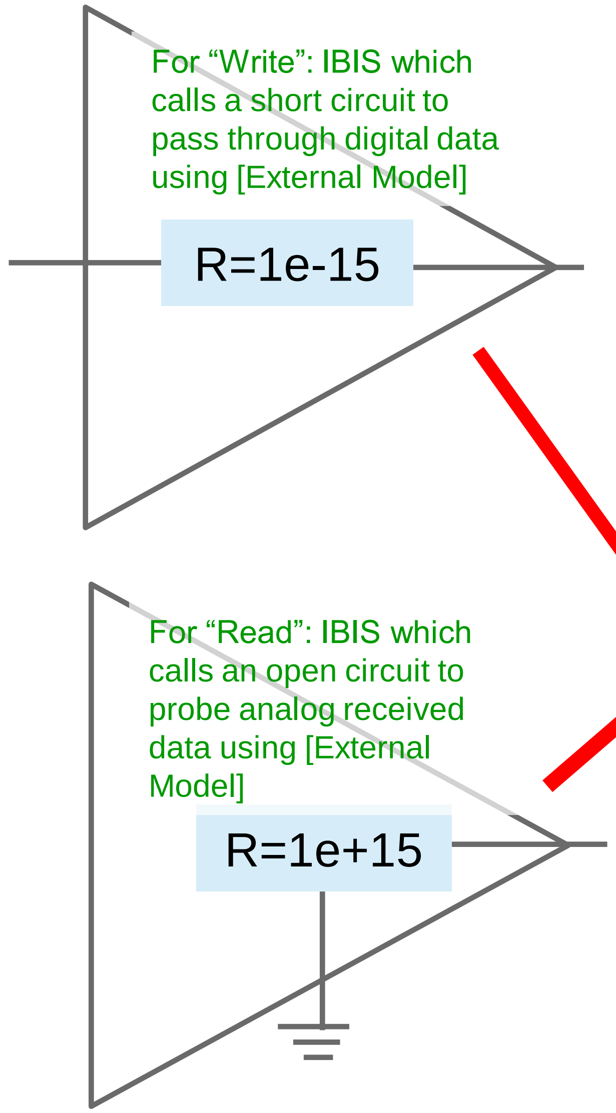
1. This is not a **digital** signal
2. You will need an IBIS model which output voltage swing is **1.2V**
3. Where's the observation point during **Read operation**?

cādence®

Solution: A dummy IBIS model using [External Model] and additional circuits

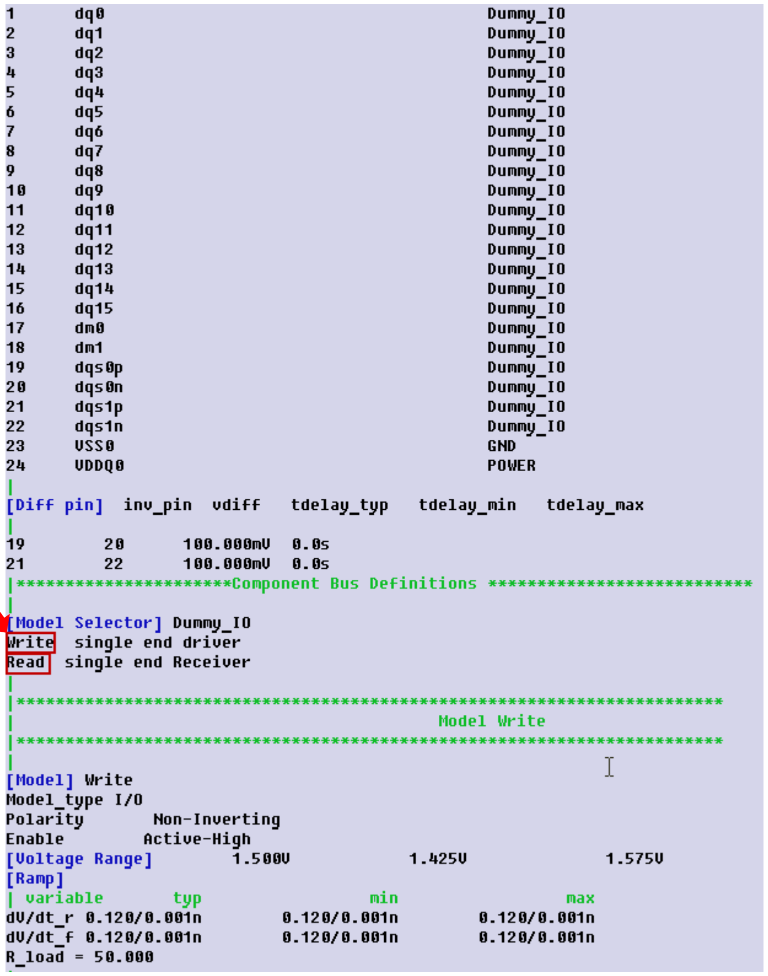
cadence®

A simple kit



For “Write”: IBIS which calls a short circuit to pass through digital data using [External Model]

For “Read”: IBIS which calls an open circuit to probe analog received data using [External Model]



```
1       dq0                                          Dummy_IO
2       dq1                                          Dummy_IO
3       dq2                                          Dummy_IO
4       dq3                                          Dummy_IO
5       dq4                                          Dummy_IO
6       dq5                                          Dummy_IO
7       dq6                                          Dummy_IO
8       dq7                                          Dummy_IO
9       dq8                                          Dummy_IO
10      dq9                                          Dummy_IO
11      dq10                                         Dummy_IO
12      dq11                                         Dummy_IO
13      dq12                                         Dummy_IO
14      dq13                                         Dummy_IO
15      dq14                                         Dummy_IO
16      dq15                                         Dummy_IO
17      dm0                                          Dummy_IO
18      dm1                                          Dummy_IO
19      dqs0p                                        Dummy_IO
20      dqs0n                                        Dummy_IO
21      dqs1p                                        Dummy_IO
22      dqs1n                                        Dummy_IO
23      VSS0                                         GND
24      VDDQ0                                        POWER
|
[Diff pin]  inv_pin  vdiff   tdelay_typ   tdelay_min   tdelay_max
|
19         20      100.000mV  0.0s
21         22      100.000mV  0.0s
|**********************Component Bus Definitions ***************************
|
[Model Selector] Dummy_IO
Write  single end driver
Read  single end Receiver
|
|************************************************************************
|                                             Model Write
|************************************************************************
|
[Model] Write
Model_type I/O
Polarity       Non-Inverting
Enable        Active-High
[Voltage Range]        1.500V            1.425V            1.575V
[Ramp]
| variable        typ                min                max
dV/dt_r 0.120/0.001n       0.120/0.001n       0.120/0.001n
dV/dt_f 0.120/0.001n       0.120/0.001n       0.120/0.001n
R_load = 50.000
```

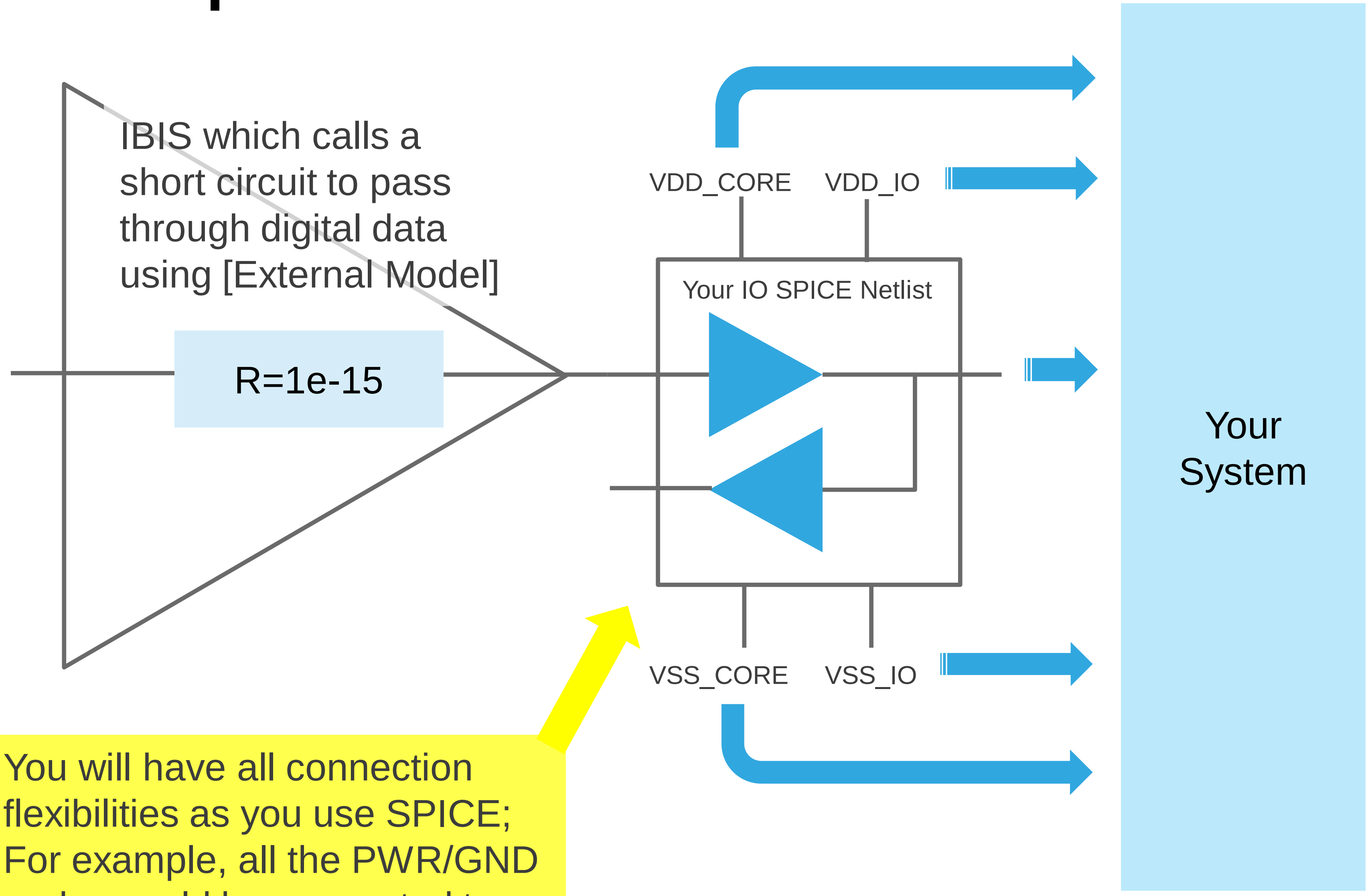
through.ibs

write.ckt

read.ckt

cādence®

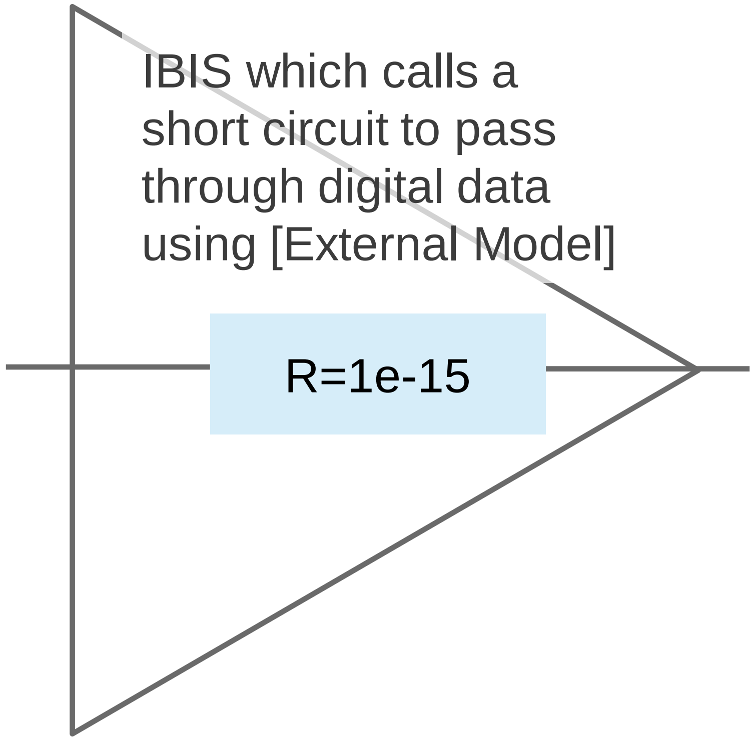
Write Operation

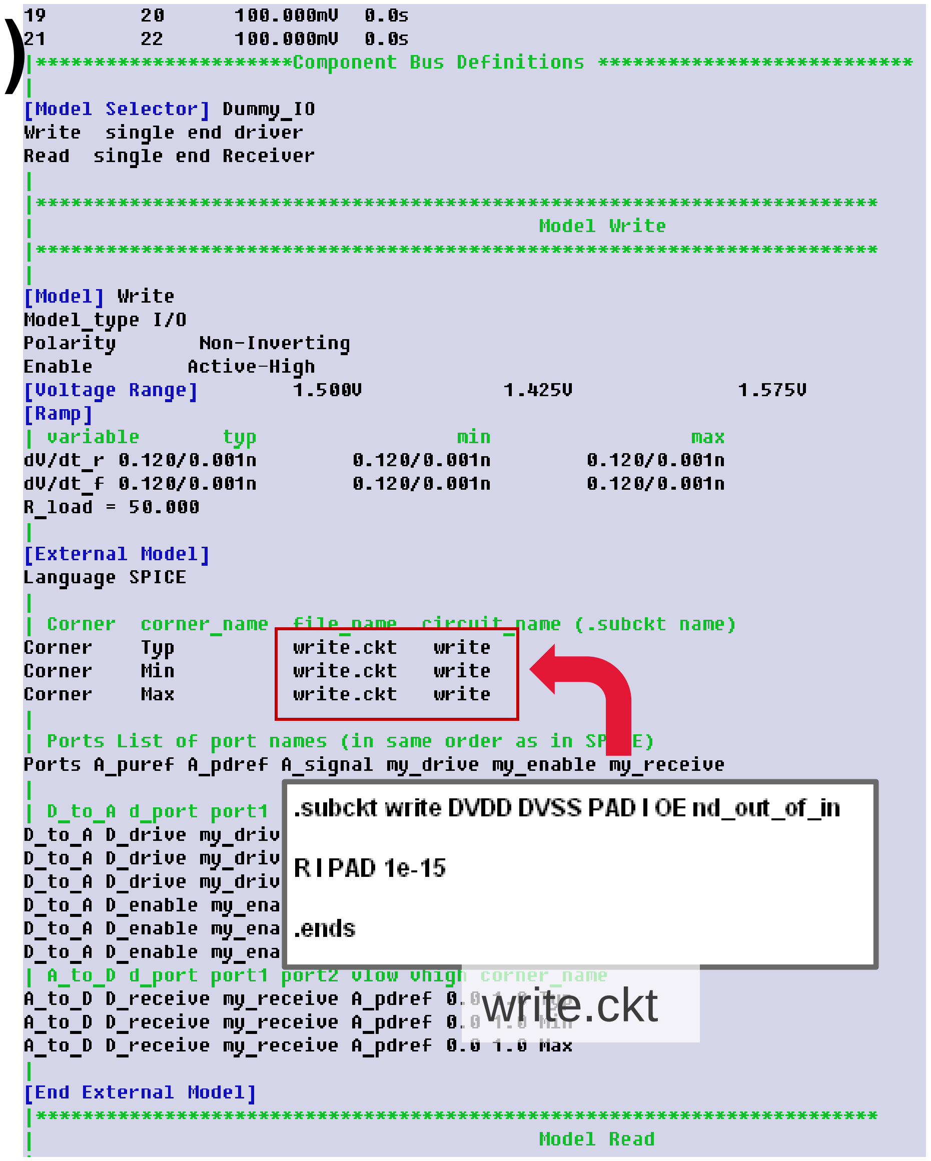

IBIS which calls a short circuit to pass through digital data using [External Model]
R=1e-15
VDD_CORE
VDD_IO
Your IO SPICE Netlist
VSS_CORE
VSS_IO
Your System

You will have all connection flexibilities as you use SPICE; For example, all the PWR/GND nodes could be connected to real PDN.

Write Operation (cont')

IBIS which calls a short circuit to pass through digital data using [External Model]





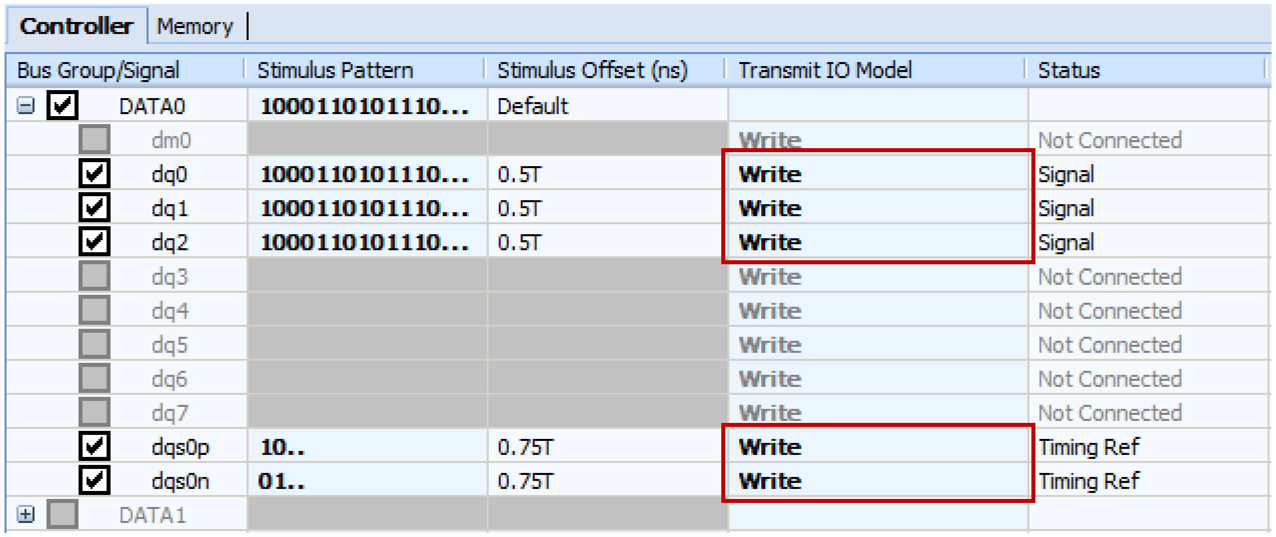
```
19         20       100.000mV   0.0s
21         22       100.000mV   0.0s
|**********************Component Bus Definitions ***************************
|
[Model Selector] Dummy_IO
Write  single end driver
Read  single end Receiver
|
|*************************************************************************
|                                          Model Write
|*************************************************************************
|
[Model] Write
Model_type I/O
Polarity       Non-Inverting
Enable        Active-High
[Voltage Range]        1.500V            1.425V              1.575V
[Ramp]
| variable       typ                  min                  max
dV/dt_r 0.120/0.001n         0.120/0.001n         0.120/0.001n
dV/dt_f 0.120/0.001n         0.120/0.001n         0.120/0.001n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner  corner_name  file_name  circuit_name (.subckt name)
Corner    Typ          write.ckt    write
Corner    Min          write.ckt    write
Corner    Max          write.ckt    write
|
| Ports List of port names (in same order as in SP  E)
Ports A_puref A_pdref A_signal my_drive my_enable my_receive
|
| D_to_A d_port port1
D_to_A D_drive my_driv
D_to_A D_drive my_driv
D_to_A D_drive my_driv
D_to_A D_enable my_ena
D_to_A D_enable my_ena
D_to_A D_enable my_ena
| A_to_D d_port port1 port2 vlow vhigh corner_name
A_to_D D_receive my_receive A_pdref 0.0 1.0 Typ
A_to_D D_receive my_receive A_pdref 0.0 1.0 Min
A_to_D D_receive my_receive A_pdref 0.0 1.0 Max
|
[End External Model]
|*************************************************************************
|                                          Model Read
```

```
.subckt write DVDD DVSS PAD I OE nd_out_of_in

R I PAD 1e-15

.ends
```

write.ckt

Write Operation (cont')

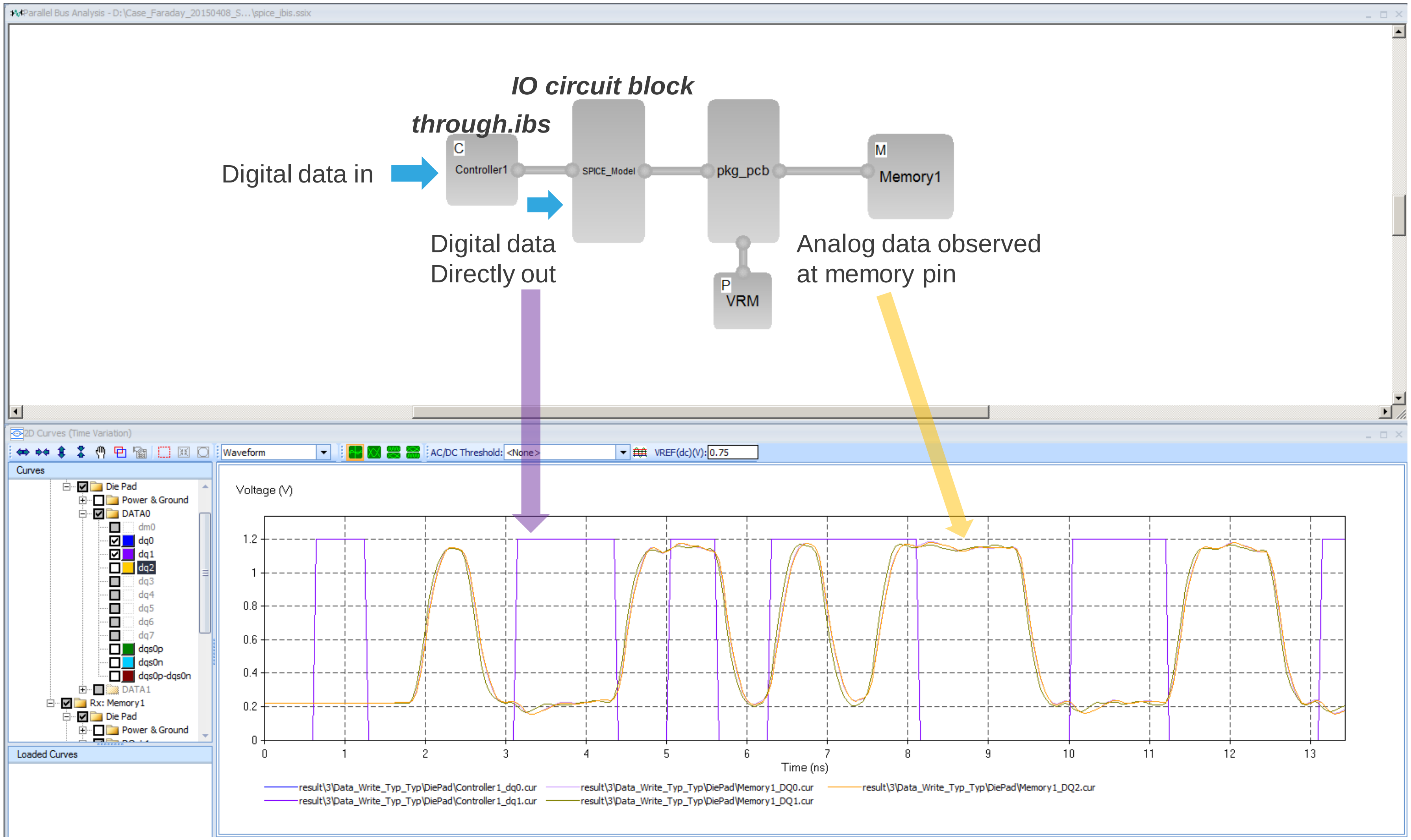
- In a bus simulation, user can easily switch the model of the Controller's buffer to "Write"



Controller | Memory

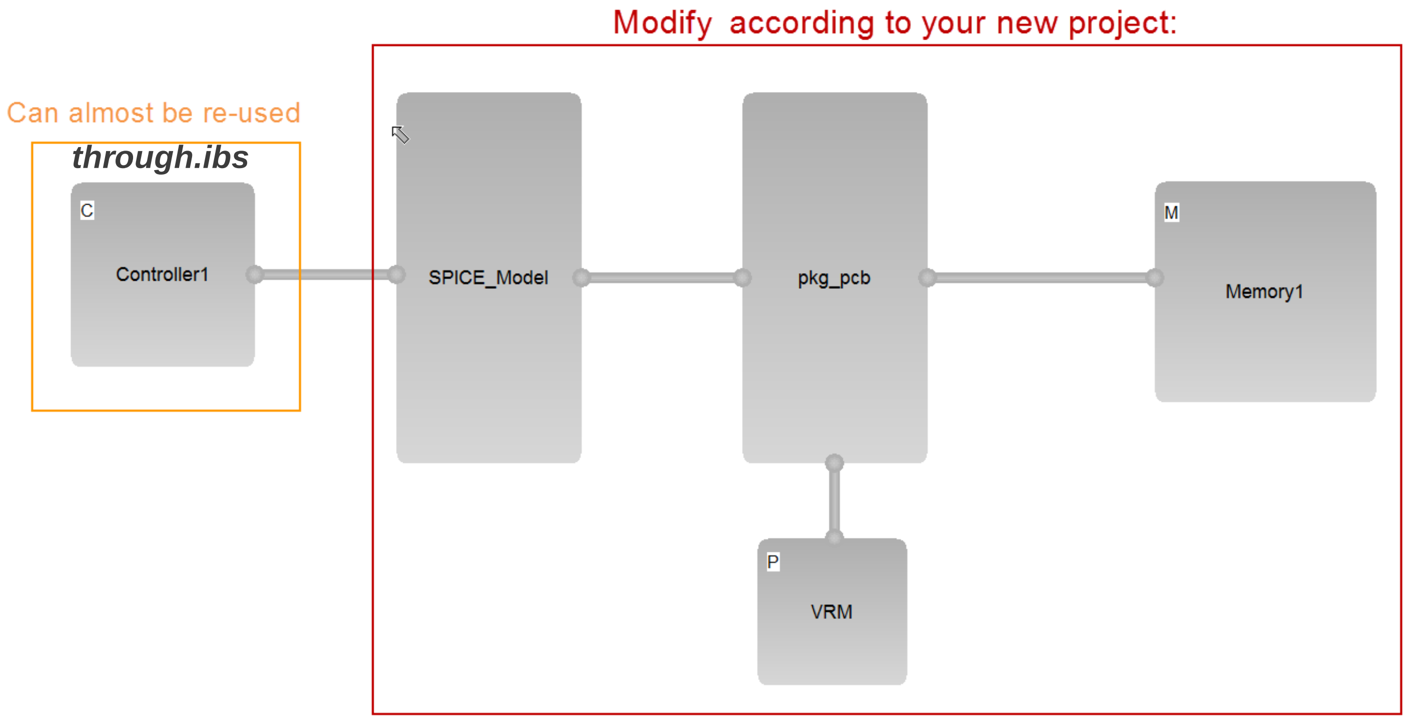
	Bus Group/Signal	Stimulus Pattern	Stimulus Offset (ns)	Transmit IO Model	Status
⊟ ✓	DATA0	1000110101110...	Default		
☐	dm0			Write	Not Connected
✓	dq0	1000110101110...	0.5T	Write	Signal
✓	dq1	1000110101110...	0.5T	Write	Signal
✓	dq2	1000110101110...	0.5T	Write	Signal
☐	dq3			Write	Not Connected
☐	dq4			Write	Not Connected
☐	dq5			Write	Not Connected
☐	dq6			Write	Not Connected
☐	dq7			Write	Not Connected
✓	dqs0p	10..	0.75T	Write	Timing Ref
✓	dqs0n	01..	0.75T	Write	Timing Ref
⊞ ☐	DATA1				

Write Operation (cont')


Parallel Bus Analysis - D:\Case_Faraday_20150408_S...\spice_ibis.ssix
IO circuit block
through.ibs
Digital data in
C
Controller1
SPICE_Model
pkg_pcb
M
Memory1
Digital data
Directly out
P
VRM
Analog data observed
at memory pin
2D Curves (Time Variation)
Waveform
AC/DC Threshold: <None>
VREF(dc)(V): 0.75
Curves
Die Pad
Power & Ground
DATA0
dm0
dq0
dq1
dq2
dq3
dq4
dq5
dq6
dq7
dqs0p
dqs0n
dqs0p-dqs0n
DATA1
Rx: Memory1
Die Pad
Power & Ground
Loaded Curves
Voltage (V)
Time (ns)
result\3\Data_Write_Typ_Typ\DiePad\Controller1_dq0.cur
result\3\Data_Write_Typ_Typ\DiePad\Controller1_dq1.cur
result\3\Data_Write_Typ_Typ\DiePad\Memory1_DQ0.cur
result\3\Data_Write_Typ_Typ\DiePad\Memory1_DQ1.cur
result\3\Data_Write_Typ_Typ\DiePad\Memory1_DQ2.cur

Write Operation (cont')

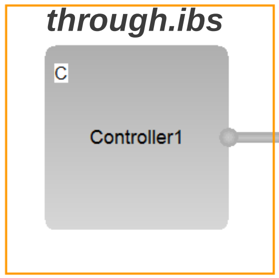
For a new project, the kit is re-usable with little modification

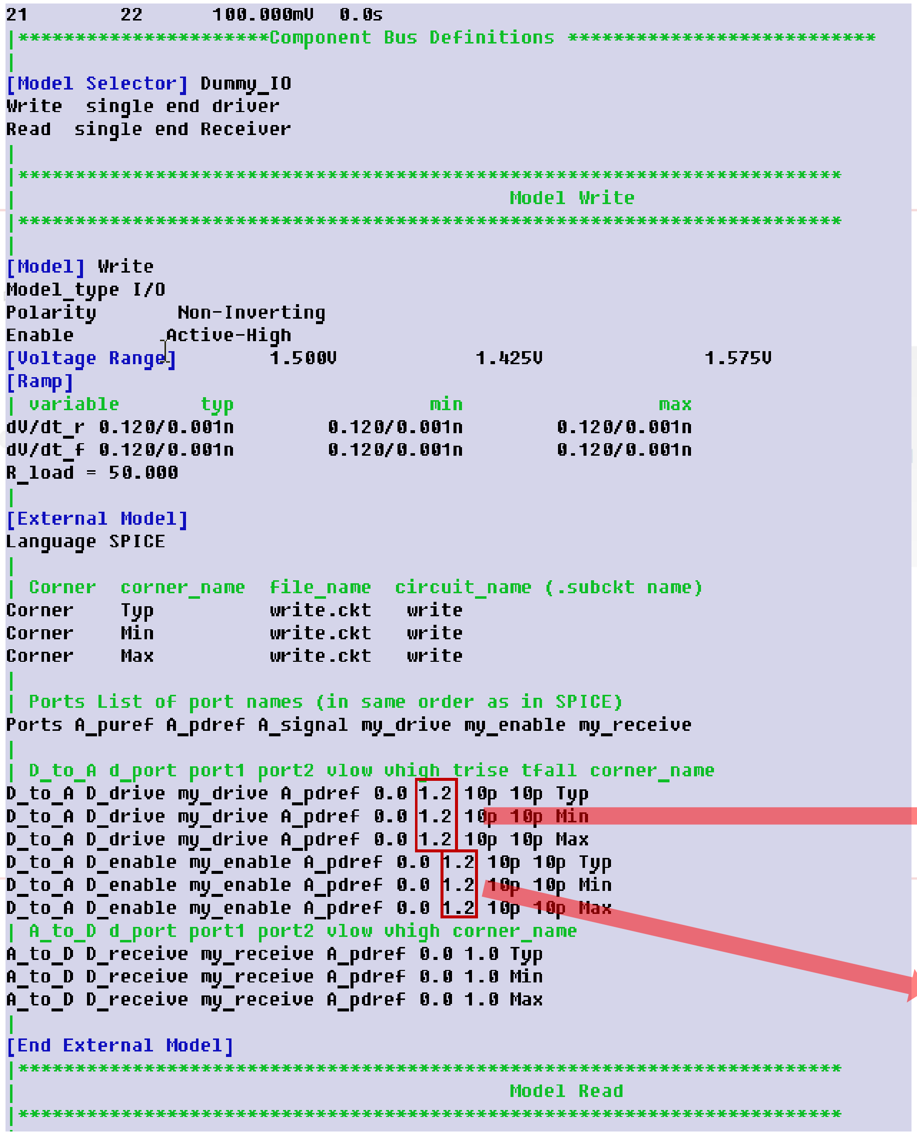


Write Operation (cont')

For a new project, the kit is re-usable with little modification

Can almost be re-used



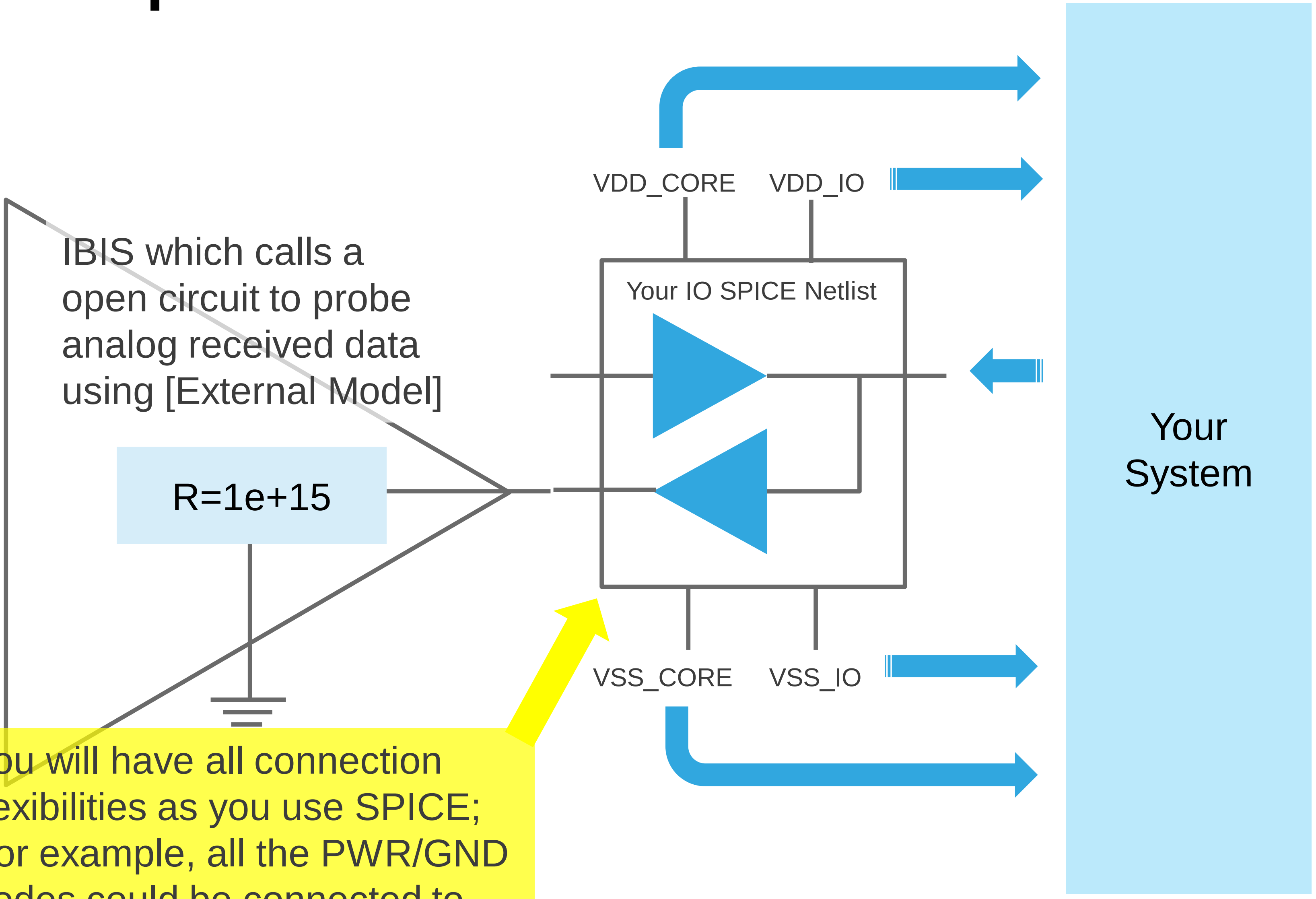


```
21         22       100.000mV   0.0s
|**********************Component Bus Definitions ***************************
|
[Model Selector] Dummy_IO
Write  single end driver
Read  single end Receiver
|
|************************************************************************
|                                          Model Write
|************************************************************************
|
[Model] Write
Model_type I/O
Polarity       Non-Inverting
Enable        Active-High
[Voltage Range]        1.500V           1.425V            1.575V
[Ramp]
| variable       typ                min                max
dV/dt_r 0.120/0.001n       0.120/0.001n       0.120/0.001n
dV/dt_f 0.120/0.001n       0.120/0.001n       0.120/0.001n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner  corner_name  file_name  circuit_name (.subckt name)
Corner    Typ          write.ckt   write
Corner    Min          write.ckt   write
Corner    Max          write.ckt   write
|
| Ports List of port names (in same order as in SPICE)
Ports A_puref A_pdref A_signal my_drive my_enable my_receive
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall corner_name
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Typ
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Min
D_to_A D_drive my_drive A_pdref 0.0 1.2 10p 10p Max
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Typ
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Min
D_to_A D_enable my_enable A_pdref 0.0 1.2 10p 10p Max
| A_to_D d_port port1 port2 vlow vhigh corner_name
A_to_D D_receive my_receive A_pdref 0.0 1.0 Typ
A_to_D D_receive my_receive A_pdref 0.0 1.0 Min
A_to_D D_receive my_receive A_pdref 0.0 1.0 Max
|
[End External Model]
|************************************************************************
|                                          Model Read
|************************************************************************
```

Modify to the correct voltage level of your IO's Input node

Modify to the correct voltage level of your IO's Enable node

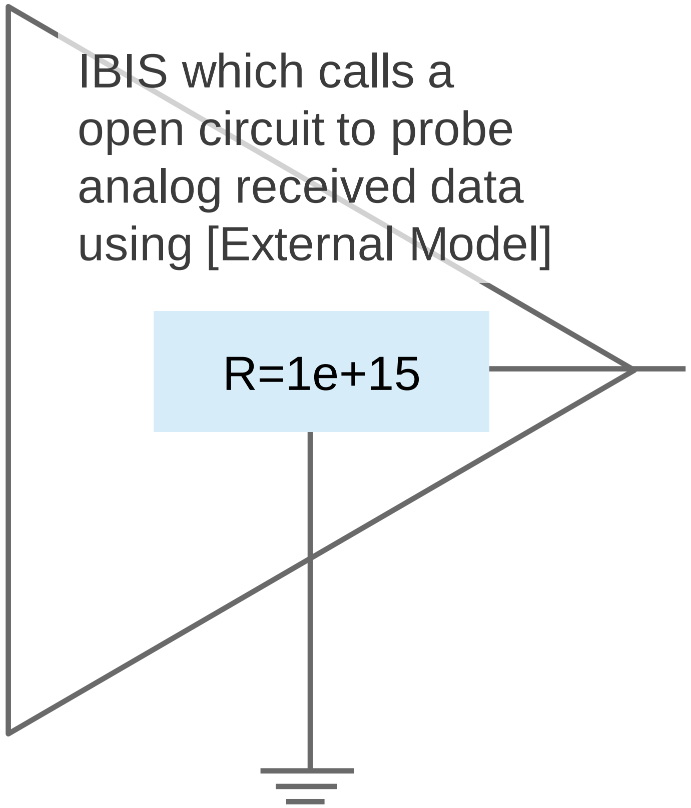
Read Operation

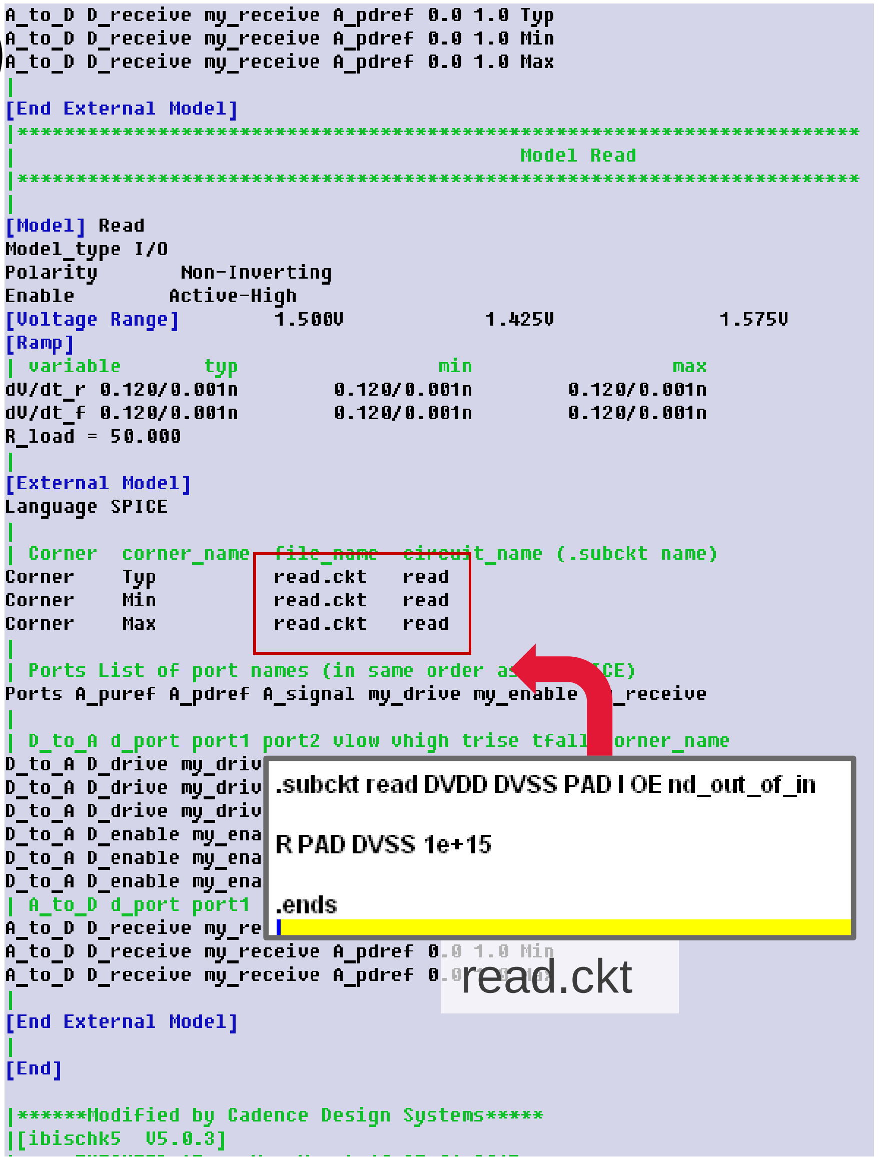


IBIS which calls a open circuit to probe analog received data using [External Model]

You will have all connection flexibilities as you use SPICE; For example, all the PWR/GND nodes could be connected to real PDN.

Read Operation (cont’)


IBIS which calls a open circuit to probe analog received data using [External Model]
R=1e+15



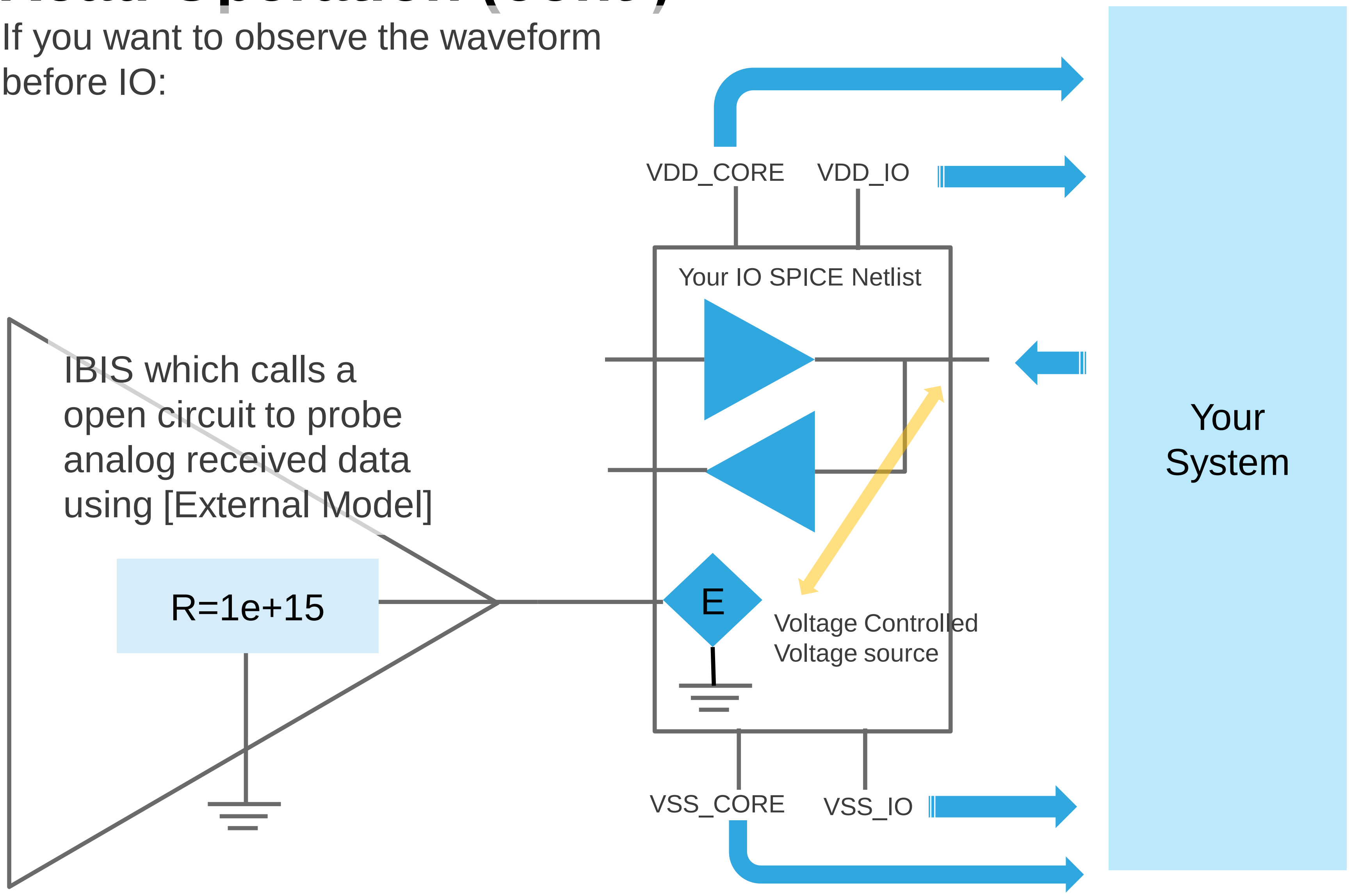
```
A_to_D D_receive my_receive A_pdref 0.0 1.0 Typ
A_to_D D_receive my_receive A_pdref 0.0 1.0 Min
A_to_D D_receive my_receive A_pdref 0.0 1.0 Max
|
[End External Model]
|**************************************************************************
|                                            Model Read
|**************************************************************************
|
[Model] Read
Model_type I/O
Polarity       Non-Inverting
Enable         Active-High
[Voltage Range]        1.500V             1.425V              1.575V
[Ramp]
| variable        typ                    min                   max
dV/dt_r  0.120/0.001n          0.120/0.001n          0.120/0.001n
dV/dt_f  0.120/0.001n          0.120/0.001n          0.120/0.001n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner   corner_name  file_name  circuit_name (.subckt name)
Corner     Typ          read.ckt   read
Corner     Min          read.ckt   read
Corner     Max          read.ckt   read
|
| Ports List of port names (in same order as      CE)
Ports A_puref A_pdref A_signal my_drive my_enable   _receive
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall  orner_name
D_to_A D_drive my_driv
D_to_A D_drive my_driv
D_to_A D_drive my_driv
D_to_A D_enable my_ena
D_to_A D_enable my_ena
D_to_A D_enable my_ena
| A_to_D d_port port1
A_to_D D_receive my_re
A_to_D D_receive my_receive A_pdref 0.0 1.0 Min
A_to_D D_receive my_receive A_pdref 0.
|
[End External Model]
|
[End]

|******Modified by Cadence Design Systems*****
|[ibischk5  V5.0.3]
```

```
.subckt read DVDD DVSS PAD I OE nd_out_of_in

R PAD DVSS 1e+15

.ends
```

read.ckt

Read Operation (cont')

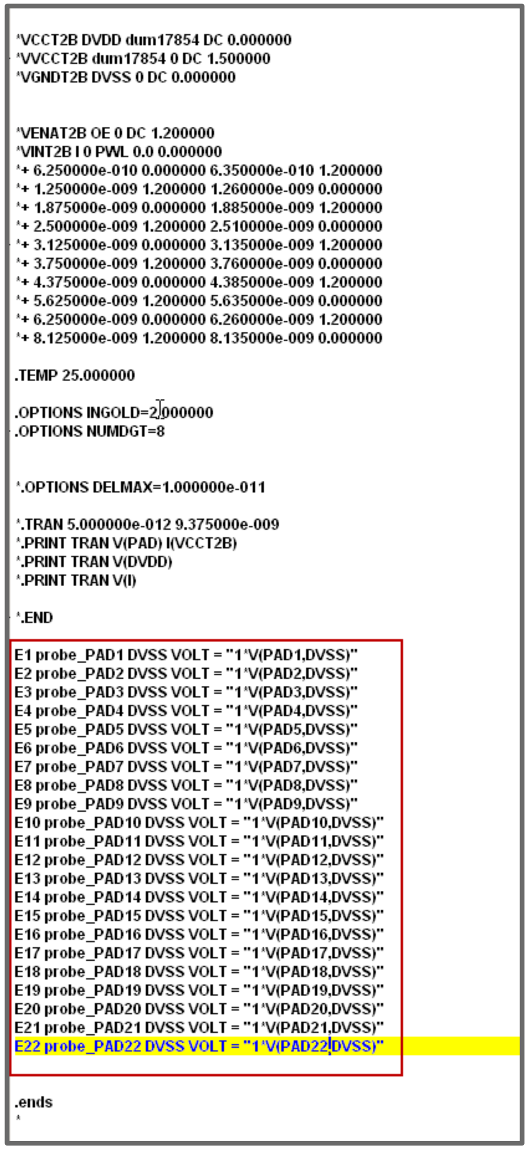
If you want to observe the waveform before IO:


IBIS which calls a open circuit to probe analog received data using [External Model]
R=1e+15
VDD_CORE
VDD_IO
Your IO SPICE Netlist
E
Voltage Controlled Voltage source
VSS_CORE
VSS_IO
Your System

Read Operation (cont')

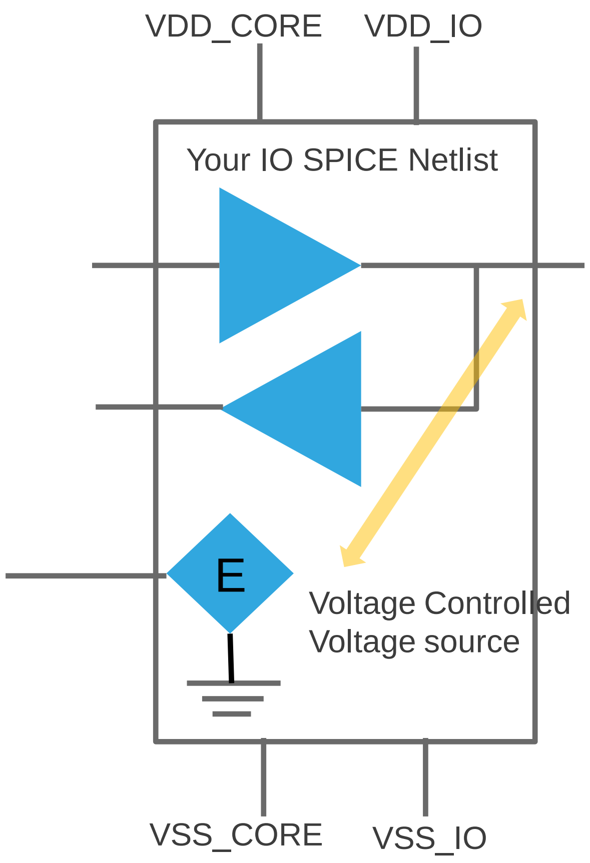
- You will need to modify the netlist of the IO circuit block by adding some "Voltage Controlled Voltage sources" which is controlled by the voltage on IO circuit block's PAD.

- Connect the dummy controller's output pad to these "Voltage Controlled Voltage sources"



```
*VCCT2B DVDD dum17854 DC 0.000000
*VVCCT2B dum17854 0 DC 1.500000
*VGNDT2B DVSS 0 DC 0.000000


*VENAT2B OE 0 DC 1.200000
*VINT2B I 0 PWL 0.0 0.000000
*+ 6.250000e-010 0.000000 6.350000e-010 1.200000
*+ 1.250000e-009 1.200000 1.260000e-009 0.000000
*+ 1.875000e-009 0.000000 1.885000e-009 1.200000
*+ 2.500000e-009 1.200000 2.510000e-009 0.000000
*+ 3.125000e-009 0.000000 3.135000e-009 1.200000
*+ 3.750000e-009 1.200000 3.760000e-009 0.000000
*+ 4.375000e-009 0.000000 4.385000e-009 1.200000
*+ 5.625000e-009 1.200000 5.635000e-009 0.000000
*+ 6.250000e-009 0.000000 6.260000e-009 1.200000
*+ 8.125000e-009 1.200000 8.135000e-009 0.000000

.TEMP 25.000000

.OPTIONS INGOLD=2.000000
.OPTIONS NUMDGT=8


*.OPTIONS DELMAX=1.000000e-011

*.TRAN 5.000000e-012 9.375000e-009
*.PRINT TRAN V(PAD) I(VCCT2B)
*.PRINT TRAN V(DVDD)
*.PRINT TRAN V(I)

*.END

E1 probe_PAD1 DVSS VOLT = "1*V(PAD1,DVSS)"
E2 probe_PAD2 DVSS VOLT = "1*V(PAD2,DVSS)"
E3 probe_PAD3 DVSS VOLT = "1*V(PAD3,DVSS)"
E4 probe_PAD4 DVSS VOLT = "1*V(PAD4,DVSS)"
E5 probe_PAD5 DVSS VOLT = "1*V(PAD5,DVSS)"
E6 probe_PAD6 DVSS VOLT = "1*V(PAD6,DVSS)"
E7 probe_PAD7 DVSS VOLT = "1*V(PAD7,DVSS)"
E8 probe_PAD8 DVSS VOLT = "1*V(PAD8,DVSS)"
E9 probe_PAD9 DVSS VOLT = "1*V(PAD9,DVSS)"
E10 probe_PAD10 DVSS VOLT = "1*V(PAD10,DVSS)"
E11 probe_PAD11 DVSS VOLT = "1*V(PAD11,DVSS)"
E12 probe_PAD12 DVSS VOLT = "1*V(PAD12,DVSS)"
E13 probe_PAD13 DVSS VOLT = "1*V(PAD13,DVSS)"
E14 probe_PAD14 DVSS VOLT = "1*V(PAD14,DVSS)"
E15 probe_PAD15 DVSS VOLT = "1*V(PAD15,DVSS)"
E16 probe_PAD16 DVSS VOLT = "1*V(PAD16,DVSS)"
E17 probe_PAD17 DVSS VOLT = "1*V(PAD17,DVSS)"
E18 probe_PAD18 DVSS VOLT = "1*V(PAD18,DVSS)"
E19 probe_PAD19 DVSS VOLT = "1*V(PAD19,DVSS)"
E20 probe_PAD20 DVSS VOLT = "1*V(PAD20,DVSS)"
E21 probe_PAD21 DVSS VOLT = "1*V(PAD21,DVSS)"
E22 probe_PAD22 DVSS VOLT = "1*V(PAD22,DVSS)"


.ends
*
```

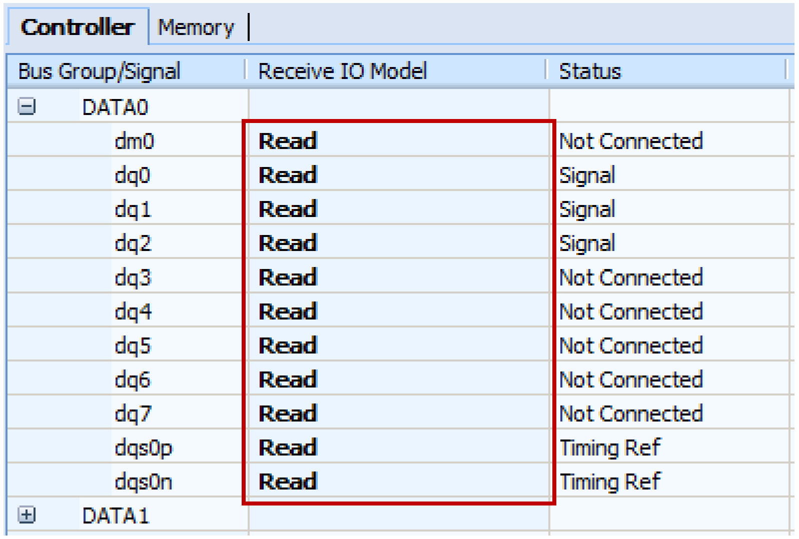
spice_io_read.ckt



cādence®

Read Operation (cont')

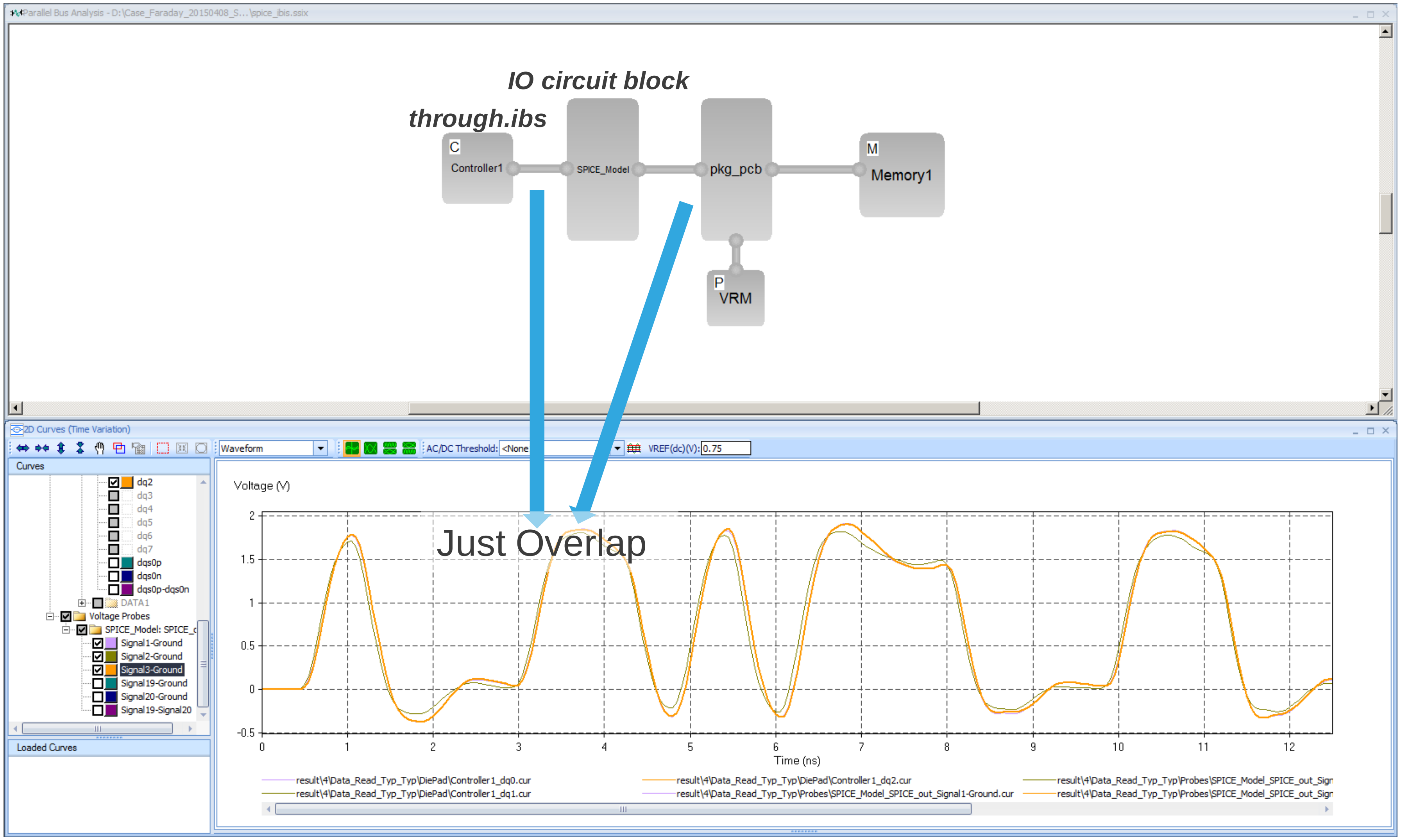
- In a bus simulation, user can easily switch the model of the Controller's buffer to "Read"



Controller | Memory

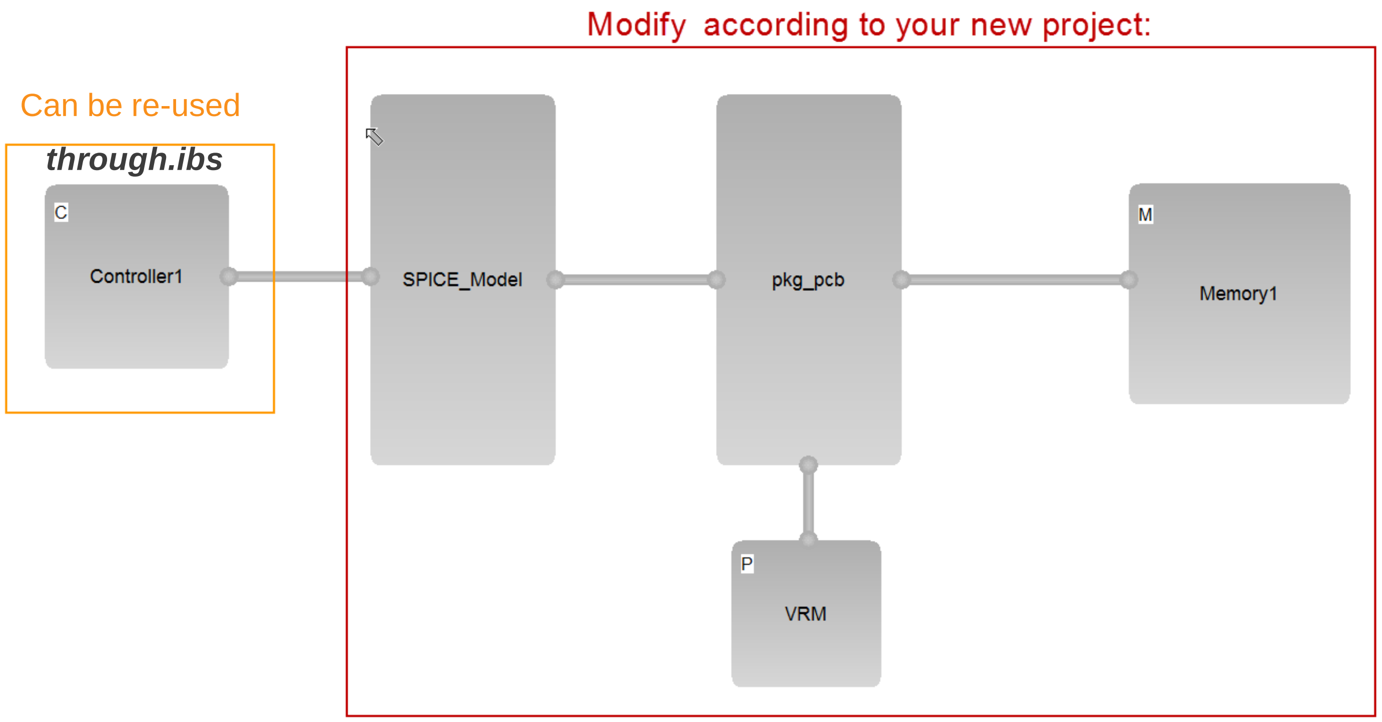
Bus Group/Signal	Receive IO Model	Status
⊟ DATA0		
dm0	**Read**	Not Connected
dq0	**Read**	Signal
dq1	**Read**	Signal
dq2	**Read**	Signal
dq3	**Read**	Not Connected
dq4	**Read**	Not Connected
dq5	**Read**	Not Connected
dq6	**Read**	Not Connected
dq7	**Read**	Not Connected
dqs0p	**Read**	Timing Ref
dqs0n	**Read**	Timing Ref
⊞ DATA1		

Read Operation (cont’)


Parallel Bus Analysis - D:\Case_Faraday_20150408_S...\spice_ibis.ssix
IO circuit block
through.ibs
C
Controller1
SPICE_Model
pkg_pcb
M
Memory1
P
VRM
2D Curves (Time Variation)
Waveform
AC/DC Threshold: <None
VREF(dc)(V): 0.75
Curves
dq2
dq3
dq4
dq5
dq6
dq7
dqs0p
dqs0n
dqs0p-dqs0n
DATA1
Voltage Probes
SPICE_Model: SPICE_c
Signal1-Ground
Signal2-Ground
Signal3-Ground
Signal19-Ground
Signal20-Ground
Signal19-Signal20
Loaded Curves
Voltage (V)
Just Overlap
Time (ns)
result\4\Data_Read_Typ_Typ\DiePad\Controller1_dq0.cur
result\4\Data_Read_Typ_Typ\DiePad\Controller1_dq1.cur
result\4\Data_Read_Typ_Typ\DiePad\Controller1_dq2.cur
result\4\Data_Read_Typ_Typ\Probes\SPICE_Model_SPICE_out_Signal1-Ground.cur
result\4\Data_Read_Typ_Typ\Probes\SPICE_Model_SPICE_out_Sigr
result\4\Data_Read_Typ_Typ\Probes\SPICE_Model_SPICE_out_Sigr

Read Operation (cont')

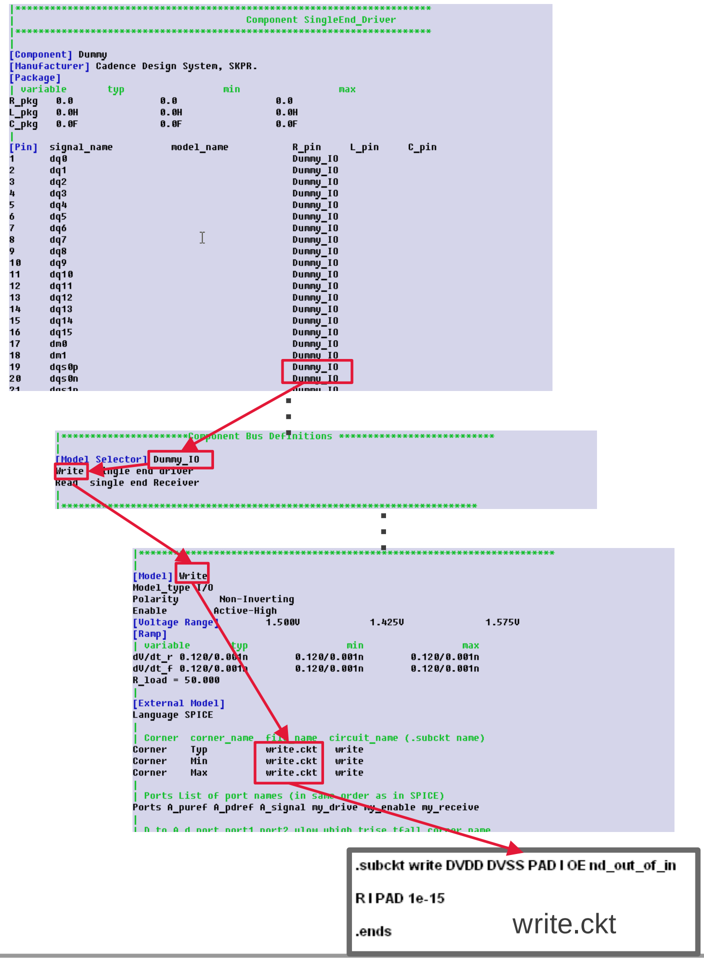
For a new project, the kit is re-usable with little modification



Differential Buffer

No matter true or pseudo, you can use the same kit without any modification because:
(take "**write operation**" as an example)

Controller (***through.ibs***)



```
|*************************************************************************
|                                     Component SingleEnd_Driver
|*************************************************************************
|
[Component] Dummy
[Manufacturer] Cadence Design System, SKPR.
[Package]
| variable       typ                min                max
R_pkg   0.0                0.0                0.0
L_pkg   0.0H               0.0H               0.0H
C_pkg   0.0F               0.0F               0.0F
|
[Pin]  signal_name          model_name           R_pin     L_pin     C_pin
1      dq0                                       Dummy_IO
2      dq1                                       Dummy_IO
3      dq2                                       Dummy_IO
4      dq3                                       Dummy_IO
5      dq4                                       Dummy_IO
6      dq5                                       Dummy_IO
7      dq6                                       Dummy_IO
8      dq7                                       Dummy_IO
9      dq8                                       Dummy_IO
10     dq9                                       Dummy_IO
11     dq10                                      Dummy_IO
12     dq11                                      Dummy_IO
13     dq12                                      Dummy_IO
14     dq13                                      Dummy_IO
15     dq14                                      Dummy_IO
16     dq15                                      Dummy_IO
17     dm0                                       Dummy_IO
18     dm1                                       Dummy_IO
19     dqs0p                                     Dummy_IO
20     dqs0n                                     Dummy_IO
21     dqs1p                                     Dummy_IO
```

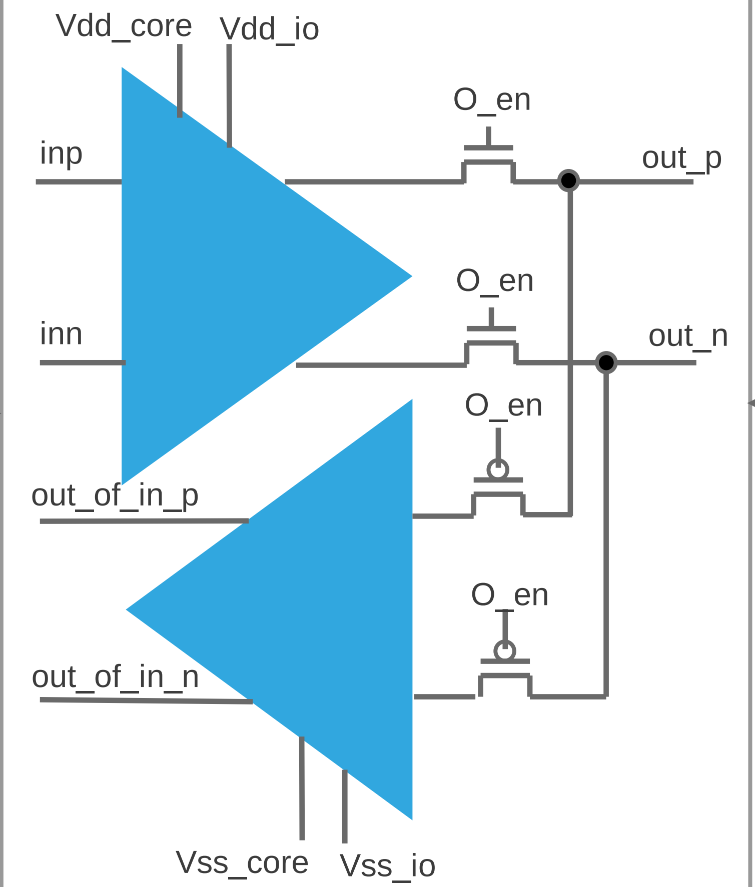
```
|**********************Component Bus Definitions ***************************
|
[Model Selector] Dummy_IO
Write  single end driver
Read  single end Receiver
|
|*************************************************************************
```

```
|*************************************************************************
|
[Model] Write
Model_type I/O
Polarity       Non-Inverting
Enable        Active-High
[Voltage Range]       1.500V          1.425V              1.575V
[Ramp]
| variable        typ                min                max
dV/dt_r 0.120/0.001n       0.120/0.001n       0.120/0.001n
dV/dt_f 0.120/0.001n       0.120/0.001n       0.120/0.001n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner  corner_name  file_name  circuit_name (.subckt name)
Corner    Typ          write.ckt   write
Corner    Min          write.ckt   write
Corner    Max          write.ckt   write
|
| Ports List of port names (in same order as in SPICE)
Ports A_puref A_pdref A_signal my_drive my_enable my_receive
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall corner_name
```

```
.subckt write DVDD DVSS PAD I OE nd_out_of_in
R I PAD 1e-15
.ends
```

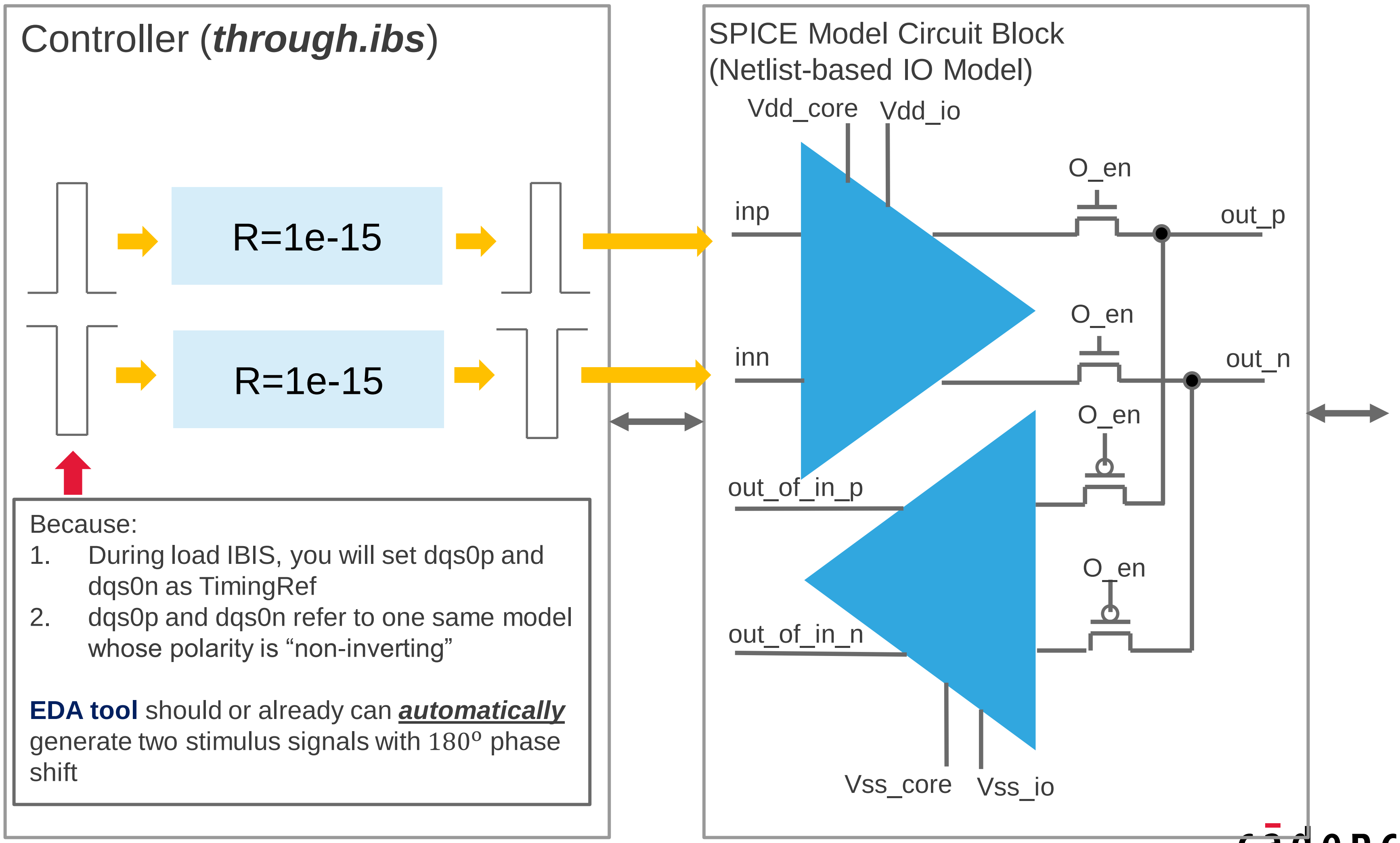
write.ckt

SPICE Model Circuit Block (Netlist-based IO Model)



Differential Buffer (cont')

For Truly Differential Buffer type1:



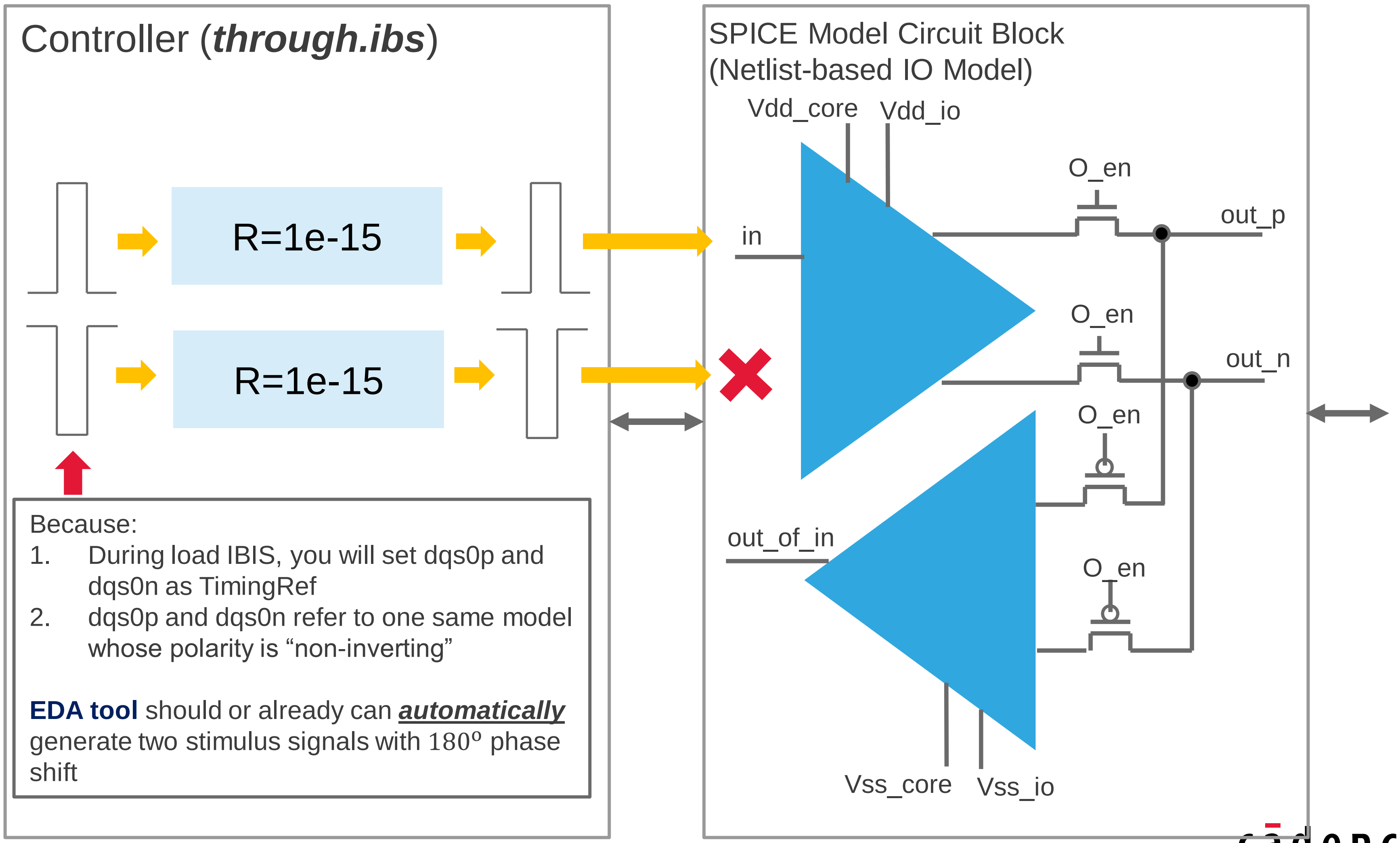
Because:

1. During load IBIS, you will set dqs0p and dqs0n as TimingRef
2. dqs0p and dqs0n refer to one same model whose polarity is "non-inverting"

EDA tool should or already can ***automatically*** generate two stimulus signals with 180º phase shift

Differential Buffer (cont')

For Truly Differential Buffer type2:



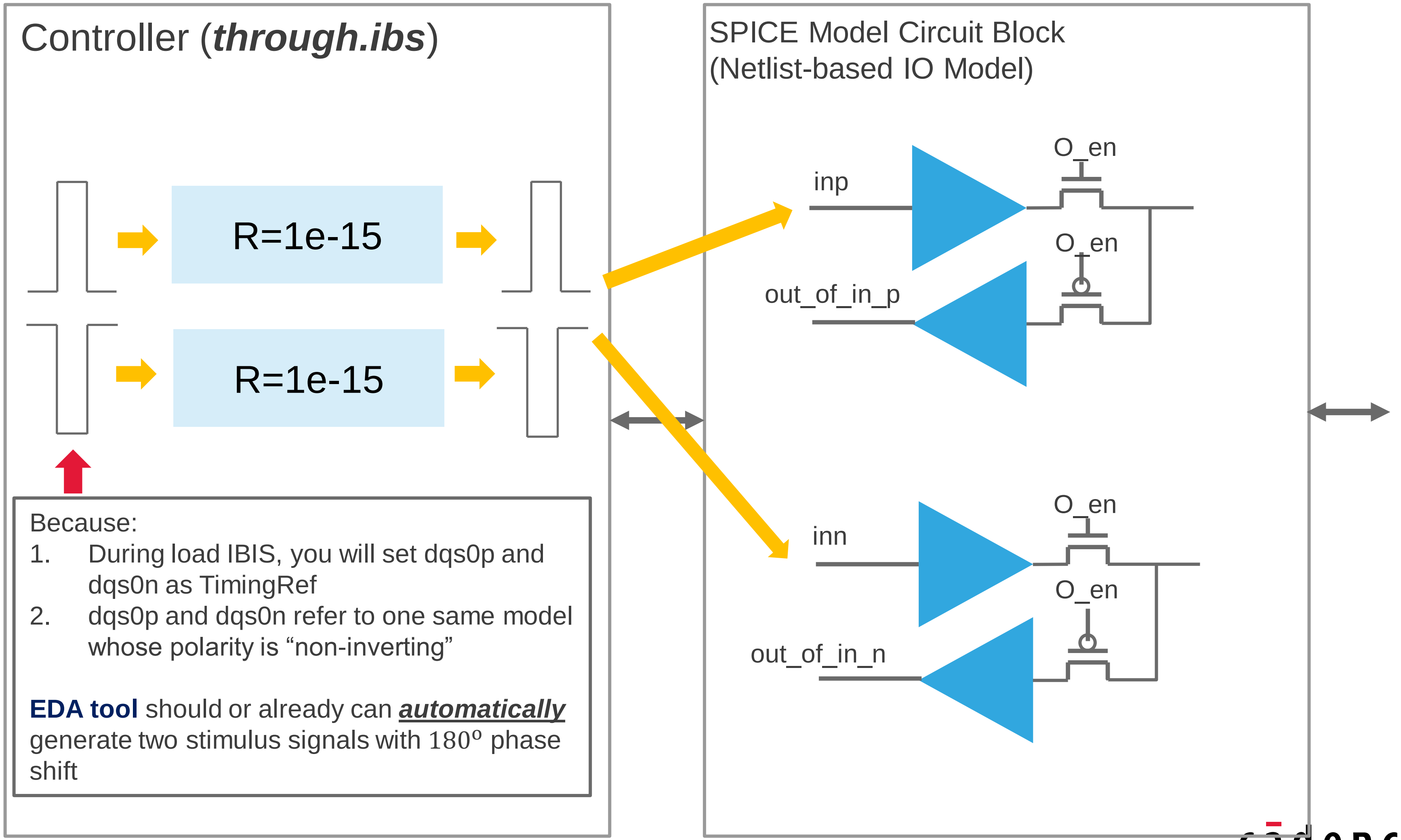
Because:

1. During load IBIS, you will set dqs0p and dqs0n as TimingRef
2. dqs0p and dqs0n refer to one same model whose polarity is "non-inverting"

EDA tool should or already can ***automatically*** generate two stimulus signals with 180º phase shift

Differential Buffer (cont')

For Pseudo Differential Buffer:



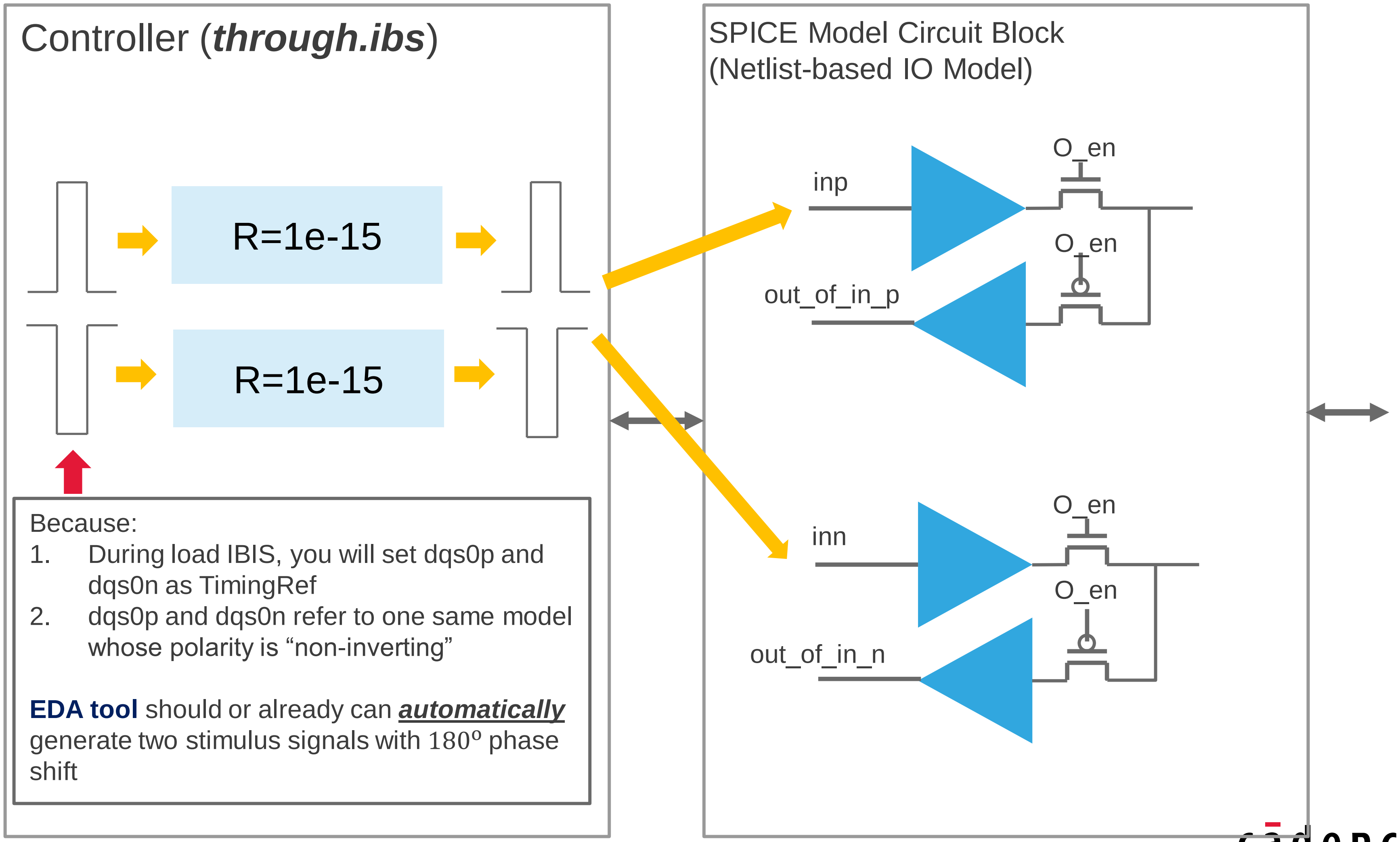
Because:

1. During load IBIS, you will set dqs0p and dqs0n as TimingRef
2. dqs0p and dqs0n refer to one same model whose polarity is "non-inverting"

EDA tool should or already can ***automatically*** generate two stimulus signals with 180^o phase shift

Differential Buffer (cont')

For Pseudo Differential Buffer:



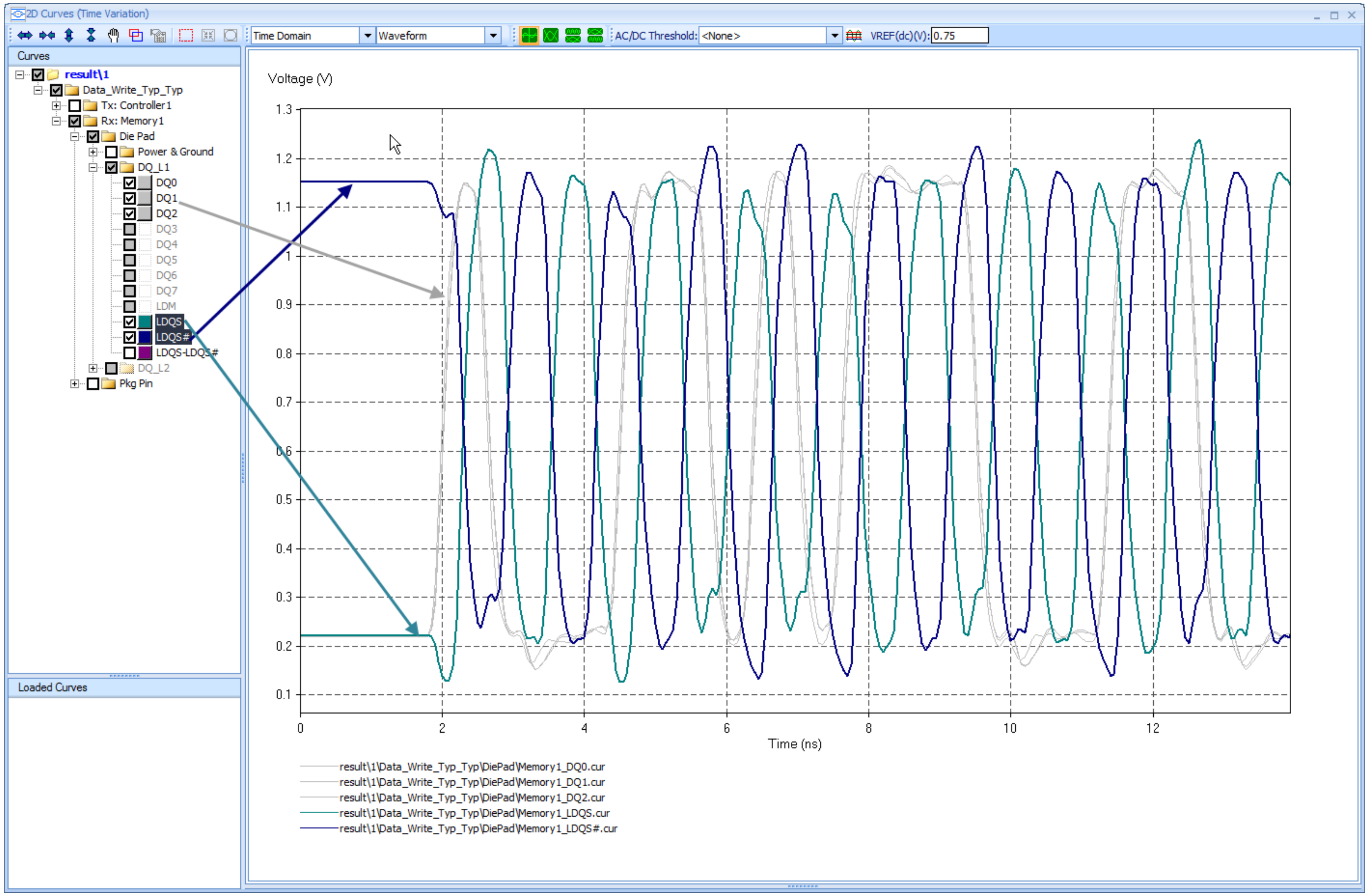
Because:

1. During load IBIS, you will set dqs0p and dqs0n as TimingRef
2. dqs0p and dqs0n refer to one same model whose polarity is "non-inverting"

EDA tool should or already can ***automatically*** generate two stimulus signals with 180º phase shift

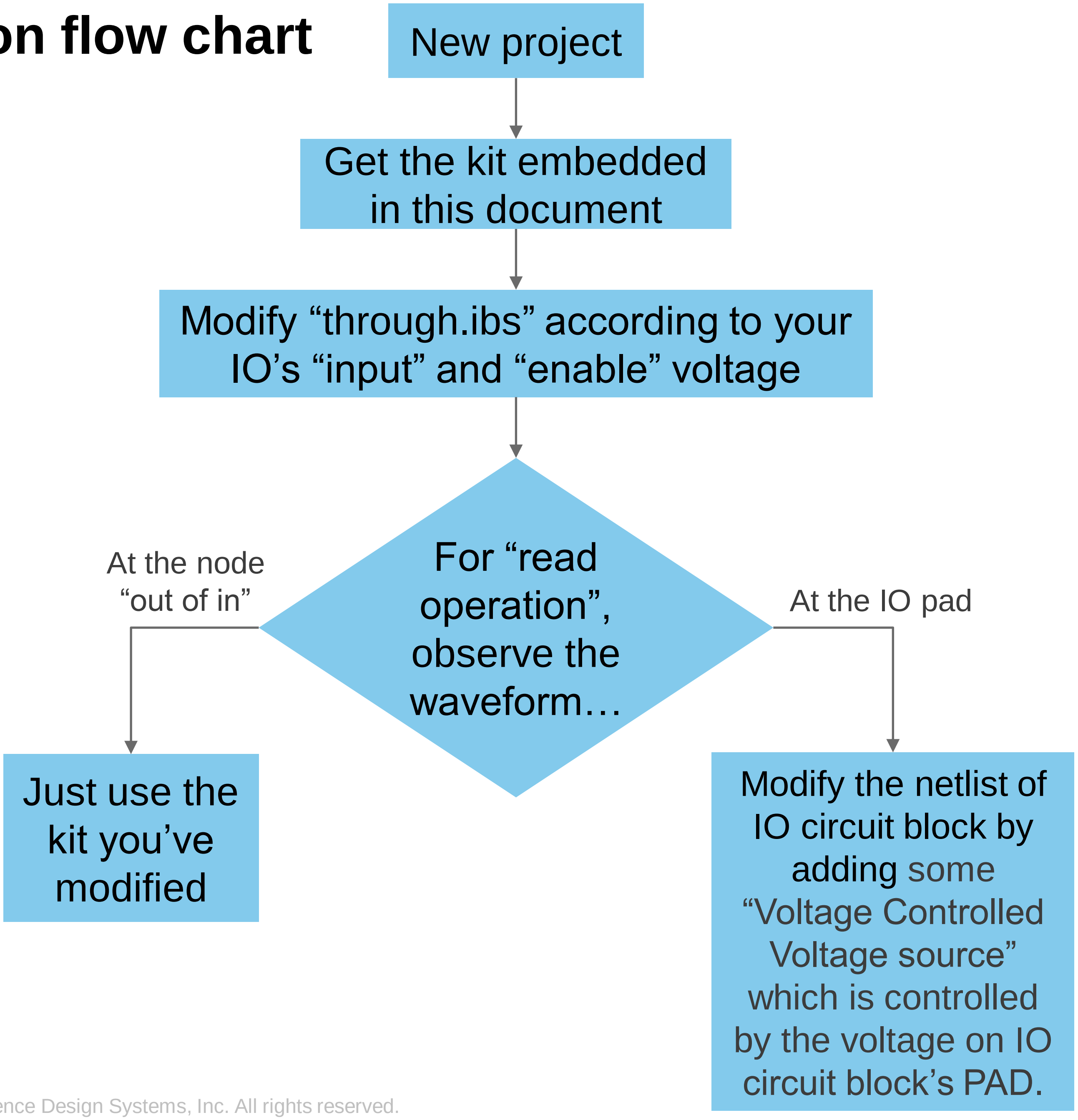
Differential Buffer (cont')

An example with method2: 3 DQ and 1 pair of DQS



result\1\Data_Write_Typ_Typ\DiePad\Memory1_DQ0.cur
result\1\Data_Write_Typ_Typ\DiePad\Memory1_DQ1.cur
result\1\Data_Write_Typ_Typ\DiePad\Memory1_DQ2.cur
result\1\Data_Write_Typ_Typ\DiePad\Memory1_LDQS.cur
result\1\Data_Write_Typ_Typ\DiePad\Memory1_LDQS#.cur

Solution flow chart


New project
Get the kit embedded in this document
Modify “through.ibs” according to your IO’s “input” and “enable” voltage
For “read operation”, observe the waveform…
At the node “out of in”
At the IO pad
Just use the kit you’ve modified
Modify the netlist of IO circuit block by adding some “Voltage Controlled Voltage source” which is controlled by the voltage on IO circuit block’s PAD.

Summary

Notes

- For analysis of SSN
  - With the full flexibilities as using SPICE, this method allows user to do **a power-aware SSN analysis between multiple signals and multiple PDNs**, such as VDD_Core/VSS_Core, VDDIO/VSSIO, VTT and VREF together at the same time.
- Model modification and change of topology
  - This method requires the change of topology - inserting a Spice block but the wrapping .sp file according to the description in [External Model] is not necessary since it's ready for use in the kit - **easy for re-using** this kit in a new project of a new IO.
- Read Operation
  - As this method requires the change of topology when building topology, during the analysis of read operation, user will need to modify **the netlist of the IO circuit block** by adding some "Voltage Controlled Voltage source" which is controlled by the voltage on IO circuit block's PAD.
- Differential Buffer
  - This method can use the same kit without additional modification for both single-end and differential buffer, **no matter it's truly or pseudo**.

References

- IBIS specification (v6.1)
  http://www.ibis.org/ver6.1/ver6_1.pdf

cadence®