02. August 2007

IBIS4.2 FOR DDR2 TIMING ANALYSIS

September 11, 2007
Asian IBIS Summit, Beijing China

Guan Tao
guantao@huawei.com

www.huawei.com

HUAWEI TECHNOLOGIES Co., Ltd.

HUAWEI

Outline

- **New technologies have brought challenges to IBIS model**
- **DDR2 timing analysis**
  - The read cycle timing analysis
  - The write cycle timing analysis
- **IBIS4.2 model meets the challenges**

Challenges

- **More SI and timing problems emerge as the rate of bus increases , these problems can be eliminated by new techniques :**
  - ODT (On Die Termination);
  - OCD (Off-chip Driver).
- **Challenges of DDR2 modeling:**
  - How to model the ODT circuit;
  - How to use the slew rate table.

  The setup and holdup time are varying with the slope of the signal in DDR2.

DDR2 Timing Analysis Method

A proper method of a DDR2 timing analysis should contain the following essentials:

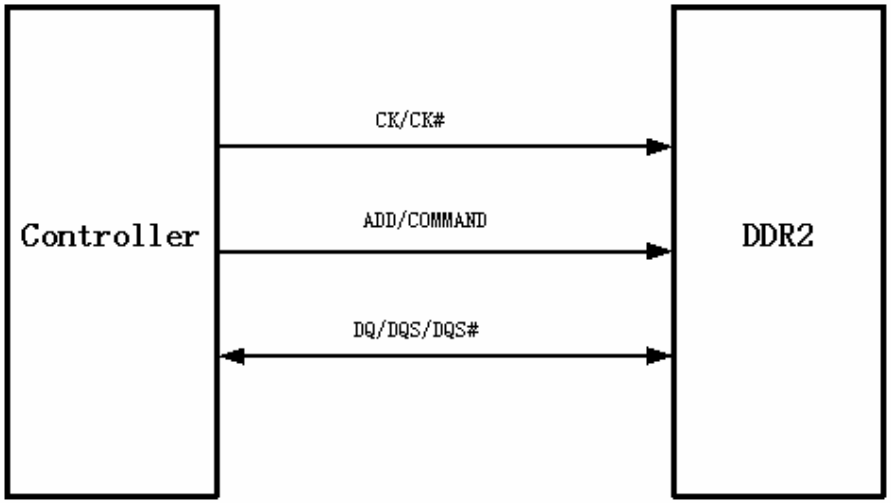
- **Seeking for a balance between simulation efficiency and waveform quality.**
- **The parallel simulation and crosstalk can be processed by EDA platforms.**
- **The EDA platform can process the PRBS (pseudo-random bit sequence) code;**
- **Topology、matching and transmitter/receiver buffer can be automatically analyzed**
- **The settle and switch delay between standard load and practical load can be considered;**
- **Influence of voltage and temperature can be included in the simulation;**
- **The ODT circuit and slew rate table can be considered.**

Current simulation method is not good enough to support the features listed here!

Timing Relationships of DDR2

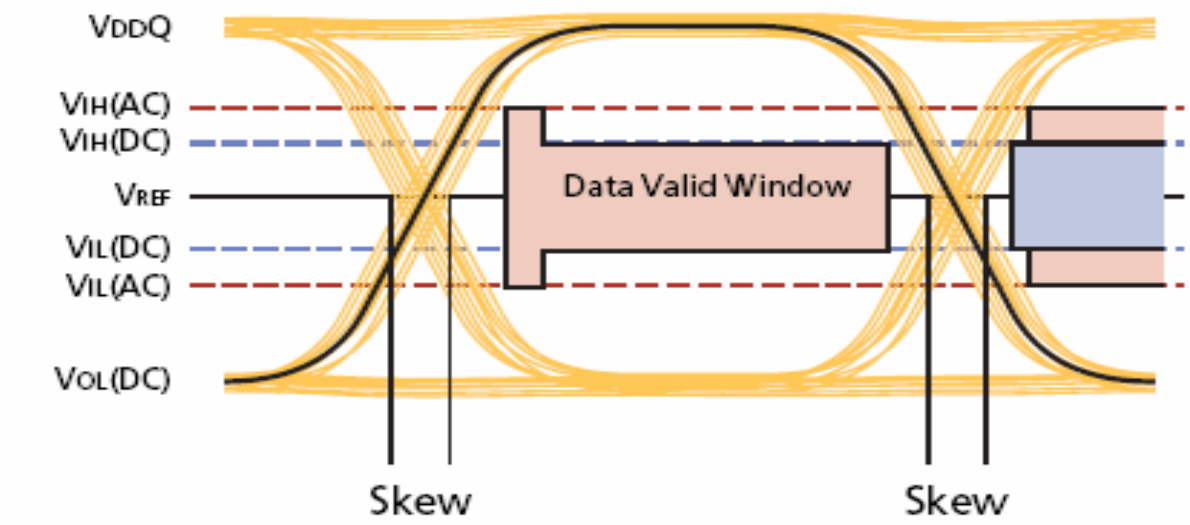
Three timing relationships in DDR2 timing analysis:

- **ADDRESS、COMMAND and CK/CK#;**
- **DQ and DQS/DQS#;**
- **DQS and CK/CK#.**



Data Valid Window

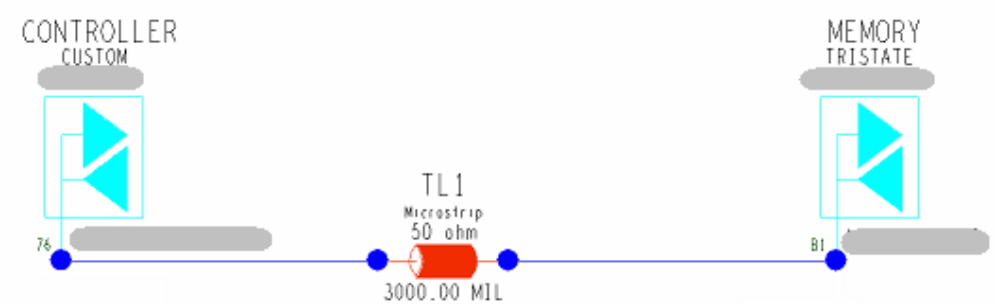
The SSTL_18 is intended for DDR2 interface application.



DDR2 Timing Analysis with IBIS 3.2

Following is an example of timing analysis of DQ and DQS. The ADDRESS/COMMAND timing analysis is similar.

- **The topology of DQ/DQS is point to point feature.**
- **The models of CONTROLLER and MEMORY are IBIS3.2 model.**

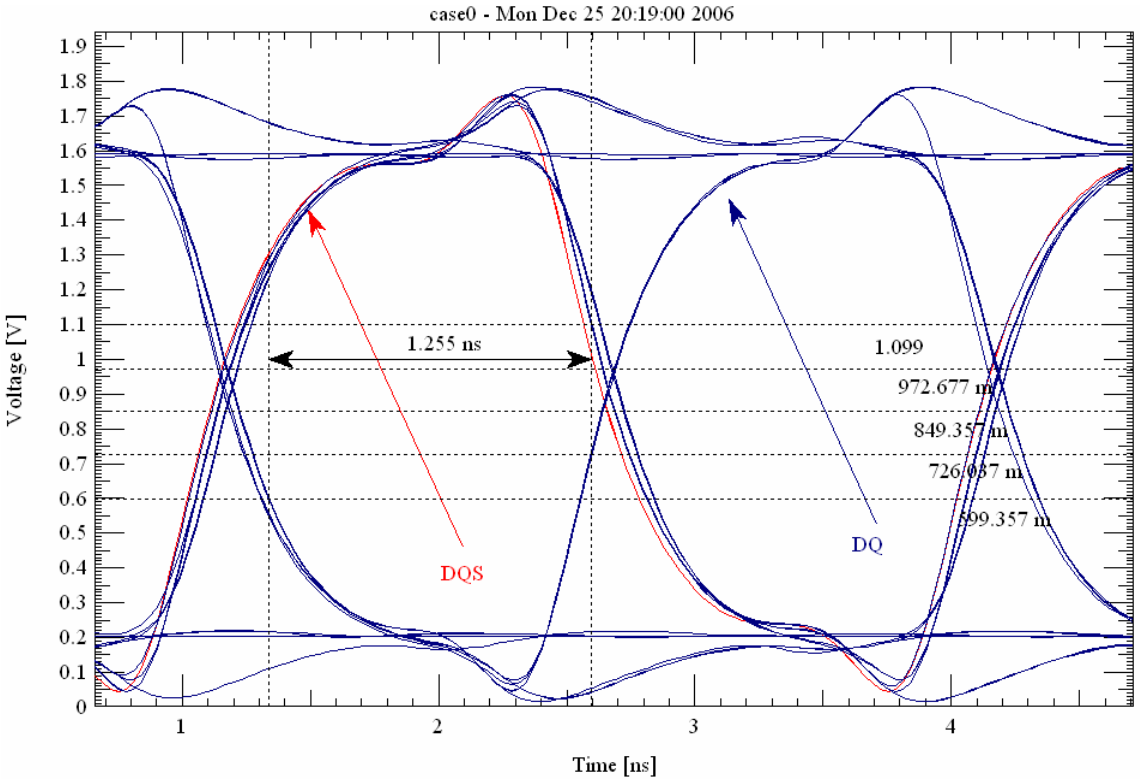


Read Cycle Timing Analysis (fast)

In fast mode, the simulated eye diagram of read cycle is as follow.

data valid window :1.255ns

waveform distortion: 0.245ns

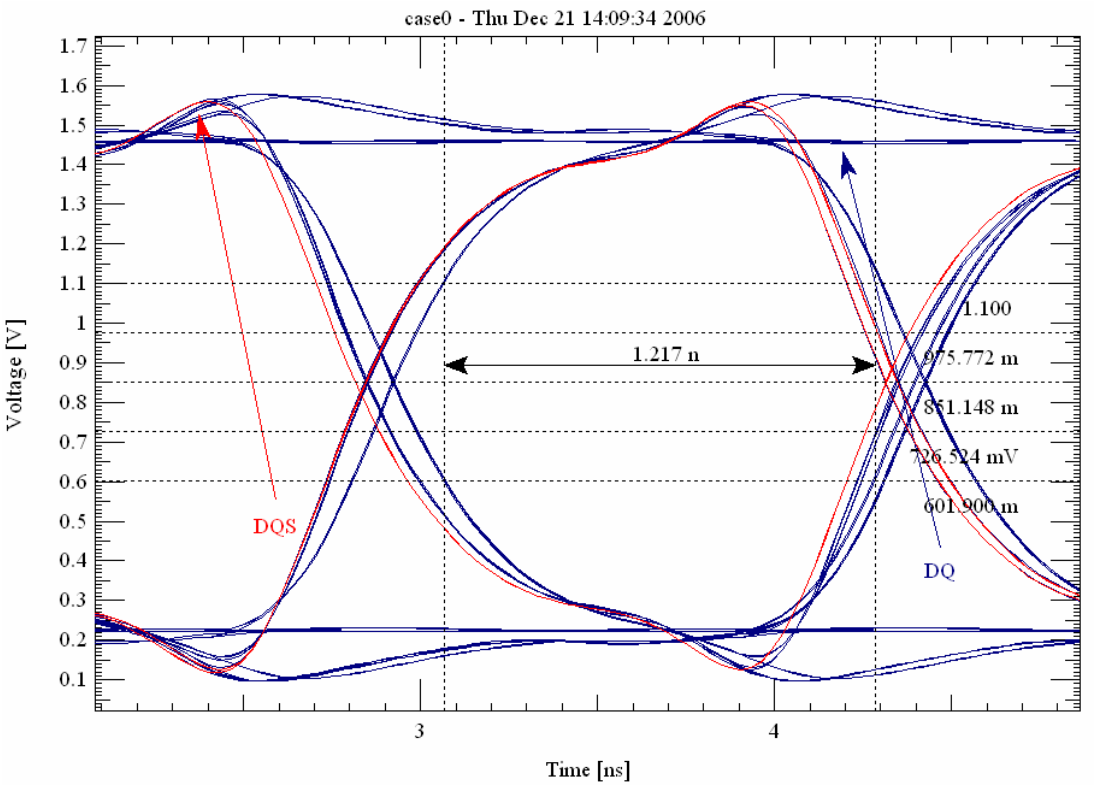


Read Cycle Timing Analysis (slow)

In slow mode, the simulated eye diagram of read cycle is as follow.

data valid window: 1.217ns

waveform distortion :0.283ns



Read Cycle Timing Calculation

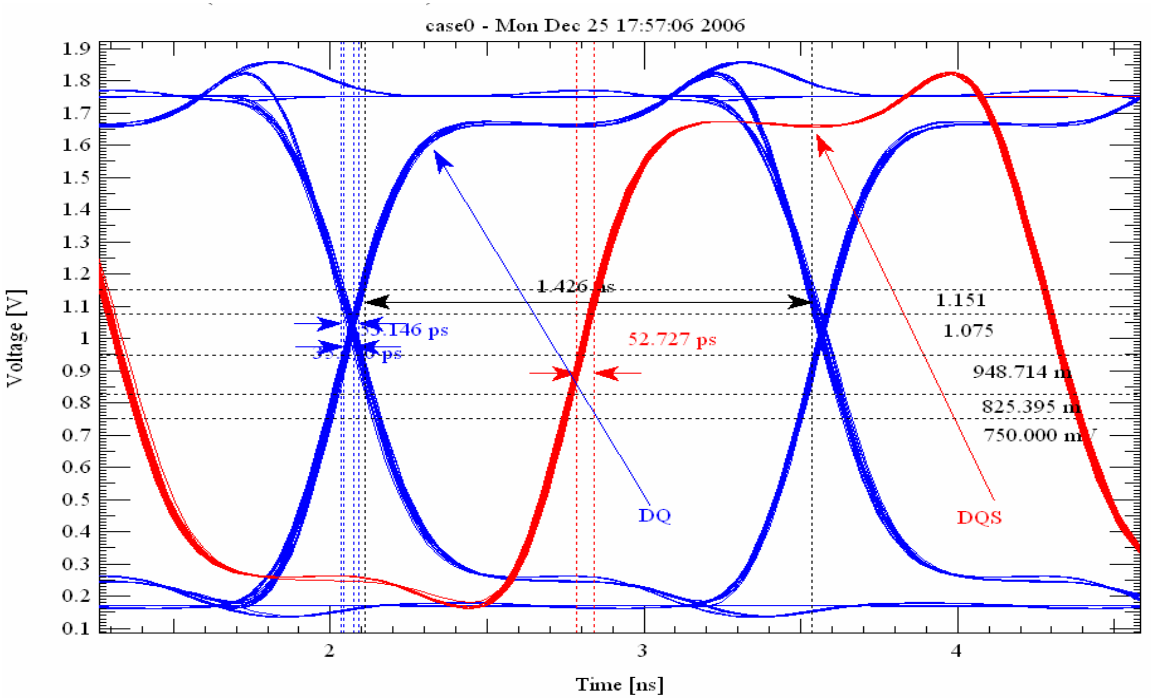
	TimingParameter	Time(ps)	Description
Memory Parameter	tHP	1500	Ideal clock
	tDCD	150	Output clock (includingmemory clock duty cycle distortion)
	tDQSQ	240	Skew between DQS and DQ from memory
	tDHS	340	Data hold skew
Waveform distortion	tSD	283	Waveform distortion
Controller Distortion	TSAMP_BUFIO	350	The minimal data width required in the IOB of FPGA
	DELAY Tap Jitter	336	The jitter induced by delay tap
Timing Margin	Timing margin		

Write Cycle Timing Analysis (fast)

In fast mode, the simulated eye diagram of write cycle is as follow.

data valid window :1.426ns

waveform distortion :0.074ns

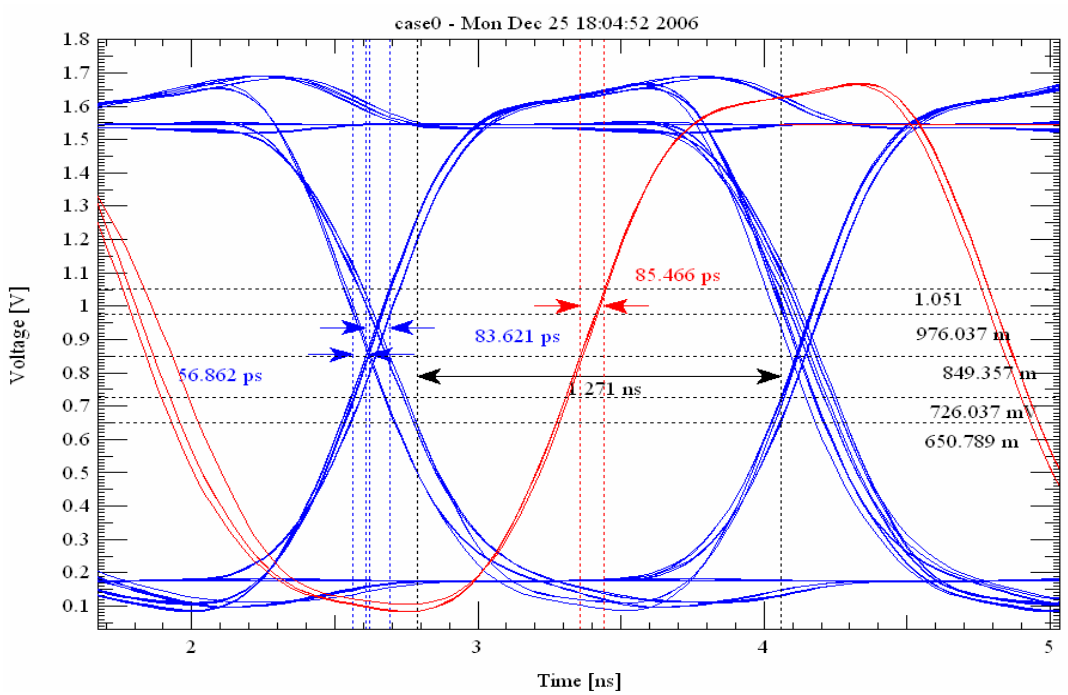


Write Cycle Timing Analysis (slow)

In slow mode, the simulated eye diagram of write cycle is as follow.

data valid window:1.271ns

waveform distortion: 0.229ns



Write Cycle Timing Calculation

	Timing Parameter	Time (ps)	Time (ps)	Time(ps)
Controller Timing Parameters	T_{CK}	3000		
	T_{DCD}	150		
	T_{DATA_OUT}	1350		
	$T_{Package\_skew}$	30	30	30
	Tjitter	50	50	50
	Tclock _skew_ max	50	50	50
	Tclock _out _phase	140	140	140
DDR2 Memory Parameters	*Tsetup*	200	200	
	Tholdup	238		238
Waveform Distortion	*Tsd*	229		229
Timing Margin	Total uncertainties		470	737
	Timing margin	143		

Drawbacks

- **Inefficient**

The method of DDR2 timing analysis above is a time-consuming and manual procedure. Simulations are run on each net to generate eye diagrams, through which signal integrity and timing characteristics of an individual net can be evaluated.

- **Sometimes meaningless**

The simulation is meaningless, when the margin is negative.

The DDR2 timing analysis is made under absolute worst-case conditions. Owing to the worst-case prediction of all parameters applied to a single net, the margin analysis method described above may predict a failure when in fact the actual design is sound.

Deficiencies of IBIS3.2

IBIS3.2 model has the following several deficiencies for timing analysis of DDR2:

1. The IBIS3.2 model is not programmable, so that it is difficult to implement a flexible simulation and process slew rate table automatically.

2. The IBIS3.2 model is not flexible enough to accommodate new techniques.

3. The IBIS3.2 model is a parameterized model, not able to process the simulated waveforms.

IBIS4.2 Model Meets the Challenges

Randy Wolff 、Gary L. Pratt. P.E and their collaborators solve these problems in IBIS4.2 model. IBIS4.2 is an extension of IBIS3.2.

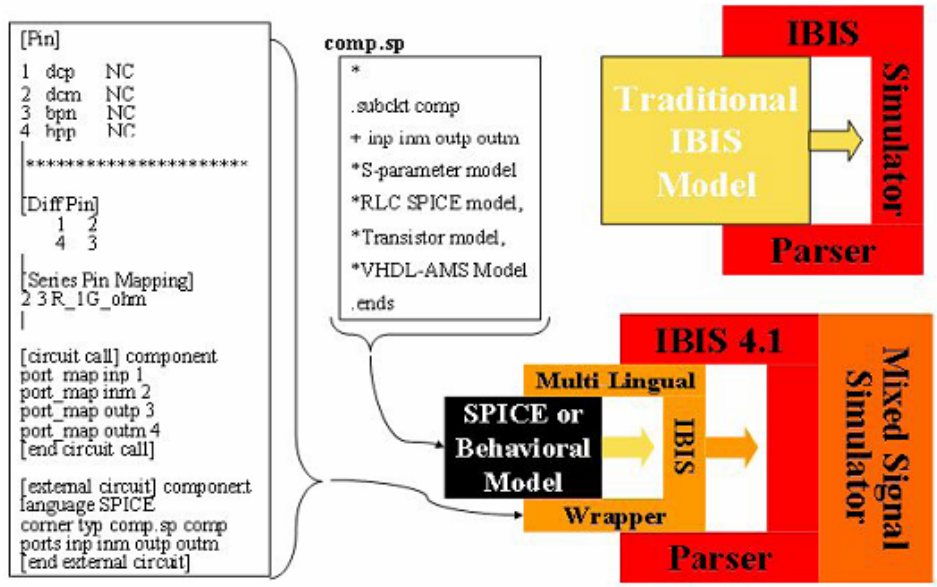
- **IBIS4.2/AMS multiplexer permits the simulator to select between an IBIS model with ODT and one without ODT;**
- **Supports seamless switch of read cycle and write cycle simulation;**
- **Through running a section of VHDL code, the simulator is able to measure the slope of relevant signals;**
- **Automatically determines the corresponding value in the slew rate table;**

It’s a good solution to problems we are facing now.

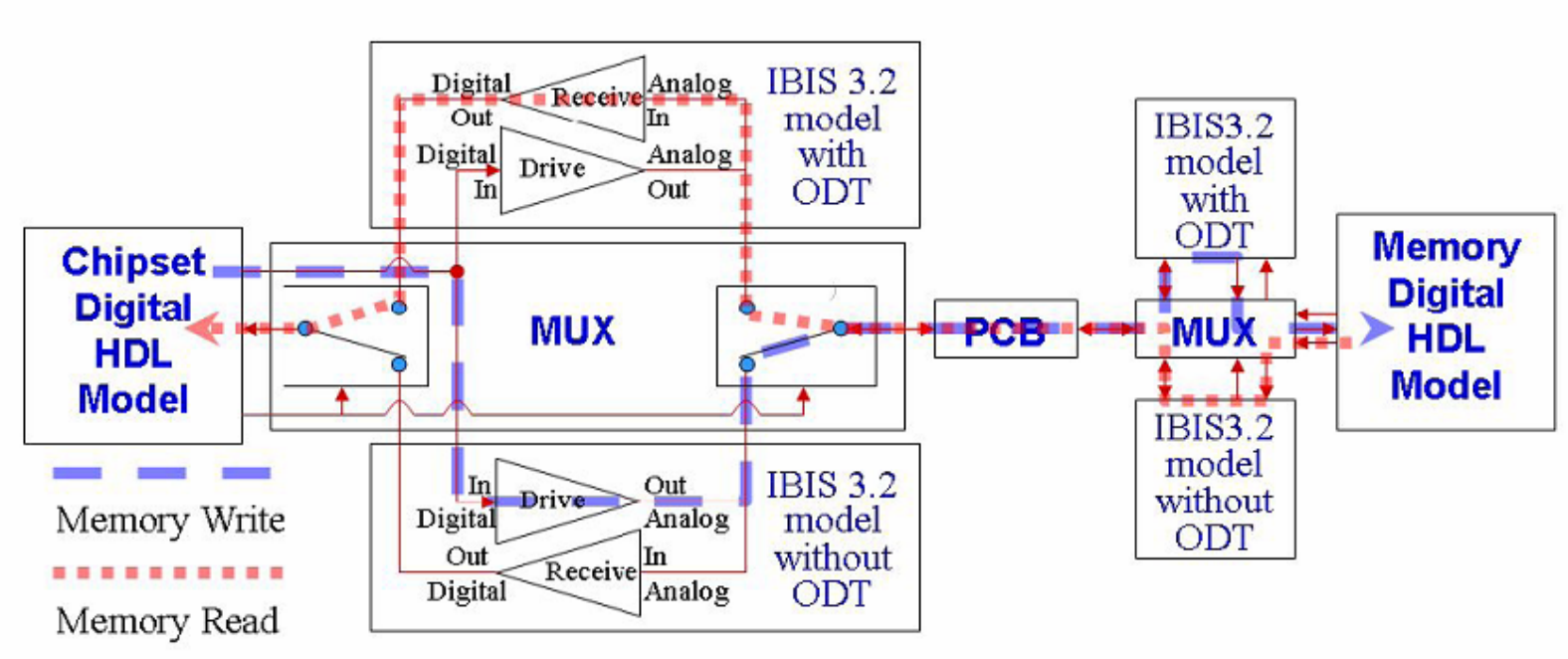
IBIS 4.2 for Timing Analysis

Using the IBIS4.2 model ,we can make the following measurements automatically:

- **Automatic overshoot area measurements**
- **Automatic setup and holdup timing measurements**
- **Slew-dependent timing calculation**



One Solution of ODT Circuit

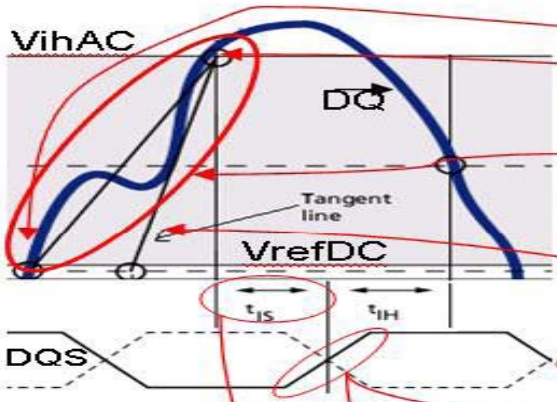


VHDL-AMS Code for Multiplexer

```
if ODTmux = '1' use  -- Connect the ODT driver
   v_a_signal == v_ODT_a_signal;
   i_a_signal == -i_ODT_a_signal;
   -- Keep the non-ODT driver C_comp charged
   v_noODT_a_signal == a_signal'reference;
else  -- Connect the non-ODT driver
   v_a_signal == v_noODT_a_signal;
   i_a_signal == -i_noODT_a_signal;
   -- Keep the ODT driver C_comp charged
   v_ODT_a_signal == a_signal'reference;
end use;
break on ODTmux;

-- ODTmux bit chooses between ODT and non-ODT digital receiver
d_receive <= ODT_d_receive when ODTmux='1' else noODT_d_receive;
-- ODTmux bit chooses between ODT and non-ODT digital driver
-- Set non-used driver output to Z to keep it charged
ODT_d_control <= d_control when ODTmux='1' else 'Z';
noODT_d_control <= d_control when ODTmux='0' else 'Z';
```

Slew Rate Table Calculation




```
-------------------------------------------------------------------
-- measure the setup time tangent
-------------------------------------------------------------------
wait until VREFDC;  -- wait for a crossing
max_slope:=0.0;data_point_cntr:=0;setup_crossing<=0.0*sec;
while not vix_ac'event loop -- store data points til vix_ac
   data_point_v(data_point_cntr) := Vin'reference;
   data_point_t(data_point_cntr) := now;
   wait on vix_ac, ASP; -- wait for next event
   data_point_cntr := data_point_cntr + 1;
end loop;   -- go on to find the maximum slope
setup_crossing <= now;
for i in min_slope to data_point_cntr-1 loop
   slope := (crossing_point_v - data_point_v(i)) /
            (crossing_point_t - data_point_t(i));
   if slope > max_slope then max_slope:=slope; end if;
end loop;
setup_slope <= max_slope;
```

DQ Slew Rate (V/ns)	DQS, DQS# Differential Slew Rate																	
	4.0 V/ns		3.0 V/ns		2.0 V/ns		1.8 V/ns		1.6 V/ns		1.4 V/ns		1.2 V/ns		1.0 V/ns		0.8 V/ns	
	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH	Δ^tDS	Δ^tDH
2.0	125	45	125	45	125	45	–	–	–	–	–	–	–	–	–	–	–	–
1.5	83	21	83	21	83	21	95	33	–	–	–	–	–	–	–	–	–	–
1.0	0	0	0	0	0	0	12	12	24	24	–	–	–	–	–	–	–	–
0.9	–	–	–11	–14	–11	–14	1	–2	13	10	25	22	–	–	–	–	–	–
0.8	–	–	–	–	–25	–31	–15	–19	–1	–7	11	5	23	17	–	–	–	–
0.7	–	–	–	–	–	–	–31	–42	–19	–30	–7	–18	5	–6	17	6	–	–
0.6	–	–	–	–	–	–	–	–	–43	–59	–31	–47	–19	–35	–7	–23	5	–11
0.5	–	–	–	–	–	–	–	–	–	–	–74	–89	–62	–77	–50	–65	–38	–53
0.4	–	–	–	–	–	–	–	–	–	–	–	–	–127	–140	–115	–128	–103	–116

INTERPOLATION

```
-- Check the setup timing
check_setup: process (dqs_slope'transaction)
   variable derating : time := 0*sec;
   variable expected, actual : line;
begin
   if last_setup_slope_transaction > last_dqs_slope_transaction then
     derating := interp_derating_table(dqs_slope, setup_slope, SETUP);
     if setup_time = 0.0*sec then
        FILE_REPORT_ELECTRICAL (now,REF," Setup Violation -- DQ transistion in progress during DQS event");
     elsif derating /= time'left then
        if (now-dqs'last_event) - setup_time < tDS_base+derating then
           FILE_REPORT_TIMING (REF, now, "Setup", "FAIL", tDS_base+derating,
             (now-dqs'last_event)-setup_time, ((now-dqs'last_event) - setup_time)-(tDS_base+derating));
        else
           FILE_REPORT_TIMING (REF, now, "Setup", "PASS", tDS_base+derating,
             (now-dqs'last_event)-setup_time, ((now-dqs'last_event) - setup_time)-(tDS_base+derating));
        end if;
     end if;
   end if;
   last_dqs_slope_transaction <= now;
end process;
```

```
  Refdes,   Event Time, Type, Results, Expected,   Measured,   Margin
U1 N3, MWE, 16.818 ns, Setup, PASS,   0.0553ns, 1.104ns,   1.0486ns
```

Output Results

Electrical results

Time	Receiver Pin(s)	Net(s)	Violations/Errors
0.003 ns	U1 J7 H8	MDQS0 and MDQS0#	Violation - DQS Failed to Meet TDQSx Minimum Pulse Width Limit
0.003 ns	U1 J7 H8	MDQS0 and MDQS0#	Violation - DQS exceeded VIXACMIN at Differential Crossing
40.92 ns	U1 E7 D8	MDQS1 and MDQS1#	Violation - DQS Non-Monotonic
41.04 ns	U1 J7 H8	MDQS0 and MDQS0#	Violation - DQS Non-Monotonic
58.407 ns	U1 G1	MDQ11	Setup Violation -- DQ transistion in progress during DQS event
58.407 ns	U1 E3	MDQ1	Setup Violation -- DQ transistion in progress during DQS event

timing results

Refdes/net	Event Time	Type	Results	Expected	Measured	Margin
U1 T3 MA5	30.579 ns	Setup	PASS	0.146751 ns	1.071 ns	0.924249 ns
U1 N3 MWE	78.579 ns	Setup	PASS	0.160759 ns	1.113 ns	0.952241 ns
U1 T8 MA4	72.579 ns	Setup	PASS	0.145127 ns	0.402 ns	0.256873 ns
U1 R2 MA10	15.579 ns	Setup	PASS	0.156969 ns	0.423 ns	0.266031 ns
U1 P7 MCAS	95.43 ns	Hold	PASS	0.266398 ns	0.276 ns	0.009602 ns
U1 P7 MCAS	21.996 ns	Hold	PASS	0.249038 ns	0.282 ns	0.032962 ns
U1 P1 MBA2	15.579 ns	Setup	PASS	0.139647 ns	0.417 ns	0.277353 ns
U1 U2 MA7	15.579 ns	Setup	PASS	0.119884 ns	0.399 ns	0.279116 ns

Warnings/Errors results

RefDes/Net	Warning/Error
U1 T3 MA5	Warning - Runt Pulse at time: 71688
U1 R2 MA10	Warning - Runt Pulse at time: 71730
U1 G3 MDQ15	Warning - Excessive Hold Slope at time: 72480
U1 E1 MDQ14	Warning - Excessive Hold Slope at time: 72495
U1 E9 MDQ8	Warning - Runt Pulse at time: 112983
U1 F2 MDQ10	Warning - Runt Pulse at time: 113013
U1 J9 MDQ1	Warning - Excessive Setup Slope at time: 125673
U1 K2 MDQ7	Warning - Excessive Setup Slope at time: 126852

Advantages

- **Flexible**

  The IBIS4.2 model is programmable, so that the model is flexible and appropriate to new technology.

- **Real-time**

  The margin is properly measured on every edge in the method of timing analysis, which takes the influence of slew rate table into consideration.

- **Automatic**

  The procedure of timing analysis is automatic and prompt. The output results are reasonable and simple.

Requirements from System Vendors

The IBIS4.2 model is a good solution to the high rate bus of memory timing analysis,

1. There is still few EDA vendors to fully support the IBIS4.2 model;

2. Due to no spice2AMS tool in place, the majority of IC vendors do not have a schedule to provide the IBIS4.2 model;

3. The results are consistent , when we use the same IBIS model on different platforms;

4. The IBIS model should be accurate enough to depict electrical characteristics of chip properly;

5. Building the package model should be paid more attention to as the bus rate increases;

The development of IBIS needs our efforts！