Contents

Figures

Tables

1 Introduction

This document describes the simulation processes that can be used to do signal integrity and timing analysis of memory interfaces. This document can be used as a companion to project-specific analysis reports, to explain the methods used in the analyses, and define the terminology.

1.1 Related Documents

1.1.1 Standards

Table 1-1 lists the standards that govern memory components. Released standards may be obtained from www.jedec.org; there is also information in the Qualcomm-internal site go/standards.

Table 1-1 JEDEC Standards

JEDEC specification available at www.jedec.org	**Interface**
JESD79F	DDR
JESD79-2E	DDR2 “PCDDR2”
JESD79-3D	DDR3 “PCDDR3”
JESD209	LPDDR
JESD209-2	LPDDR2
JESD209-3	LPDDR3
JESD209-4	LPDDR4
Unpublished as of Oct. 2018	LPDDR5

1.1.2 Qualcomm-Internal

- QCAE/PSIG Memory SI Analysis Kit
- Study of DRAM package characterization
- Memory technology for signal integrity engineers
- Modeling distributed pad parasitics in DRAM IOs

2 Definitions

The nets in a DDR interface can be grouped into timing groups, sometimes called strobe groups. A **timing group** consists of a strobe or clock, usually differential, and a set of signals that use that strobe as a timing reference. There are timing constraints between each of the signals and the strobe.

3 Design Data

3.1 Strobe groups

Table 3-1 lists an example of the strobe groups present in an LPDDR2 or LPDDR3 32-bit channel. The CK/DQS strobe group allows expressing timing constraints between the other strobe groups. A project may include more than one channel, but there are no timing constraints between channels.

Table 3-1 Example of strobe groups for a 32-bit LPDDR2 or LPDDR3 channel

Strobe group	Strobe nets	Signal nets
Byte 0	DQS[0]_t, DQS[0]_c	DQ[7:0]
Byte 1	DQS[1]_t, DQS[1]_c	DQ[15:8]
Byte 2	DQS[2]_t, DQS[2]_c	DQ[23:16]
Byte 3	DQS[3]_t, DQS[3]_c	DQ[31:24]
Command/Address	CK_t, CK_c	CA[9:0]
Control	CK_t, CK_c	CS, CKE
CK/DQS	CK_t, CK_c	DQS[0]_t, DQS[0]_c, DQS[1]_t, DQS[1]_c, DQS[2]_t, DQS[2]_c, DQS[3]_t, DQS[3]_c

Table 3-2 lists strobe groups in an LPDDR4 or LPDDR4X channel.

Table 3-2 Example strobe groups for a 16-bit LPDDR4 channel

Strobe group	Strobe nets	Signal nets
CA	CK_t, CK_c	CA[5:0], CS[1:0]
Byte 0	DQS[0]_t, DQS[0]_c	DQ[7:0], DMI[0]
Byte 1	DQS[1]_t, DQS[1]_c	DQ[15:8], DMI[1]

In LPDDR4, CKE is not simulated since it has no timing requirements.

3.2 PHY type and timing

The DDR SI analysis process can be applied to systems that have many different circuit designs, affecting which signaling and timing metrics are relevant. The DDR-PHY blocks developed by Qualcomm are grouped into generations. Each generation has its own format for timing and signaling budgets, so the generation that is used affects the metrics that should be returned from SI analysis. Table 3-3 lists the DDR PHY generations, where they can be applied, and what they imply for analysis techniques.

Table 3-3 Implications of DDR PHY Generation

DDR PHY Gen	Applicable standards and speed bins	Analysis techniques
1	▪ LPDDR2 up to 533 MHz ▪ LPDDR3 up to 933 MHz ▪ PCDDR3/3L up to 600 MHz	▪ Generic or detailed memory package model ▪ Group-wise skew spans
2	▪ LPDDR3 up to 933 MHz	▪ DRAM package characterization using detailed DRAM package models ▪ Per-bit skew spans
2.5	▪ LPDDR2 up to 533 MHz ▪ LPDDR3 up to 933 MHz ▪ PCDDR3/3L up to 1066 MHz	▪ DRAM package characterization using detailed DRAM package models ▪ Per-bit skew spans
3	▪ LPDDR3 up to 933 MHz ▪ LPDDR4 and LPDDR4X up to 2133 MHz	▪ Detailed DRAM package models ▪ Per-bit skew spans

3.2.1 Gen-3-PHY timing paradigm

The Gen-3 (and later) DDR PHYs use bundled closed-loop timing with per-bit skew compensation. Figure 3-1 illustrates the key results from SI simulation. Each data (or command/address/control) net will have some aperture time when it is valid at the receiver. The

SI analysis should find the minimum aperture width for the link jitter budget. For these PHYs we calculate triggered apertures for each net and report the minimum over all nets. We use triggered apertures so that if there is any correlation between the signal jitter and the strobe jitter, the correlated portion of the jitter does not close the aperture. We use per-net apertures because the PHY will train out the net-to-net skew.

It is also necessary to check the static skews between the strobe (or clock) and each data (or command/address/control) net and report the min and max skews for the per-bit deskew budget.

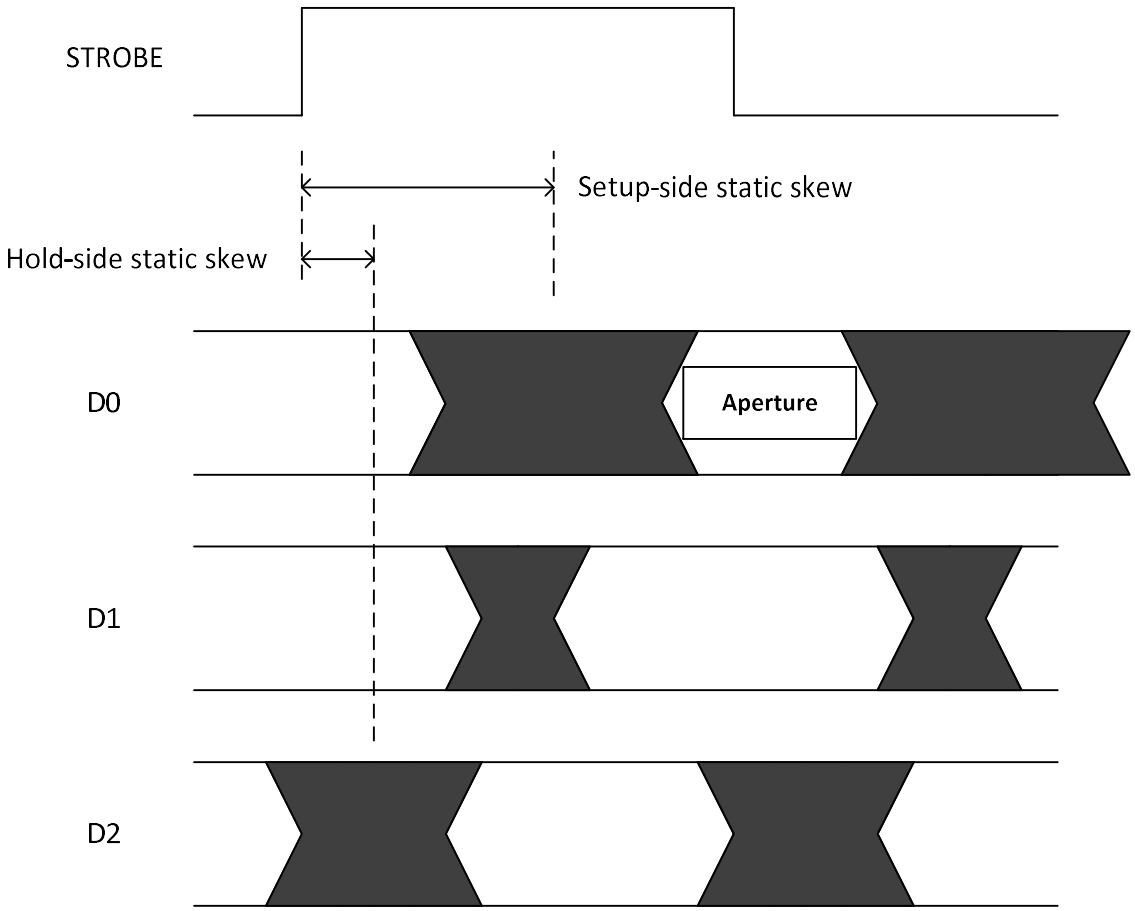

Figure 3-1 Timing paradigm

In Qualcomm LPDDR4X chips, training of delay and Vref for CA and CS is generally disabled. So a link might fail due to excessive static skew (high or low) between CK and CA or between CK and CS.

3.3 Signaling

3.3.1 LPDDR4X

The input thresholds (VdIVW and VcIVW) for LPDDR4X are the same as for LPDDR4. Even though the power-supply voltage is lower in LPDDR4X, the signal swings are similar.

The VIHL_AC specification applies to the single-ended signals: DQ, DMI, CS, CA. It does not apply to CK or DQS.

3.4 Vref training

Qualcomm chipsets that support LPDDR4X and LPDDR5 train Vref at each receiver individually. That is, we can assume per-net Vref training in read mode for LPDDR4X and LPDDR5 chipsets.

4 Analysis Techniques

4.1 Mode coverage

4.1.1 LPDDR4

Unterminated mode can be treated like terminated mode in terms of metric extraction and reporting. The only differences are in the HSPICE netlists.

For signoff analysis a pattern that is just one long burst is used. Burst stops and starts, and bus-turnaround conditions are not routinely analyzed.

4.1.2 LPDDR5

The LPDDR5 spec allows for transactions (low-speed, unterminated) to proceed even when VDDQ is ramping between its 0.5 V and 0.3 V set points for VDDQ.

The maximum VDDQ slew rate is 20 mV/µs. If we run unterminated at up to WCK=2133 MHz (UI=267.5 ps), a 16-bit burst would take 4.28 ns. In that burst time the VDDQ voltage might change by as much as 86 µV. So if we can pass metrics at both 300 mV and 500 mV, we can neglect the slewing.

4.2 Skew analysis

4.2.1 Generic Memory Package Model

In projects with the Gen-1 DDR PHY we report timing under what we can call a "generic memory package model" assumption. That is, in the SI simulations, a generic uncoupled package model is used and the skew is computed from die pad to die pad. This skew is then added to the JEDEC ac timing parameters. In this way, the skew due to crosstalk in the memory package is part of the JEDEC budget. The skews due to crosstalk in the controller package, and reflections, are included in the "SI-simulated" part of the budget.

Table 4-1 lists typical parasitics from the package-model section of a DRAM IBIS file.

Table 4-1 Typical package parasitics from SKHynix IBIS file

Parameter	Typical Value
R_pkg	0.41 Ω
L_pkg	1.88 nH
C_pkg	0.85 pF

The LC values can be converted into impedance and time delay for an equivalent transmission line. For a net with two loads in the package, it was assumed that the branches to the individual DRAMs were about 5 ps long. The schematic in Figure 4-1 and parameter list in Table 4-2 resulted. The package resistance was neglected.

Table 4-2 Parameters for generic memory package model

Parameter	Typical Value
Z0	47 Ω
TD1	35 ps
TD2	5 ps

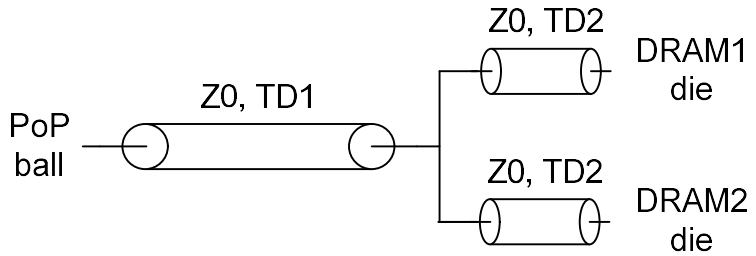

Figure 4-1 Schematic for memory package model

An alternate method is described in the next section.

4.2.2 DRAM package characterization

As an alternative to the generic-memory-package method, another method of doing timing budgeting exists. This method, which we can call "DRAM package characterization," removes some pessimism and clarifies the boundaries in the timing budget.

Skew calculation needs to be done from die pad to die pad to be meaningful. To create a timing budget using a die-pad-to-die-pad skew calculation, we need to add it to a die-level spec for the on-chip timing.

Since memory vendors usually do not provide die-level ac parameters, we need to infer those parameters from the guaranteed package-level ac parameters and the memory package models.

To do this the SI/timing simulations can be done in two steps:

1. Characterize the memory package by replicating the test setups and computing package skew values for data-write setup and hold, data-read setup and hold, and command-address setup and hold.
2. Find the total system skew by simulating top and bottom package together, using the same top-package model as in the characterization step.

4.2.2.1 Details of Characterization Simulations

For characterizing memory inputs (data-write and command-address), the memory component is to be driven at its balls by a linear voltage source with a fixed output impedance, swing, and slew rate. The values chosen for the tester-IO parameters will be project-dependent.

For characterizing memory outputs (data-write), the memory component is loaded at its balls by the JEDEC-specified reference test load. This load is also used for the timing reference circuits.

4.2.2.2 Compared to using a generic memory package model

Advantages:

- Including power noise is straightforward.
- Does not require assumptions about a generic memory package model.

Disadvantages:

- Requires accurate, vendor-specific coupled DRAM package models.
- Subject to any errors in replicating the test setup.
- Memory vendors refresh their designs every 9 to 12 months, so die-level and package-parasitic parameters will vary from device to device, maybe several times over the lifetime of a memory-controller product.

4.2.2.3 Use of total skew rather than just dynamic skew

For standards like LPDDR3 and DDR3, which do not provide for per-bit training to remove the static components of skew, both the static and dynamic components should be subtracted in order to find the die-level memory ac parameters.

Figure 4-2 illustrates with some timing diagrams for the tDQSQ parameter. The top diagram shows the idealized waveforms, if there were no skew at all.

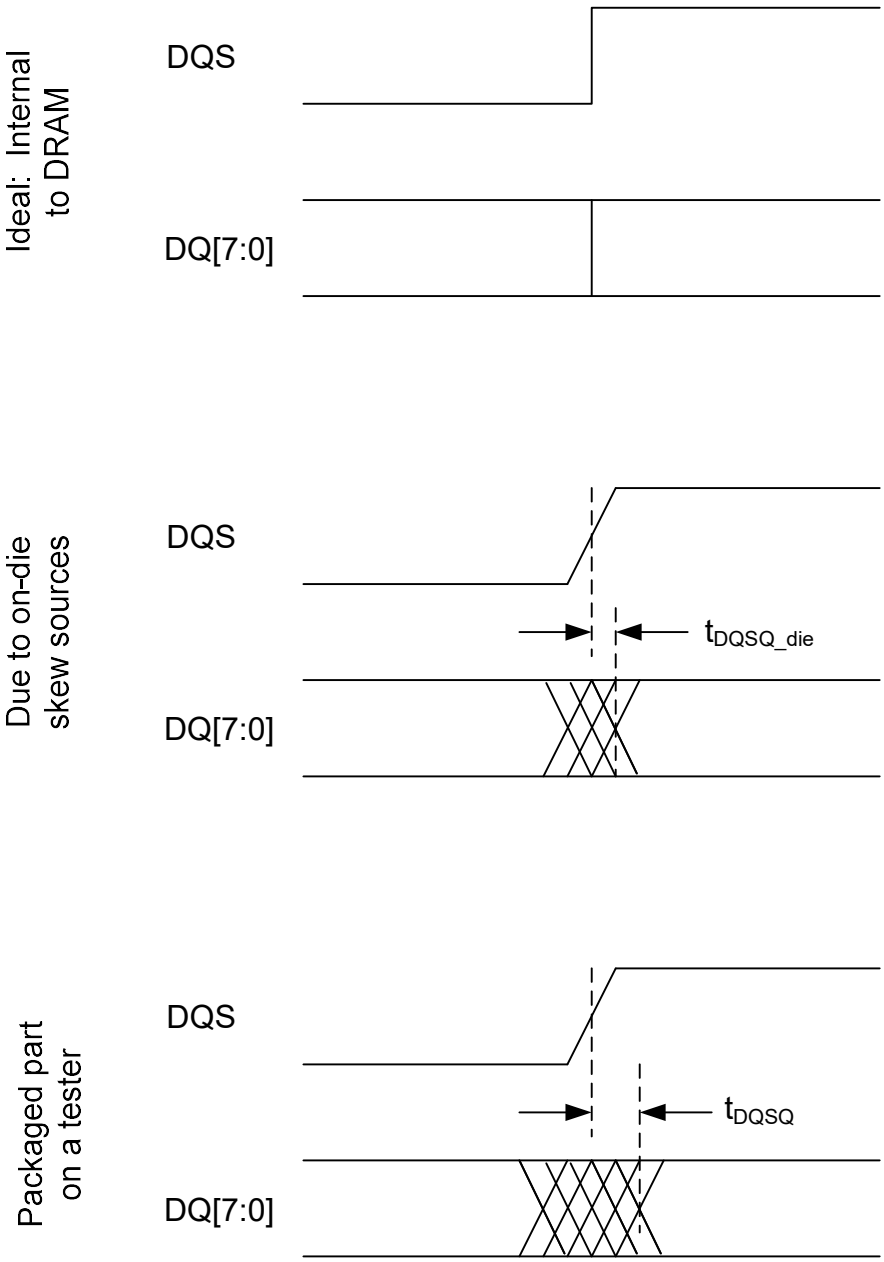

Figure 4-2 Components of tDQSQ parameter

The middle diagram shows that some part of tDQSQ is due to on-die sources, such as skew in on-die routing or ISI in the output cell. The bottom diagram is the definition of tDQSQ, which reflects both die and package contributions. To compute the die contribution, we need to simulate the package only and subtract the worst-case total skew. Similar arguments apply for the other DRAM ac parameters.

4.2.2.4 PVT corners

We are really derating the DRAM component's timing parameters, so we know that those timing parameters are guaranteed across corners. So we can use results from whatever PVT corner gives

the largest skew in DRAM-package characterization. We do not have to match the PVT corner in the characterization simulation with the PVT corner in the system simulation.

4.3 Vref training (Vcent)

Some standards (PCDDR4 and LPDDR4) provide for training the Vref level per component. If Vref training is done, the computation of signaling metrics (such as apertures and slew rates) should be done taking the trained Vref into account.

The CKE net is an exception; Vcent does not get applied to CKE.

4.3.1 Limitation

Strictly speaking, Vcent should be a property of an entire DRAM component (per signal net type). However, the MSAK and MERiT are set up to compute and use Vcent on a per-strobe-group basis. This leads to MERiT's calculations of eye height and noise margin being a bit optimistic. As an example, consider a system where byte 0 was found to have a Vcent of 200 mV and a noise margin of 80 mV. Byte 1 in that same system had a Vcent of 190 mV according to MERiT. In actual operation, the two bytes would have to use the same Vcent. If that Vcent came out to, say 195 mV, then byte 0's noise margin might really be only 75 mV.

5 Modeling

5.1 Explicit local references in interconnect models

In most cases, interconnect models used in circuit simulation should be wrapped in a wrapper with nodes exposed for references at each location.

Figure 5-1 shows an example of subcircuit wrappers for a partial-element model (on the left) and for a loop model (on the right). In the subcircuit for the loop model, a zero-Volt voltage source is used to rename the vss node.

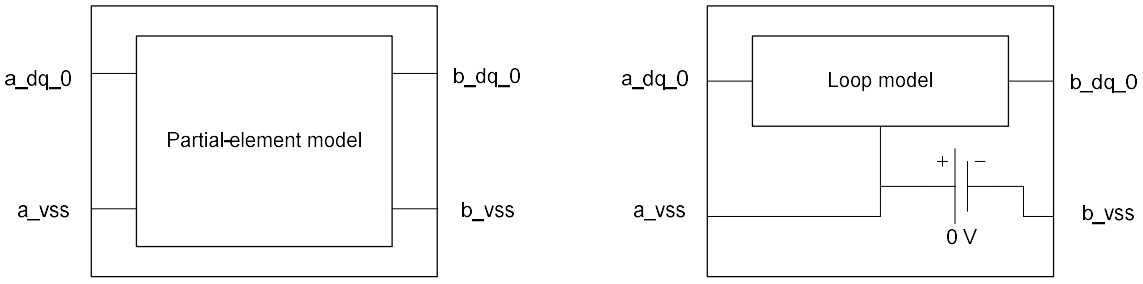

Figure 5-1 Example of subcircuit wrappers for different model types

The node list seen by the upper-level circuit is the same in either case.

5.2 Board models for prelayout analysis

In prelayout analysis, we have no information about routing details in the boards or packages, and no reason to distinguish one signal from another signal of the same type. For example, we can represent any DQ bit as a generic DQ signal.

We do have this information:

- Stackups of boards and packages
- Line impedance targets
- Routing space or pitch targets

We can use that information and a 2D field solver to make coupled-transmission-line models for each signal type.

Figure 5-2 shows how the DQ network might look. It is only necessary to model one victim line, but it should be coupled to as many aggressor lines as appropriate based on the board stackup and floorplan. In the figure, two aggressors are shown. In a board with significant tandem-layer coupling, it may be appropriate to include more than two aggressors. We assume that the aggressor signals are also DQ signals since it's likely that a whole byte will be routed together.

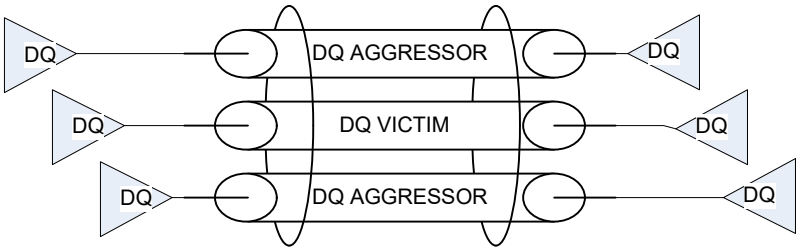

Figure 5-2 Prelayout transmission line model for DQ network

Figure 5-3 shows how the DQS network might look. Here we are assuming that the main aggressors onto the DQS nets are DQ signals. To get differential-mode noise onto the DQS net, the aggressors closer to the DQSP conductor should get a pattern that is the inverse of the pattern for the aggressors closer to the DQSM conductor.

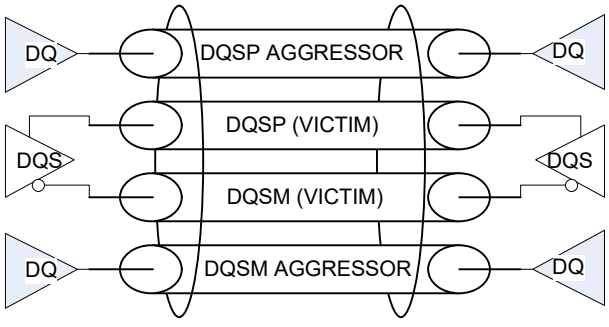

Figure 5-3 Prelayout transmission line model for DQS network

Figure 5-4 shows a coupled transmission line model for a CA network. The aggressors are also CA signals.

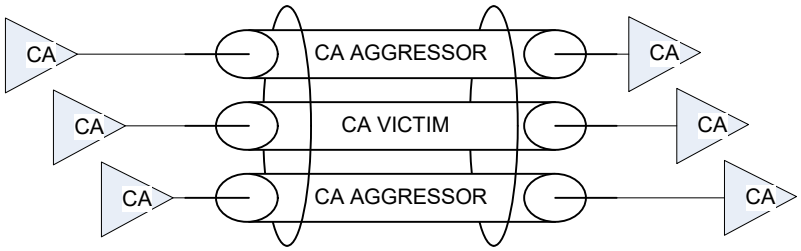

Figure 5-4 Prelayout transmission line model for CA, CS, or CKE network

Figure 5-5 shows a prelayout transmission line model for a clock network. The aggressors are CA signals, and again, the aggressors that are more strongly coupled to the CKP conductor should have a data pattern inverse to the aggressors that are more strongly coupled to the CKM network.

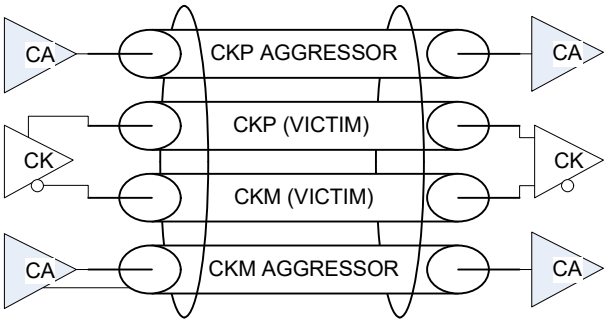

Figure 5-5 Prelayout transmission line model for Clock network

5.3 IO Models

5.3.1 Standard nodes

Figure 5-6 shows the flow of signals (omitting power-supply connections) in an IO subcircuit. The minimum connections are shown on the outside:

- A trigger, which commands the transmitter to drive either high or low
- An enable, which controls whether the transmitter is enabled or disabled
- A die pad, which connects to the package model

If the IO subcircuit includes a receiver, we may have access to the receiver output (labeled core_in in Figure 5-6). This signal is not brought out to the IO subcircuit but it may be probed inside the subcircuit.

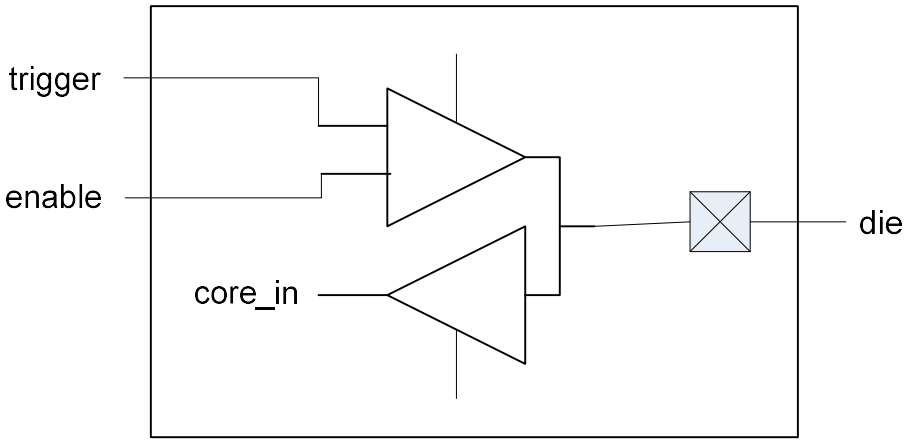

Figure 5-6 Typical signaling nodes in IO subcircuit

5.3.2 Linear drivers for LPDDR4X

Drivers at the DRAM locations for LPDDR4X are parameterized by VOH, pulldown drive strength, and SoC ODT.

To make a generic linear driver model we use a Thevenin-equivalent where the resistance is set by the pulldown drive strength and the open-circuit voltage is set to get the correct output high level when terminated by the SoC ODT.

$$V_{ocv} = \begin{cases} VDDIO & if\ ODT\ not\ used \\ VOH * VDDIO * \left(\frac{PDDS + ODT}{ODT}\right) & if\ ODT\ used \end{cases}$$

5.3.3 IBIS models in HSPICE

Modeling IOs with IBIS should be done with HSPICE's B element. The following keywords should be set to control the interpretation of the buffer's pad capacitance:

```
+ c_com_pc=0.5 c_com_gc=0.5
+ c_com_pu=0.5 c_com_pd=0.5
```

If the IBIS file calls out C_comp components specifically, without also setting C_comp, HSPICE will use the C_comp components called out in the IBIS file. An HSPICE warning will be generated but it can be ignored.

The phenomenon of IBIS overclocking may cause incorrect results in the transient waveform, often seen as transitions that are missed or are much too early or late.

To prevent overclocking, the following B-element keywords may be set:

```
+ ramp_rwf=0 ramp_fwf=0
```

Setting those keywords reduces the accuracy of the simulation because they result in the IBIS V-t curves being ignored. Therefore, they should only be set if an overclocking issue is observed or suspected.

5.4 Package Models

5.4.1 Extracted models

Package models may be partial-element (for example, the output of Ansys Q3D) or loop (for example, the output of Ansys HFSS) models. If partial-element models are used, the reference nodes may be grouped or ungrouped. If grouped reference nodes are used, the project report should indicate which nodes were grouped. If the reference nodes are not grouped, the project report should indicate which layout nodes were used as references for which signals.

5.4.2 Generic memory package models

When generic memory package models are used, the package model should be made with uncoupled, lossless interconnects. Table 5-1 lists parameters and Figure 5-7 shows a schematic of an example generic memory package model.

Table 5-1 Parameters for generic memory package model

Parameter	Typical Value
Z0	47 Ω
TD1	35 ps
TD2	5 ps

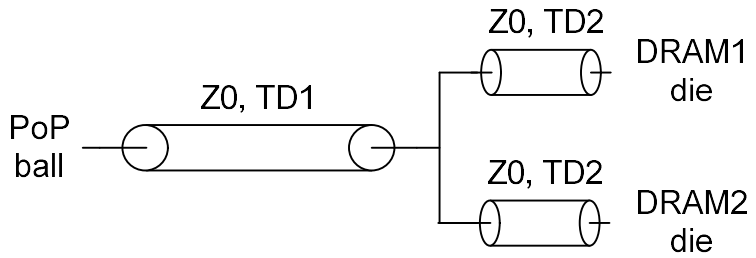

Figure 5-7 Schematic for memory package model

5.5 On-die interconnect models

The networks for on-die interconnects (RDL) are point to point, just connecting the IO pad to the die pad or bondwire.

DRAM packages that contain multiple dies will have multiport nets in the package models, but not in the RDL models.

5.6 Standard clock periods

The memory specifications are written such that tCK (the length of the clock period) is an exact number. For SI simulation, we base all the timing calculations off tCK, except for in LPDDR5, where tWCK is used.

5.7 Stimuli

Stimuli may be generated as individual run-length-limited pseudorandom patterns.

5.7.1 Prelayout

In a prelayout simulation, it's only necessary to observe one signal for each signal type. To model crosstalk, each signal type must have an aggressor. So two independent stimuli are generated: a victim pattern and an aggressor pattern.

For single-ended signals, the aggressors are single-ended signals of the same type, and all aggressors are stimulated with the same pattern.

For differential victims, the aggressors are represented by the corresponding single-ended signal (CA for Clock and DQ for DQS). The aggressors on one side of the victim signal get the aggressor pattern, and the aggressors on the other side get the inverse pattern, so that the differential victim receives differential-mode noise. Using the labels from Figure 5-2, Figure 5-3, Figure 5-4, and Figure 5-5, Table 5-2 lists the stimuli patterns needed for a prelayout simulation. "Time correlation" in calculating skew is discussed in Section 7.6.4 .

Table 5-2 Stimuli types for prelayout simulation

Signal	Stimulus, if skew calculation considers time correlation	Stimulus, if skew calculation does not consider time correlation
DQ VICTIM	Data pattern #1	Data pattern #1
DQ AGGRESSOR	Data pattern #2	Data pattern #2
DQS VICTIM	Strobe pattern	Strobe pattern
DQSP AGGRESSOR	Data pattern #3	Strobe aggressor pattern
DQSM AGGRESSOR	Inverse of data pattern #3	Inverse of Strobe aggressor pattern
CA VICTIM	CA pattern #1	CA pattern #1
CA AGGRESSOR	CA pattern #2	CA pattern #2
CLOCK VICTIM	Strobe pattern	Strobe pattern
CKP AGGRESSOR	CA pattern #3	Strobe aggressor pattern
CKM AGGRESSOR	Inverse of CA pattern #3	Inverse of Strobe aggressor pattern

If time correlation is considered, we need to generate two groups of stimuli (data and CA), each with three uncorrelated sequences (victim, aggressor, and strobe aggressor). If time correlation is

not considered, we can generate three groups of stimuli (data, CA, and strobe aggressor), with a maximum of two uncorrelated sequences (victim and aggressor). The lower number of sequences in a group should lead to a better achievable test coverage.

5.7.2 Postlayout

In a postlayout simulation, each signal gets an independent stimulus. The stimuli can begin with a simultaneous-switching section. Figure 5-8 shows one section of this stimulus. Of the data and CA bits, one bit is chosen to be the victim and every other bit acts as an aggressor. The victim and aggressors get the six-bit pattern shown in Figure 5-8. This pattern includes the four worst cases for skew: at bit time #2 the victim and aggressors are rising and falling together and at bit time #5 the victim and aggressors are switching out of phase. This pattern does not include the worst cases for quiet-line noise.

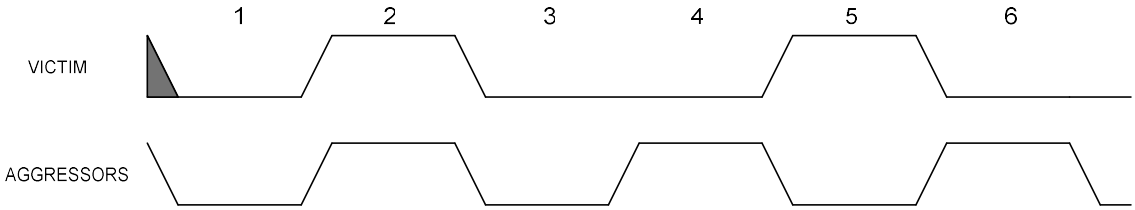

Figure 5-8 Segment of simultaneous-switching stimulus

This sequence is repeated with each data or CA net taking its turn as the victim.

After each signal has had a turn as the simultaneous-switching victim, a random-stimulus section may begin. In this section, each signal gets an independent random stimulus.

5.7.3 DQS-DQ Stimulus

In read mode, DQS and DQ arrive at the memory controller in phase, as illustrated in Figure 5-9.

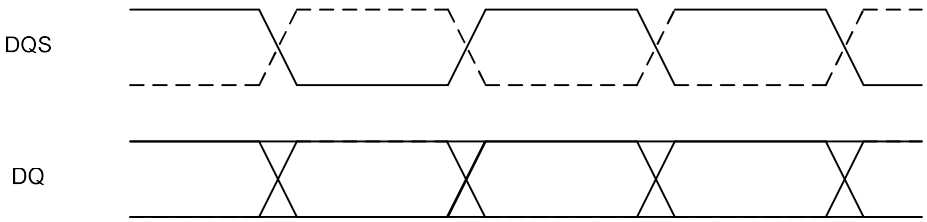

Figure 5-9 DQS and DQ stimulus for read mode

In write mode for standards except LPDDR4, DQS and DQ are out of phase arriving at the memory.

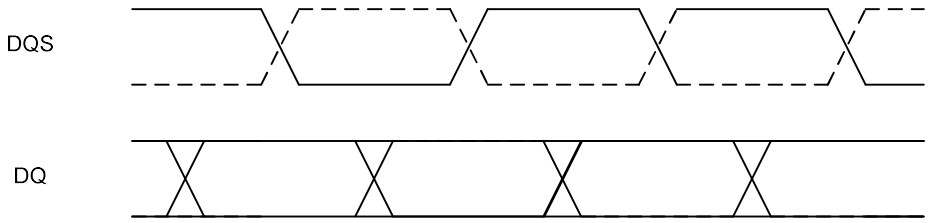

Figure 5-10 DQS and DQ stimulus for write mode

In write mode in LPDDR4, DQS leads DQ by the tDQS2DQ time. The value of tDQS2DQ will be determined at runtime by a training algorithm, so SI analysis must assume that tDQS2DQ can take on any value in its range.

5.7.4 LPDDR2

In LPDDR2, the clock and DQS arrive at the DRAM nominally in phase, so in these simulations as indicated in Figure 5-11, the stimuli are in phase. The timing parameters tDSH and tDSS in the JEDEC specification imply that the falling edge of DQS is to be kept away from the rising edge of CK.

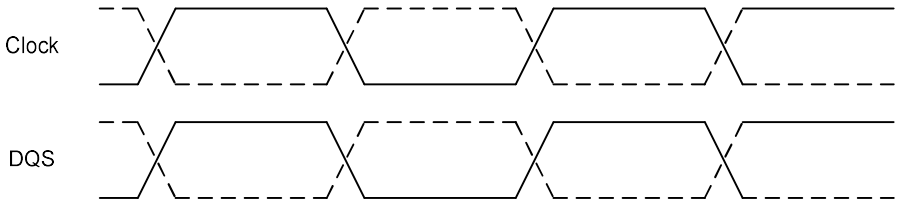

Figure 5-11 Clock/DQS stimulus for LPDDR2

In LPDDR2, the CA signals are double-data-rate and switch one quarter-cycle off from the clock edges. The CS and CKE signals are single-data-rate. They are launched at the clock falling edge and captured at the clock rising edge.[1] Figure 5-12 illustrates a way to schedule their stimuli.

[1] See Figure 22 of JESD209-2B for the CS timing diagram. See Figure 77 of JESD209-2B for the CKE timing diagram.

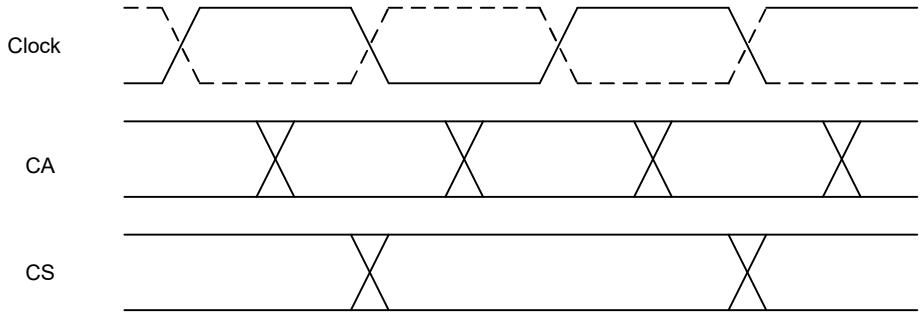

Figure 5-12 CA/CS/CKE stimulus for LPDDR2

CKE has a minimum-pulse-width guarantee; it can only switch once every three clock cycles. For prelayout this distinction is not important but for postlayout the stimulus timing in Figure 5-13 should be used.

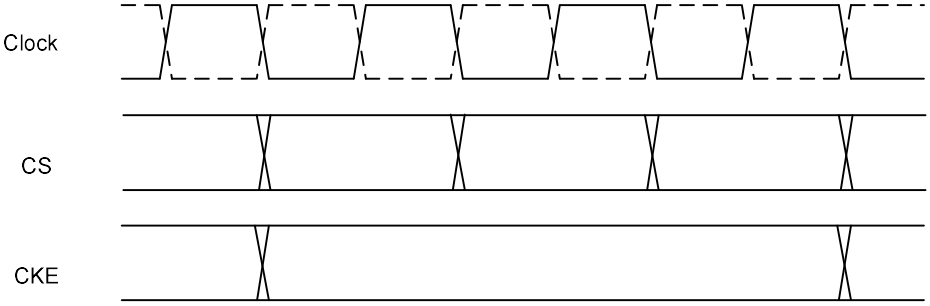

Figure 5-13 CS and CKE stimulus for LPDDR2

5.7.5 LPDDR3

One addition to LPDDR3 that was not present in LPDDR2 is the preamble toggle for DQS when starting a write burst. Figure 5-14 shows a timing diagram. The DQS rising edge marked with an arrow is the first edge used to latch data. Prior to that is a DQS pulse (one rising edge and one falling edge), and before that is some time when DQS was valid low, prior to the start of the write burst.

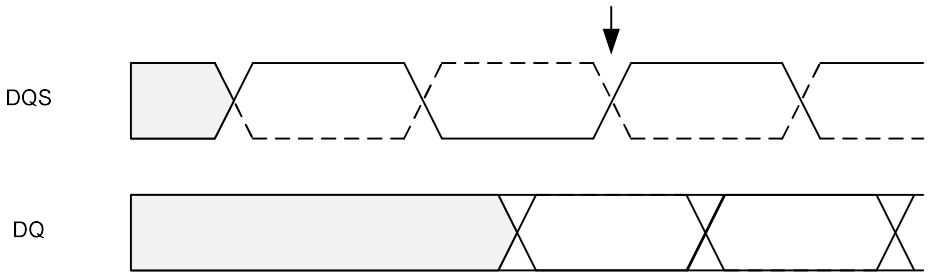

Figure 5-14 LPDDR3 write burst

Figure 5-15 shows a timing diagram for the beginning of a read burst. There is no toggle in the preamble.

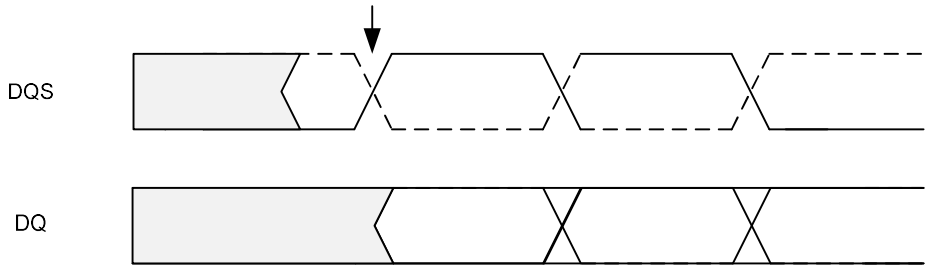

Figure 5-15 LPDDR3 read burst

In our HSPICE simulations we typically neglect the time that signals are floating during bus turnaround and assume that they are actively driven.

5.7.6 PCDDR2

In PCDDR2, the clock and DQS arrive at the DRAM in phase. Figure 5-16 shows a waveform.

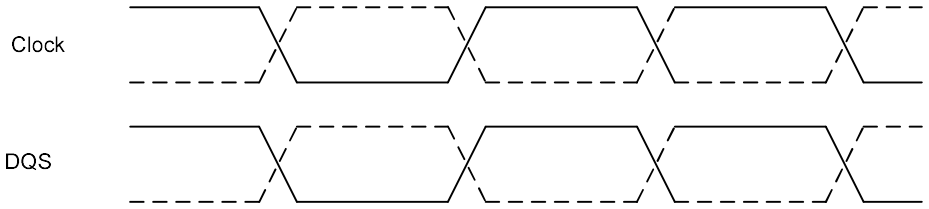

Figure 5-16 CK and DQS stimulus for PCDDR2 write mode

The CA signals are single-data-rate, switching at the falling edge of CK.

DQ and DQS are in tristate at the beginning of each data pattern. To model the preamble, DQS and DQ are driven to valid levels a short time before switching begins. Quick bus turnarounds have not been modeled.

5.7.7 PCDDR3

In PCDDR3, in the absence of write leveling, the clock and DQS arrive at the DRAM in phase.[2] Figure 5-17 shows a waveform.

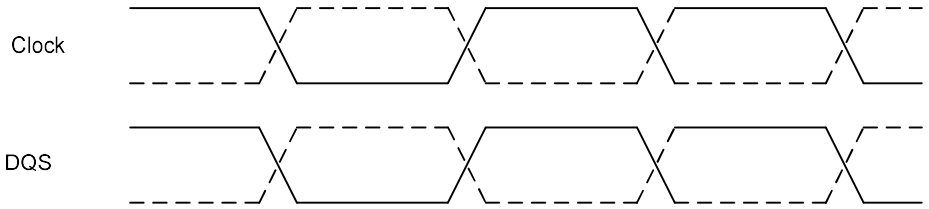

Figure 5-17 CK and DQS stimulus for PCDDR3 write mode

5.7.8 LPDDR4 and LPDDR4X

In LPDDR4 in write mode, DQS leads DQ by the tDQS2DQ time.

[2] See Figure 43 of JESD79-3D.

The JEDEC specification requires a toggling write preamble of 2*tCK. The read preamble is required to be 2*tCK in length, with the first tCK static at the valid logic levels. The second clock cycle may be toggling or not, depending on the contents of a mode register.

5.7.9 LPDDR5

The JEDEC spec allows arbitrary phase between CK and WCK. But in our simulations we will launch CK and WCK in phase; that is, the rising edge of CK is coincident with a rising edge of WCK.

We will arrange stimuli so that CK and WCK are stable before the bus signals start switching. We will not model WCK shutting down at the end of a burst.

We will not check CS signaling or timing when it is in its asynchronous mode (at power-down exit).

The default stimulus patterns for signoff should meet the following criteria:

- CK should start up and be fairly stable before CA and CS start toggling.
- WCK should start up and be fairly stable before DQ and DMI start toggling.
- RDQS should have the appropriate preamble time.
- The parallel signals (DQ/DMI/CA/CS) should stop switching and become quiet before the end of the simulation, so that the strobes can return to steady state before the simulation stops.
- The startup of the data signals (DQ and DMI) should have a pattern that creates a current step (like A5A5).
- The patterns should include victim-aggressor sections.
- The patterns should include pseudorandom sections.
- The WriteCA pattern should stimulate CA6.

Figure 5-18 shows a stimulus pattern for CK, CA, and CS that meets those criteria.

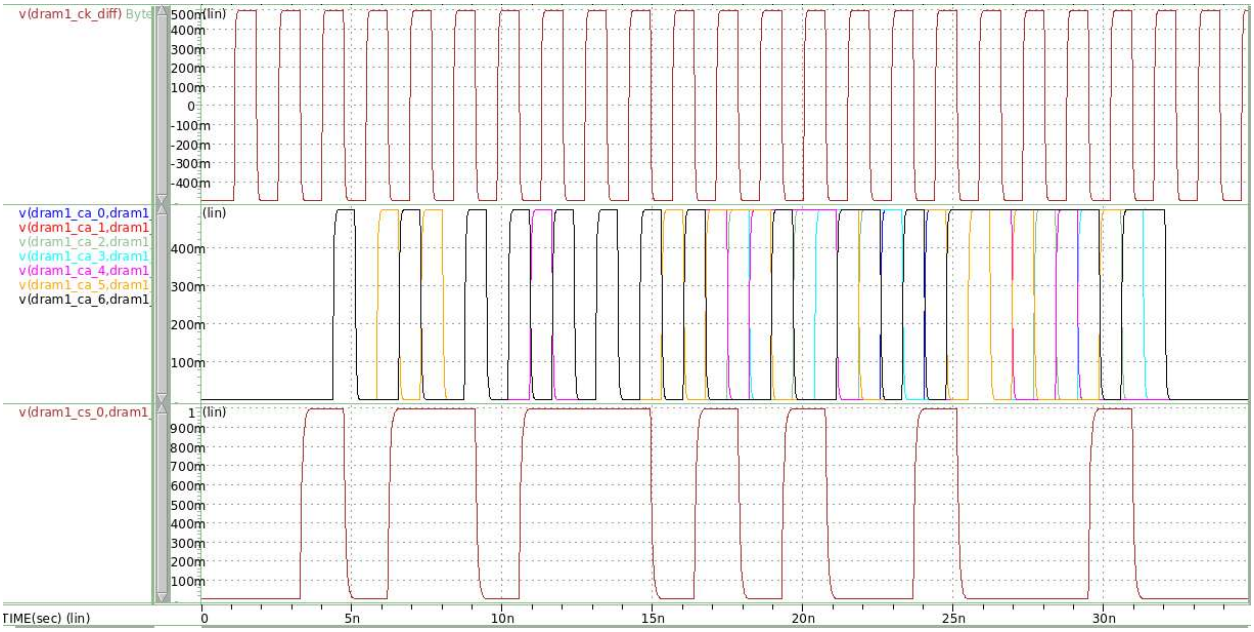

Figure 5-18 LPDDR5 CK/CA/CS stimulut

Figure 5-19 shows a stimulus pattern for WCK and DQ/DMI that meet those criteria.

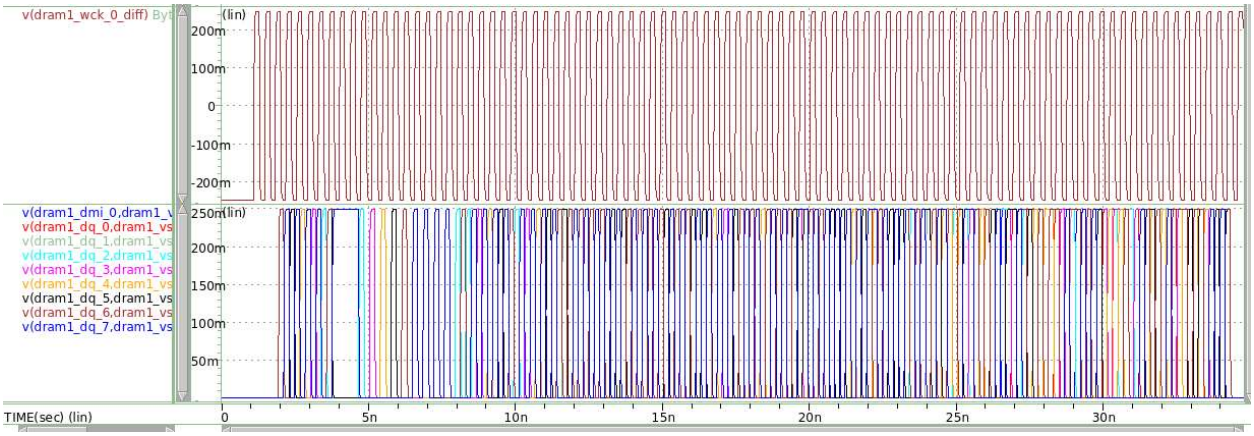

Figure 5-19 LPDDR5 write stimulus

Figure 5-20 shows a stimulus pattern for read (RDQS and DQ/DMI) that meets those criteria.

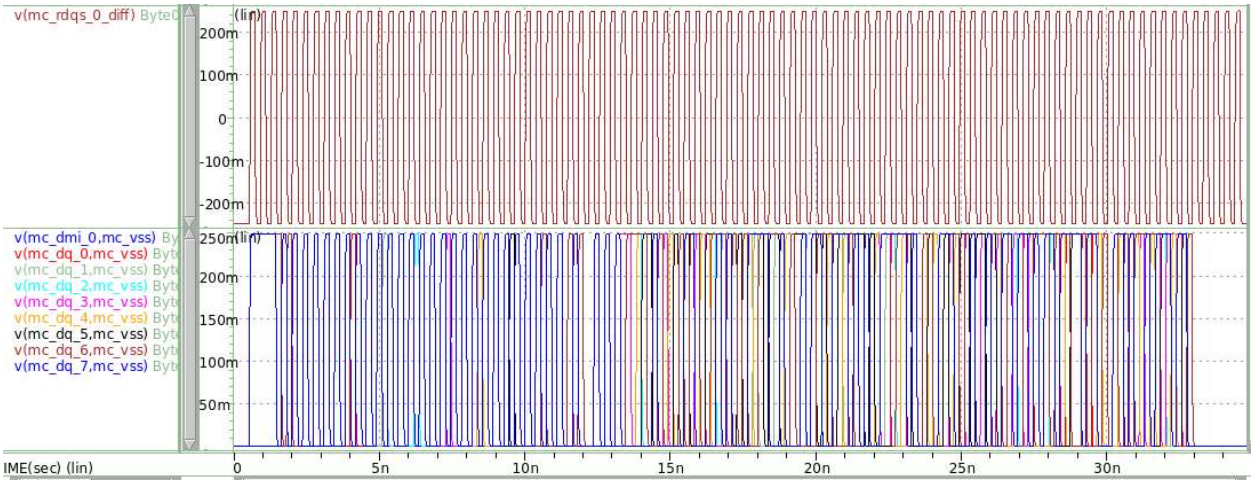

Figure 5-20 LPDDR5 read stimulus

We are allowing CS to have an arbitrary pattern. It is allowed to toggle on consecutive clock cycles and to remain high or low for multiple clock cycles.

5.7.10 (PC)DDR4

In DDR4 systems that have multiple command/address loads, simulating with the simultaneous-switching pattern yields a *lot* of crosstalk noise that makes it pretty much impossible to open the eye. It's recommended to skip the simultaneous-switching section and simulate these groups with just PRBS patterns.

5.7.11 Standard Pattern File

The standard pattern file uses patterns with a simultaneous-switching section followed by a pseudorandom section.

The pseudorandom sections are based on PRBS, but with a twist. The first field in the pattern name is the maximum run length. That is, data_4_1of10 uses patterns that have maximum run lengths of four 1's or four 0's. So that's similar to what you would get with a PRBS4 (and data_7_1of9 is similar to what you would get from PRBS7).

Between the data lanes, different LFSR seeds are used, and circular shifts are used.

5.7.12 Data-bus inversion (DBI)

In LPDDR4, LPDDR4X, and LPDDR5, DBI is of the "DC," not "AC," variety.

When DBI-DC is used, there will be a maximum of four bits held high in each byte, including the DQ and DMI pins.

5.7.13 DQS toggling preamble

Table 5-3 lists the cases when the DQS preamble contains a toggle, per standard.

Table 5-3 Toggling preambles in DQS

Standard	Mode	Toggling preamble on DQS?	Expected values of DQS_wait_edges in MERiT
LPDDR2	All	None	0
LPDDR3	Read	None	0
LPDDR3	Write	Optional	0, 1, or 2
LPDDR4	Any	Optional	0 or greater
LPDDR4X	Any	Optional	0 or greater
PCDDR3 or PCDDR3L	Any	None	0
PCDDR4	Any	TBD	0 or greater

5.7.14 Read burst followed by tristate

One bus-turnaround condition that can be analyzed is ringing on the DQS following a read burst.

For this analysis a read burst is simulated, then the DRAM drivers are disabled after a time compatible with the JEDEC tRPST parameter. Figure 5-21 shows a diagram of the stimuli timing.

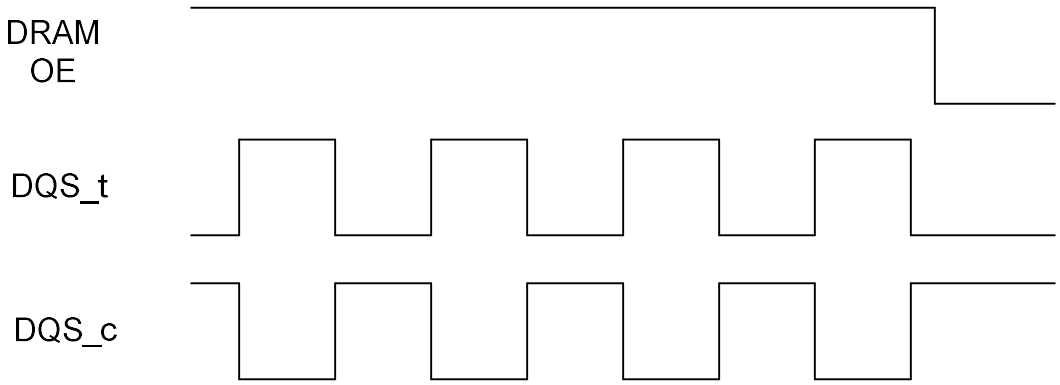

Figure 5-21 DQS and DRAM-enable stimulus for DQS noise margin check

The analysis needs to check the DQS waveforms at the SoC receiver to make sure there will not be any extra toggles due to ringing on the lines.

5.8 Power Distribution and Bypass Capacitors

If power noise is not being considered, it is not necessary to include bypass capacitors on-chip or off-chip. Since the power voltage will not change with time, there will be no current through any bypass capacitors. Any portion of the power distribution network that carries signal return current (possibly including bypass capacitors), however, should be included.

If co-SI/PI simulation is being done, the power networks of the drivers should be modeled. If the receiver power supply is isolated from the driver's power supply, it is not necessary to model the receiver power supply.

5.8.1 LPDDR5

Two of the power supplies used by the SoC IO pads are VDDIO and VDDA. For VDDIO, we model the IO drawing power from a PDN impedance, so we can see the VDDIO noise that results from the current drawn. Furthermore, we will see the impact on the output signal of any noise on VDDIO.

For VDDA we model the VDDA noise as a voltage waveform imposed from the outside; we will see the impact of VDDA noise on the output signal, but we will not model the noise on VDDA due to the current drawn by the IO.

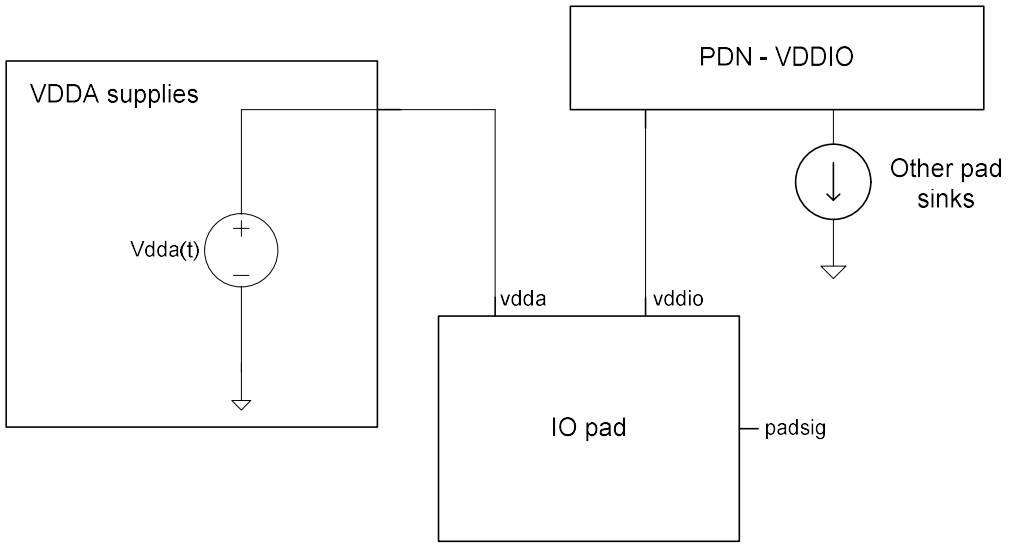

Figure 5-22 Noisy power supplies for SoC IO pads

The "VDDA supplies" and VDDIO PDN boxes can be modeled with subcircuits in user-defined libraries.

5.9 PCB modeling with on-board termination resistors

For most DDR3 and DDR4 channels, the command/address/clock lines will be terminated with discrete resistors on the PCB, and the termination voltage (called "VTT" here) will be some rail other than ground.

Getting the extraction and subcircuit correct takes some special care; options are described in the following sections.

In addition to the comments in the following sections, engineers should take care with the clock termination. There may be a small, randomly-named net between two resistors forming a differential clock termination.

5.9.1 Method 1: Include the resistors in the PCB extraction

This is probably the simplest method, but it has the drawback that it will not be possible to vary the value of the termination resistance in transient simulation.

When using this method, the engineer keeps the on-board resistors enabled in the PCB extraction tool, and places a port somewhere on the VTT net. Then the extracted network-parameter model will include a port for VTT, which should be connected to a power supply of the appropriate voltage.

5.9.2 Method 2: Place ports across the resistors

When using this method, the engineer will disable the resistors in the PCB extraction tool and place a port across each resistor—using the resistor terminals for the port terminals. It is also necessary to place a port on VTT as in method 1.

The extracted network-parameter model will then include ports for each of the resistors. Each of those ports should be connected to a resistor of the appropriate value. Since a port is really a two-terminal pair, one terminal of the resistor will connect to the "port" terminal and the other terminal will connect to the reference node (probably ground).

5.9.3 Method 3: Place ports, referenced to ground, at the resistor terminals

When using this method, the resistors are disabled and two ports are used at each resistor. Then in the circuit simulation, the resistor is connected between the two ports, with no connections between the resistor and ground.

A port on VTT is still needed.

6 Circuit Simulation Setup

6.1 Net types

The net types in DDR standards are listed below, along with the net names used in the various standards.

- Clock net type: ck
- DQS net type: dqs
- Command/address net type: ca, a, ba, cas, ras, we, bg, par, c
- Non-CKE control net type: cs, odt, act
- Clock-enable net type: cke
- Bidirectional data net type: dq, dmi
- Unidirectional data net type: dm

MERiT recognizes the net type “dmi” for nets that have the DM and DBI functions multiplexed together, as in DDR4 and LPDDR4.

6.2 Voltage Levels

6.2.1 LPDDR2

Table 6-1 lists dc voltages used as power supplies or references. The supply voltage at each chip is set by its process, voltage, and temperature (PVT) corner for a particular simulation. If the simulation calls for the driver and receiver to have different supply voltages, an adjustment is made and the receiver’s supply voltage is set to be the same as the driver’s. This is realistic because in an actual system, the dc levels of the supplies would be the same. In Table 6-1, V_{DD} refers to this common supply voltage. Additionally, it is assumed that V_{REF} is always exactly half of V_{DD}, and that V_{TT} always tracks V_{REF}, although the specification does allow some variation.

Table 6-1 System voltages

Symbol	Interpretation	Min	Typical	Max
VDDQ and VDDCA	I/O voltage at DRAM	1.14 V	1.2 V	1.30 V
V_{DDPX}	I/O voltage at memory controller	1.14 V	1.2 V	1.30 V
V_{REF}	Input reference voltage	$0.5*V_{DD}$		
V_{TT}	Termination voltage	V_{REF}		

JEDEC defines two sets of V_{IH} and V_{IL} input logic levels, to accommodate noise due to ringback. The ac levels $V_{IH(ac)}$ and $V_{IL(ac)}$ apply right after a transition. Once a signal passes the high or low ac level, it is considered valid high or low. Once in a high or low state, the signal is allowed to ring back past the ac level but not past the dc level. If the signal crosses the dc level $V_{IH(dc)}$ or $V_{IL(dc)}$, it is no longer a valid logic level. Table 6-2 lists the input dc logic levels and

Table 6-3 lists the input ac levels.

Table 6-2 Input DC logic levels for single-ended signals, LPDDR2-466 to LPDDR2-1066

Symbol	Interpretation	Value
$V_{IH(dc)}$	Minimum value at which dc input is a logic HIGH	V_{REF}+0.13 V
$V_{IL(dc)}$	Maximum value at which dc input is a logic LOW	V_{REF}-0.13 V

Table 6-3 Input AC logic levels, LPDDR2-466 to LPDDR2-1066

Symbol	Interpretation	Value

$V_{IH(ac)}$	Minimum value at which ac input is a logic HIGH	V_{REF}+0.22 V
$V_{IL(ac)}$	Maximum value at which ac input is a logic LOW	V_{REF}-0.22 V

Table 6-4 lists the input dc and ac levels for differential signals (clock and data strobe). These levels are only used to compute the slew rates of the differential signals. Timing for the differential signals is always referred to the differential cross point.

Table 6-4 Input logic levels for differential signals, LPDDR2-466 to LPDDR2-1066

Symbol	Value
$V_{ID(ac)}$	0.44 V
$V_{ID(dc)}$	0.26 V

6.2.2 PCDDR2

Table 6-5 lists the supply voltages and

Table 6-6 lists the dc logic levels for PCDDR2.

Table 6-5 System voltages

Symbol	Interpretation	Min	Typical	Max
VDDQ	I/O voltage at DRAM	1.7 V	1.8 V	1.9 V
V_{DDPX}	I/O voltage at memory controller	1.7 V	1.8 V	1.9 V
V_{REF}	Input reference voltage	0.5*V_{DD}		
V_{TT}	Termination voltage	V_{REF}		

Table 6-6 Input dc logic levels for single-ended signals

Symbol	Interpretation	Value
$V_{IH(dc)}$	Minimum value at which dc input is a logic HIGH	V_{REF}+0.125 V
$V_{IL(dc)}$	Maximum value at which dc input is a logic LOW	V_{REF}-0.125 V

Table 6-7 lists the ac logic levels for single-ended signals. Note that the levels as specified by JEDEC depend on data rate. Higher-data-rate DRAMs are required to have more sensitive receivers.

Table 6-7 Input ac logic levels for single-ended signals

Symbol	Interpretation	PCDDR2-400 DRAM, or PCDDR2-533 DRAM Value	PCDDR2-667 DRAM Value
$V_{IH(ac)}$	Minimum value at which ac input is a logic HIGH	V_{REF}+0.25 V	V_{REF}+0.20 V
$V_{IL(ac)}$	Maximum value at which ac input is a logic LOW	V_{REF}-0.25 V	V_{REF}-0.20 V

Table 6-8 lists the swing requirement for differential signals. In contrast to LPDDR2, in PCDDR2 one level is used for both sides of the eye for differential signals.

Table 6-8 Input signal swing for differential signals

Symbol	Value
$V_{ID(ac)}$	0.5 V

6.2.3 PCDDR3

Table 6-9 lists the supply voltages and Table 6-10 lists the dc logic levels for PCDDR2.

Table 6-9 System voltages

Symbol	Interpretation	Min	Typical	Max
VDDQ	I/O voltage at DRAM	1.425 V	1.5 V	1.575 V
V_{DDPX}	I/O voltage at memory controller	1.425 V	1.5 V	1.575 V
V_{REF}	Input reference voltage	$0.5*V_{DD}$		
V_{TT}	Termination voltage	V_{REF}		

Table 6-10 Input dc logic levels for single-ended signals

Symbol	Interpretation	Value
$V_{IH(dc)}$	Minimum value at which dc input is a logic HIGH	V_{REF}+0.1 V
$V_{IL(dc)}$	Maximum value at which dc input is a logic LOW	V_{REF}-0.1 V

Table 6-11 lists the ac logic levels for single-ended signals. Note that the levels as specified by JEDEC depend on data rate. Higher-data-rate DRAMs are required to have more sensitive receivers.

Table 6-11 Input ac logic levels for single-ended signals

Symbol	Interpretation	DDR3-800 or DDR3-1067 DRAM using AC175
$V_{IH(ac)}$	Minimum value at which ac input is a logic HIGH	V_{REF}+0.175 V
$V_{IL(ac)}$	Maximum value at which ac input is a logic LOW	V_{REF}-0.175 V

Table 6-12 lists the input dc and ac levels for differential signals (clock and data strobe). These levels are only used to compute the slew rates of the differential signals. Timing for the differential signals is always referred to the differential cross point.

Table 6-12 Input logic levels for differential signals, DDR3-800 to DDR3-1066, using AC175

Symbol	Value
$V_{ID(ac)}$	0.35 V
$V_{ID(dc)}$	0.20 V

6.3 IO settings

6.3.1 LPDDR4

If data and command/address are simulated together, like in a “WriteCA” simulation, the VOH value for the DQ/DMI drivers may be set differently than the VOH value for the CA/CS drivers. Each net type will get its own Vcent.

The Vref values for DQ and CA are stored in different mode registers in the DRAM.

6.3.2 LPDDR5

Write clock (WCK) does not start and stop when ranks are switched, so there is no “target-rank” or “non-target rank” concept for WCK. In a two-rank system, each rank should use the same ODT value for WCK.

6.4 On-die interconnect

Some SoCs will have a significant amount of on-die interconnect between the die bump and the receiver input. This interconnect will filter the signal, possibly by a lot if it includes a series resistor for ESD protection.

Figure 6-1 shows how to include a model of the on-die interconnect (the padsig route) and how to name nodes for compatibility with the MSAK templates.

The on-die interconnect subcircuit should include the effective load of the receiver, probably just an input capacitance.

In read mode, it's expected that the ODT will be modeled in the TX subcircuit.

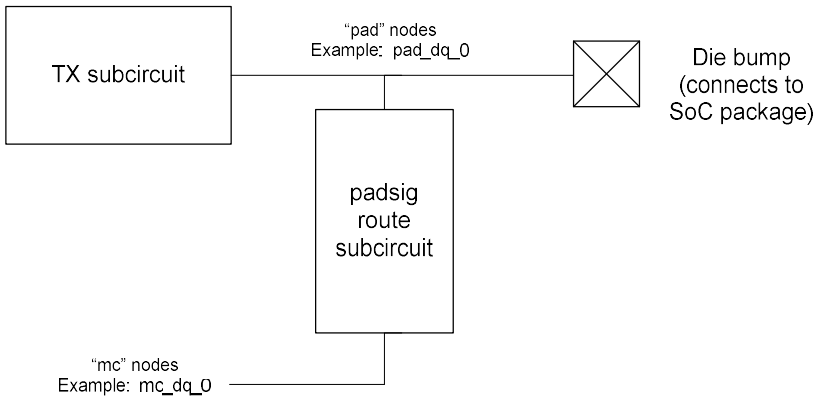

Figure 6-1 Schematic for modeling SoC IOs with on-die interconnect between die bump and receiver

6.5 Timing reference loads

6.5.1 LPDDR4

The reference load is 50 Ω to ground.

6.6 Tool-specific considerations

6.6.1 Synopsys HSPICE

The .MODULE/.ENDMODULE statements may be used to wrap IO models that use foundry-specific transistor models, so that models from multiple foundries may be used without conflicting with each other.

It is recommended to wrap the Qualcomm IO models in .MODULE / .ENDMODULE instead of the DRAM IO models. The DRAM IO models will be encrypted and if there is a problem with the encapsulation it might not be possible to debug.

6.6.2 Synopsys CustomSim (XA)

When writing netlists intended to be submitted to XA, there are a few syntax restrictions beyond what is required with HSPICE.

First, a space should be present between a use of ".IF" and its condition. For example, this is not acceptable:

```
.IF(pvt==0)
```

But this is acceptable:

```
.IF (pvt==0)
```

Second, any parameter that is to be probed in the transient output file should be renamed when it is probed. That is, this will cause an error:

```
.probe driver=PAR('driver')
```

But this is acceptable:

```
.probe driver=PAR('driver_int')
```

The .nodeset statement may be needed to set the initial conditions of some nodes. It should not be used more than is needed. It is probably only needed to set the initial conditions of outputs of differential drivers.

6.6.3 Synopsys Finesim

The following statement, when run in Finesim, leads to mc_dqs_0_diff appearing as a parameter in the result file instead of as a voltage waveform:

```
.probe mc_rdqs_0_diff = V(mc_rdqs_0_t, mc_rdqs_0_c)
```

The probe should be done with these statements instead, for compatibility with Finesim:

```
Emc_rdqs_0_diff    mc_rdqs_0_diff    0    VCVS    mc_rdqs_0_t    mc_rdqs_0_c    1
Rmc_rdqs_0_diff    mc_rdqs_0_diff    0    1e6
.probe V(mc_rdqs_0_diff)
```

6.7 Return current

When signaling models are extracted, if power nets carry return current, the power nets should be included in the extraction and ports should be placed on those power nets.

Figure 6-2 shows a circuit schematic. An ideal dc voltage source should be used at the boundary of the interconnect model to close the loop for any return current that does flow in the power net. The voltage is arbitrary since the important thing is to have zero impedance. It's recommended to set the voltage at the nominal value of the power rail (vddval).

The dc source is placed at the boundary between the interconnect and the IO. It is preferred (in the interest of keeping the interconnect model simple) to place the dc source at the top level of the circuit simulation, outside the interconnect model.

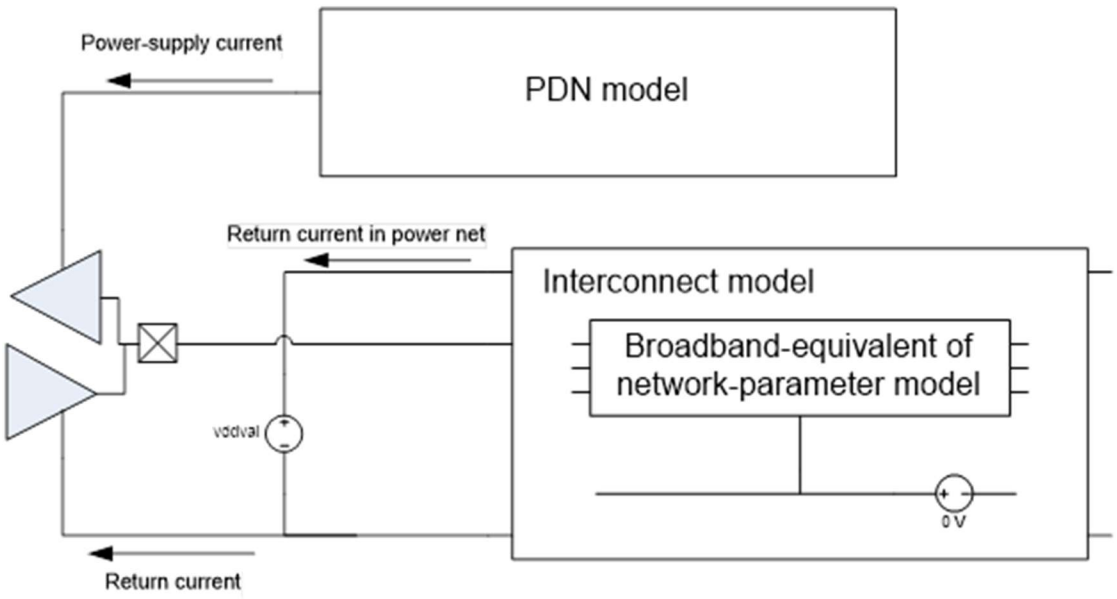

Figure 6-2 Circuit schematic showing stitching of power-supply return current

6.8 Parameters in netlists generated by MakeMSAK

6.8.1 LPDDR4X

VOH_RATIO is the ratio of VOH to VDDQ.

"write_int" and "write" are Boolean parameters used by the simulator decks to record whether a certain case is in write mode or read mode. If the value of write_int is 1, it's in write mode and if the value is 0, the case is in read mode. "write_int" is really just an alias for "write" that we introduced to work around a syntax limitation in XA.

7 Metrics

This section defines several metrics that can be calculated from the results of circuit simulation. These metrics indicate how well the design performs.

7.1 Aperture Calculation for Single-Ended Signals

Signal apertures (eye openings) are not specified by JEDEC for standards up through LPDDR3/DDR3, but calculating apertures can still give a lot of insight.

To compute the aperture of a signal, an eye diagram is formed. A trapezoidal mask is placed in the eye diagram, using the ac logic levels on the left and the dc logic levels on the right. The maximum width of the mask is reported as the aperture width. Figure 7-1 shows an example, where the aperture width is 1.45ns.

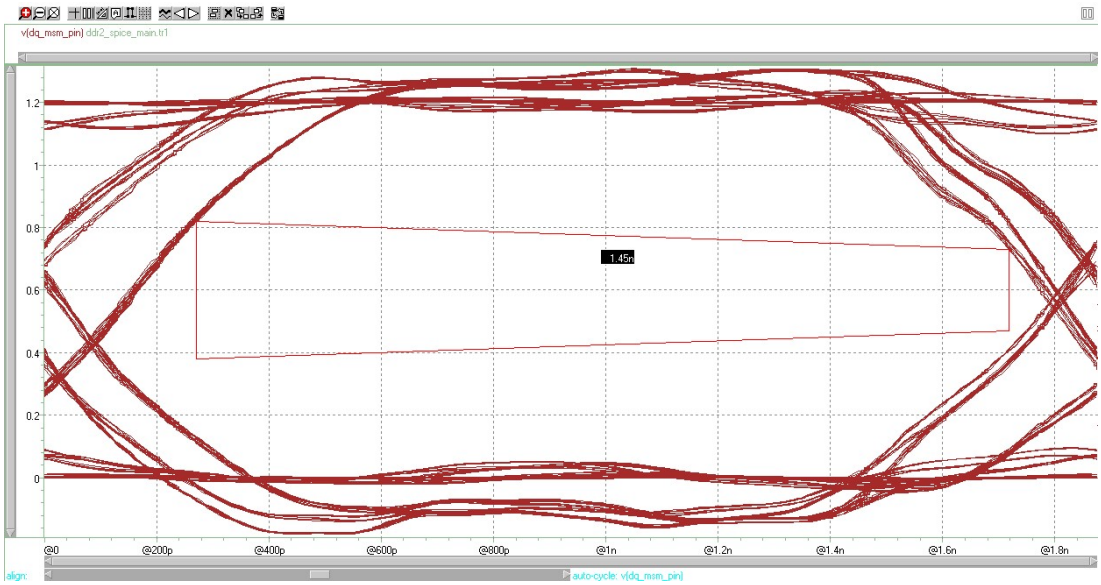

Figure 7-1 Example of aperture calculation

7.1.1 Folded apertures

If the eye is formed by breaking the waveform into sections that are each exactly one UI long, then overlaying them, the resulting eye diagram is called “folded.”

In Custom WaveView, folded apertures result if the eye diagram is configured to have a trigger period “same as UI.” Figure 7-2 shows an example.

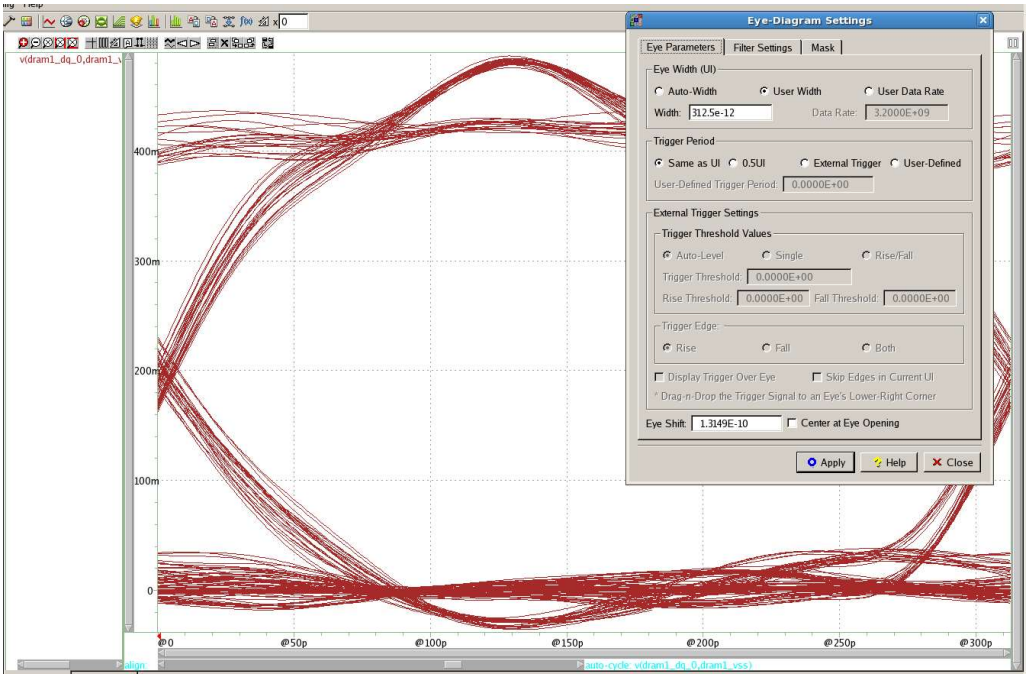

Figure 7-2 Example of folded eye

7.1.2 Triggered apertures

A “triggered” eye is formed by taking sections of the waveform that are exactly one UI long, each one centered on an edge of the differential strobe or clock signal, then overlaying them. This is the same behavior that Custom WaveView has if the “External Trigger” option is chosen in the eye-diagram settings. Figure 7-3 shows an example.

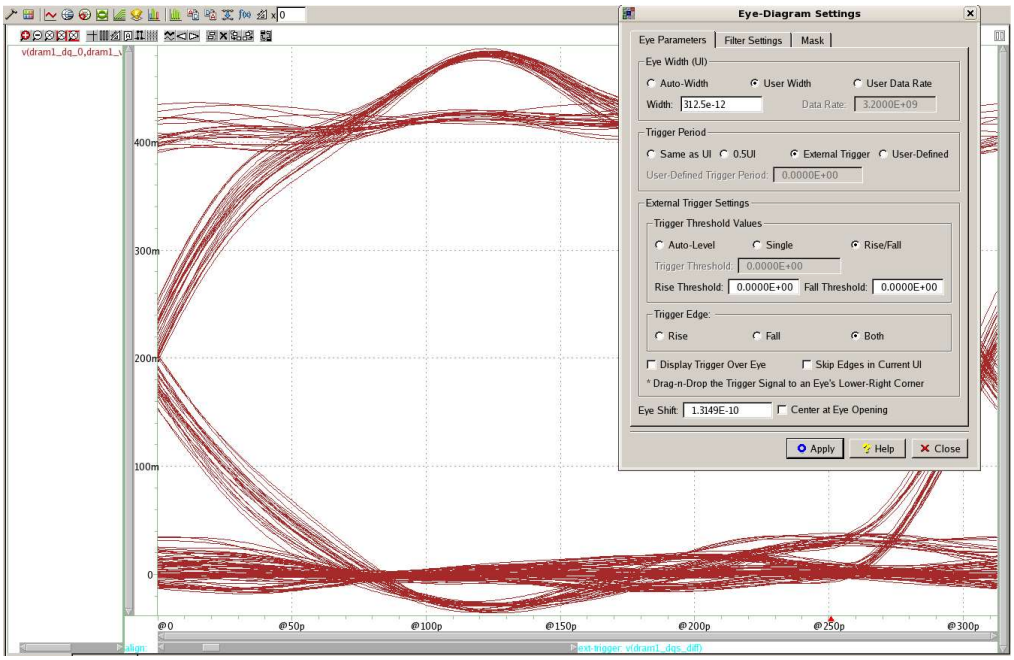

Figure 7-3 Example of triggered eye

7.1.3 Aperture height

An aperture height can also be calculated by computing the minimum height between the lower and upper eye contours in the middle portion of the eye. The "middle portion of the eye" would be the time range when the signal needs to be valid, such as the tDIVW time in LPDDR4.

7.1.4 Clock Enable (CKE)

Apertures are computed based on the eye width that is relevant for the receiver sampling, not necessarily the time between signal toggles. This distinction is important for CKE in LPDDR3. The LPDDR3 standard restricts the toggle rate of CKE: it can switch at most once every three clock cycles. But it is sampled on every rising clock edge, so the setup and hold constraints need to be met at every clock edge. Therefore, for aperture calculation, it should be treated as if it could toggle once every clock period.

7.1.5 Hexagonal masks in LPDDR5

Figure 7-4 shows an eye diagram of one data net with a hexagonal receiver mask superimposed. The LPDDR5 specification calls out a hexagonal receiver mask with parameters for the width at Vref (TdIVW1 in this case), the width at the receiver thresholds (TdIVW2 in this case), and the height (VdIVW).

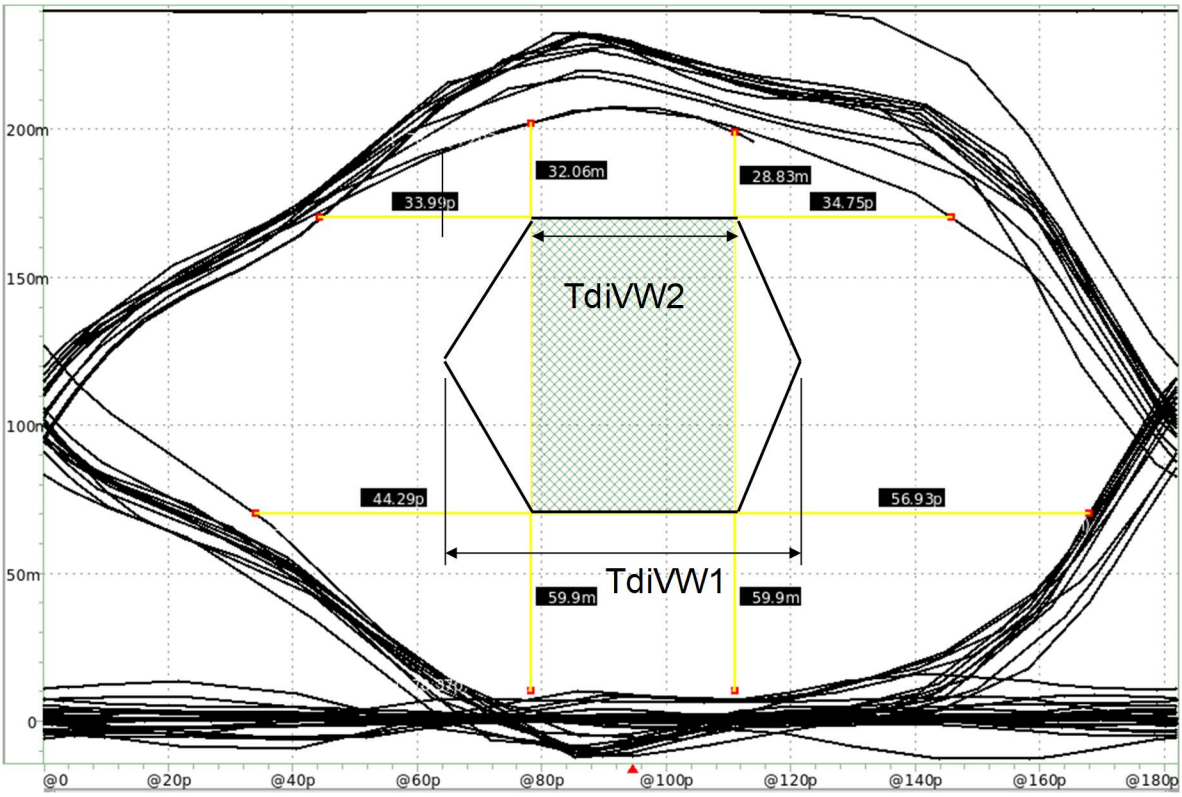

Figure 7-4 LPDDR5 data eye with hexagonal receiver mask

We report metrics that express margins against each of those parameters

Table 7-1 Definitions of eye metrics for hexagonal masks

Metric	Related to JEDEC parameter	Notes
Eye width at Vref	TdIVW1, TcIVW1, tCSIVW1	This metric gives the width across the whole eye, including the receiver mask.
Aperture at VxIVW	TdIVW2, TcIVW2, tCSIVW2	This metric gives the minimum of the eye widths at the upper and lower receiver thresholds.
Noise margin	VdIVW, VcIVW, vCSIVW	This metric tells how much additional voltage noise could be added to either the high side or the low side of the eye before violating the receiver threshold.

7.1.6 LPDDR5X single-pulse specs

MERiT, up to 6.1, does not calculate metrics that check against the vDIHP1, vDIHP2, vDILP1, or vDILP2 specifications in LPDDR5X.

7.2 Aperture Calculation for Differential Signals

Apertures and jitter are also defined for the differential clock and strobe signals. Table 7-2 lists the apertures reported and, for each aperture, which levels are used to define the left side and the right side of the eye. These apertures do not directly apply to any JEDEC specifications.

Table 7-2 Levels used for apertures of differential signals

	Left side of eye	Right side of eye
LPDDR2		
Clock or DQS aperture	$V_{ID(ac)}$	$V_{ID(dc)}$
PCDDR2		
Clock aperture	$V_{ID(ac)}$	$V_{ID(ac)}$
DQS aperture	$V_{ID(ac)}$	$V_{ID(ac)}$

For LPDDR2, JEDEC defines the tDVAC parameter, which is the time that the differential signal spends above (or below) the ac level.

For PCDDR2, there is only one threshold level for differential signals ($V_{ID(ac)}$) so the apertures are reported with a rectangular window based on that threshold.

7.3 Eye Width

Eye width metrics are much like apertures but may be computed at the Vref level in addition to the signaling thresholds.

7.4 Time Valid Above AC Level

The parameters tVAC and tDVAC are the times that single-ended or differential signals (respectively) spend above or below the AC level for a valid transition. These parameters are defined for signals received by the DRAM as specified by JEDEC.

7.5 Differential Cross Point Error

The voltages at which the P and M sides of differential pairs cross define the VIX and VOX parameters. VIX and VOX are the difference between the actual cross point and the ideal cross point (V_{REF}). VIX refers to the differential cross point at an input, and is observed at the input pins of the appropriate device. VOX is the differential cross point of an output, and is observed at the timing reference circuits.

Figure 7-5 shows an example. Three cross points of this differential signal are shown. At the middle one, the cross point is at 0.561 V. The ideal cross point is 0.6 V, so VIX is -39 mV in this case.

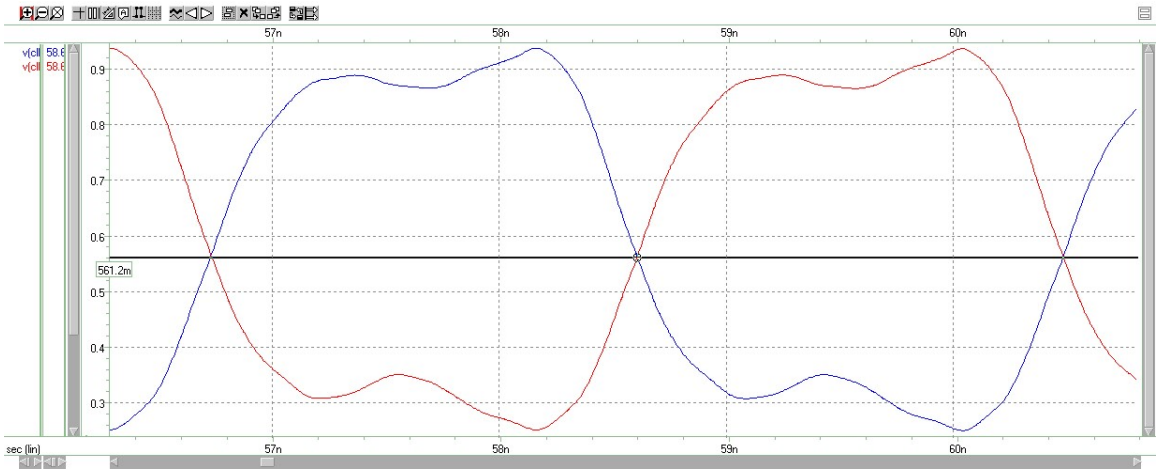

Figure 7-5 Example of VIX or VOX calculation

For LPDDR4, the specification on differential cross point error is expressed as a ratio, so VIX is reported as a percentage.

7.6 Skew Calculation

The skew reports are intended to support setup and hold time calculations. One line item in each setup or hold budget is the skew across the PCB and packages. The skew in the interconnect is just the difference in delay for two signals crossing the interconnect—the quarter-cycle offsets are not considered. To compute the difference in delays, we first compute the delays of each signal type.

7.6.1 Interpretation of arrival-time shifts

Figure 7-6 shows a strobe waveform with its transitions lined up with the transitions of the signal waveform. The setup-time constraints apply between corresponding edges and the hold-time constraints apply between each strobe edge and the following signal edge.

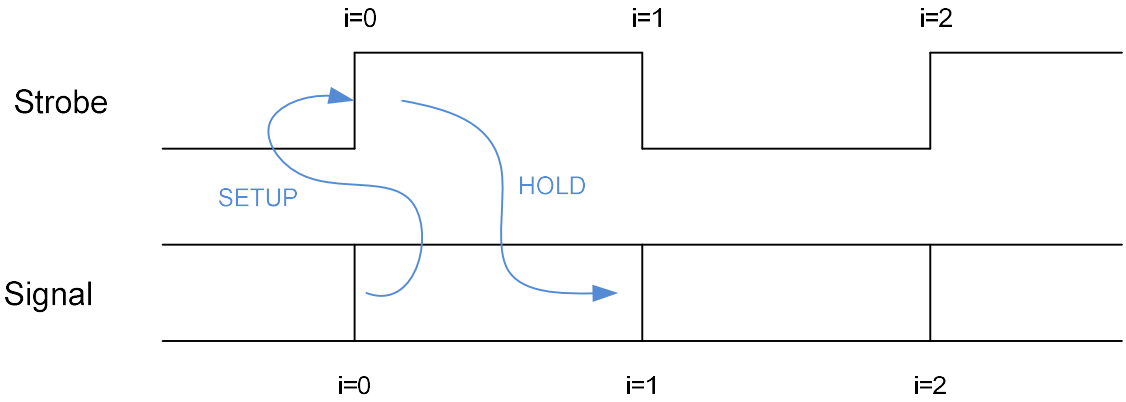

Figure 7-6 Correspondence of strobe and signal edges, after any needed shifting

Table 7-3 lists the standard-dependent time shift that should be applied to align strobe and signal edges. In that table, "UI" refers to the width of the signal pulse, not necessarily tCK/2.

Table 7-3 Arrival-time shifts

Standard	Net type / mode	Direction and amount to shift strobe when processing arrivals and computing skews
All	Data / read	None
All except LPDDR4	Data / write	Earlier (to the left) by UI/2
PCDDR3	CA	Earlier (to the left) by UI/2. And use only the rising edges of CK.
LPDDR2/3	CA	Earlier (to the left) by UI/2. Use both rising and falling edges of CK.
PCDDR3	Control (CS, CKE, ODT)	Earlier (to the left) by UI/2. Use only the rising edges of CK.
LPDDR2/3	Control (CS, CKE)	Earlier (to the left) by UI/2. Use only the rising edges of CK.
LPDDR4	Data / write	Later (to the right) by tDQS2DQ – UI/2
LPDDR4	CA or CS	Earlier (to the left) by UI/2. And use only the rising edges of CK.

7.6.2 Timing Reference Circuits and Buffer Delays

Each driver is associated with an identical driver, identically stimulated, that drives a timing reference circuit. In Figure 7-7, the upper network is the "real" driver driving the network under test with a real receiver at the load. The lower network shows an identical driver driving an RC circuit, which is the timing reference circuit. The function of the timing reference circuit is to define the time at which the driver launches a signal into the interconnect. By using the timing reference circuit, it's possible to separate the delay of the driver from the delay of the interconnect.

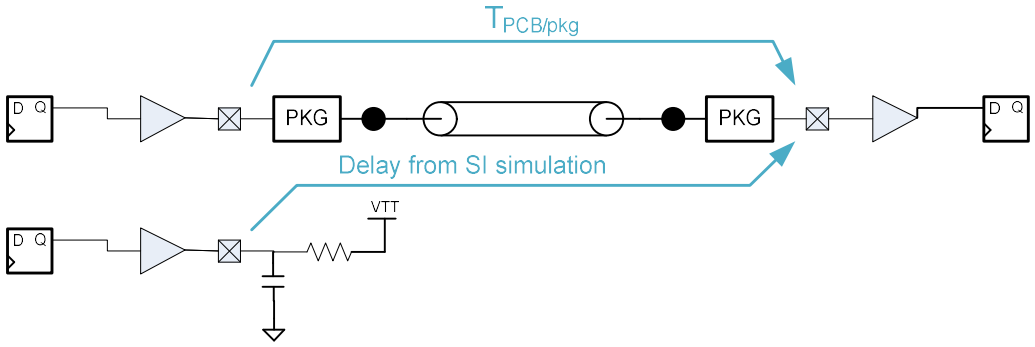

Figure 7-7 Link delay definitions

JEDEC specifies timing reference circuits to use to define the output times of DRAM drivers.

Design of timing reference circuits should obey these guidelines:

1. For source-synchronous timing analysis, the signals must use the same timing reference circuits as their associated strobes.
2. For common-clock timing analysis (or to compute the overall path delay), the timing reference circuit should be the same as the reference load used in characterizing the delay of the IO cell. If the parasitics of the redistribution layer (RDL) are included in the chip-level timing analysis, the timing reference circuit should include capacitance for the typical RDL route in addition to the IO cell load.
3. The timing reference circuit should be similar to the actual load the circuit will drive, especially in the voltage swing.

The third point (matching voltage swing between the timing reference circuit and the actual load) is important because the driver may be nonlinear. The delay of a nonlinear driver into a low-swing timing reference circuit will not track the delay into a full-swing load.

The timing reference circuit should not include a package model. The point of the timing reference circuit is to see the IO delay without seeing reflections, and a package model could introduce reflections.

7.6.2.1 Vref

For LPDDR4X it is necessary to use a different Vref for the timing reference circuits than for the actual receivers. This difference is necessary because the timing reference load in LPDDR4X is 50 Ω to ground, which is not a valid termination for any real LPDDR4X circuits. So for LPDDR4X we compute a special Vref value, only used for the timing-reference circuits, which is computed as half of the expected output-high level.

7.6.3 Skew Calculation

Skew is just the difference between delays. Figure 7-8 shows a strobe circuit and a data circuit, both with their timing reference circuits. The blue arrows show how the delays of the interconnects are defined.

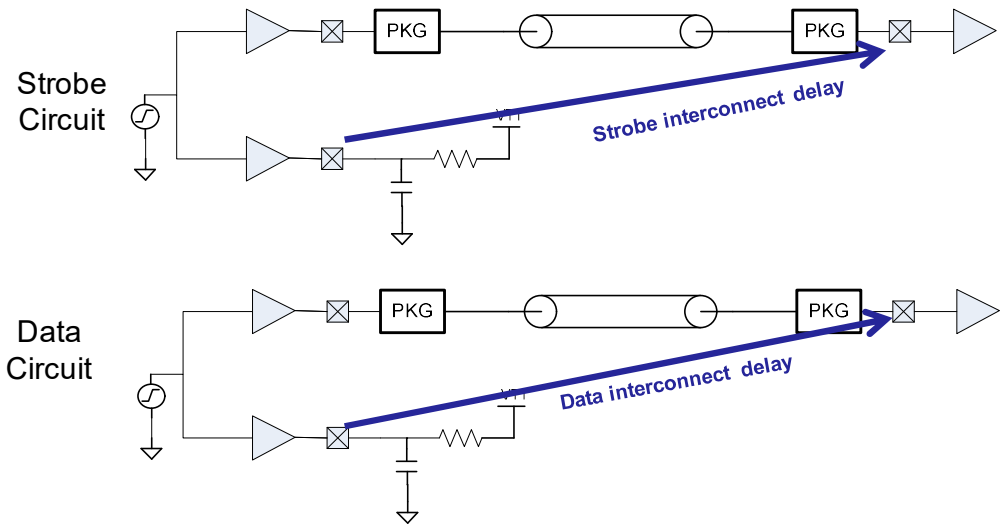

Figure 7-8 Interconnect delay definition

The skew is defined as

$$skew = strobe\ interconnect\ delay - data\ interconnect\ delay$$

The interconnect delays can be defined from the endpoints of the arrows in Figure 7-8, or they can be defined with the use of the trigger nodes, as in Figure 7-9.

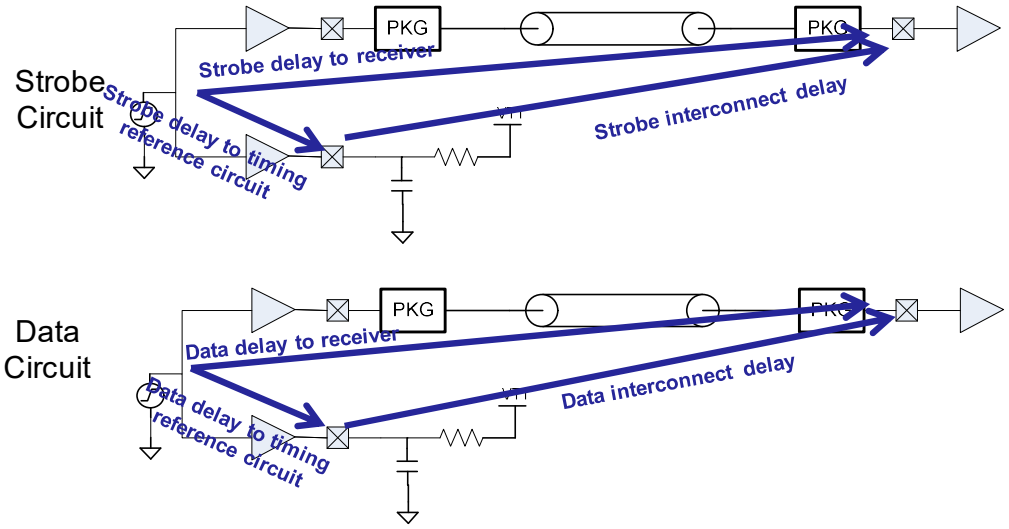

Figure 7-9 Interconnect delays using triggers

Then the skew formula is

$$skew = strobe\ receiver\ delay - strobe\ buffer\ delay - (data\ receiver\ delay - data\ buffer\ delay)$$

This second form may be more useful because it allows precomputing the buffer delays.

If the trigger nodes are used as in Figure 7-9, the delay is computed from when the trigger fires to when the voltage at the node of interest (either a receiver or a timing-reference node) crosses

Vref. In many standards, Vref is pegged to half of the IO voltage, but in some standards it is adjustable.

7.6.4 Time Correlation

A simulation of strobes and data for several bit times will produce vectors of delays: one delay for every edge simulated. How do we combine those results to find the minimum and maximum skews?

There are two approaches.

1. Not considering time correlation
   Define min skew as min(strobe interconnect delay[n]) – max(data interconnect delay[n])
   Define max skew as max(strobe interconnect delay[n]) – min(data interconnect delay[n])
2. Considering time correlation
   Define min skew as min(strobe interconnect delay[n] - data interconnect delay[n])
   Define max skew as max(strobe interconnect delay[n] - data interconnect delay[n])

If the simulation includes significant coupling between the strobe network and the data network(s), the "considering time correlation" approach should be used.

7.6.5 Setup and Hold Time Calculation

The owner of the system timing budget is responsible for calculating the setup and hold time margins based on the skews. The system timing budget is outside the scope of this document, but the analysis needs to know the skew definition used in the budget.

The input timing parameters for DRAMs are specified at the $V_{IH/L(ac/dc)}$ levels. So in write mode, if the skews at the ac/dc levels are used, they may be compared to the specified setup and hold time requirements, after the requirements have been derated. If the skews at the V_{REF} level are used, they need to be compared to setup and hold time requirements that are also relative to V_{REF}. One way of adjusting the DRAM setup and hold time requirements to refer to V_{REF} is described in a Qualcomm document[3].

In read mode, we should choose voltage levels that correspond to the voltage levels used to define timing in the I/O cell characterization and ASIC timing. For now, skews are reported relative to V_{REF}.

7.6.6 DQS-DQ Skew

Figure 7-10 shows an example of calculating a hold-time DQS-DQ skew.

The upper pane shows the differential DQS signal at the timing reference circuit (red) and at the receiver (blue). Since the DQS is plotted as differential, 0 V on this plot corresponds to the differential cross point. The delay of DQS is calculated from 0 V on the red trace to 0 V on the blue trace, or 74 ps.

The lower pane shows a DQ signal at its timing reference circuit (red) and its receiver (blue). The delay of the DQ signal is 63 ps, computed from when the red trace crosses V_{REF}=0.6 V to

[3] "DRAM setup and hold time adjustment," Scott Powers, 19 May 2009. Available: http://qshare2.qualcomm.com/qshare2/drl.html?objectId=090100258053c959&display=attachment&v=c

when the blue trace crosses V_{REF}=0.6 V. The delay of the DQS signal is 74 ps and the delay of the DQ signal is 63 ps, so the hold-time skew for this edge is 11 ps.

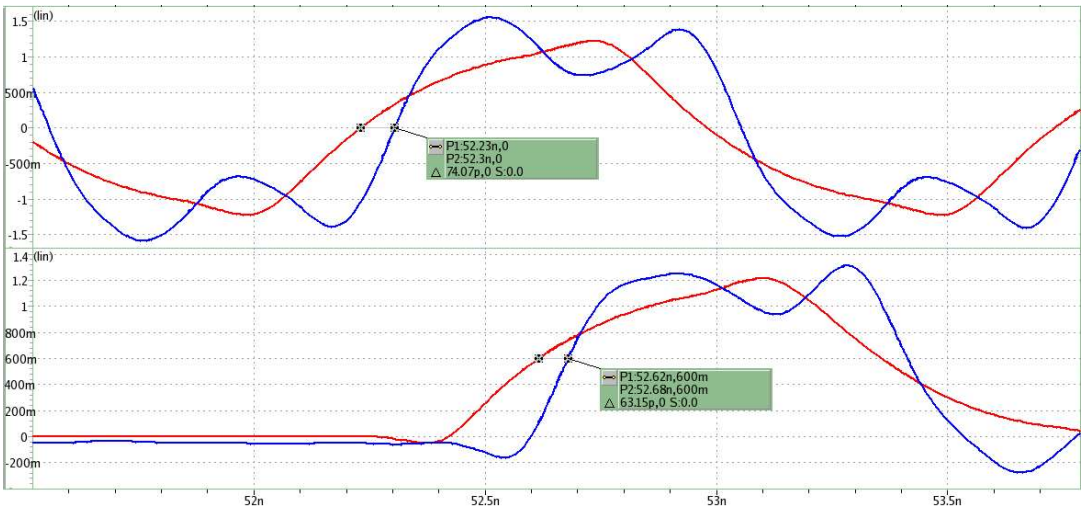

Figure 7-10 Example of skew calculation

7.6.7 Clock-DQS Skew

Skew may also be reported between clock and DQS. Both rising and falling edges of the clock are considered, since clock and DQS are at the same rate.

Computing clock-DQS skew may require some additional knowledge about the logical organization of the channel, since in the simulation, it will be necessary to run multiple bytes and connect each byte's DQS to the appropriate DRAM load or loads.

7.6.8 Clock-Command/Address Skew

Skew is also reported between clock and the command/address signals.

7.6.9 Static-Dynamic Skew Breakdown

The setup and hold skews found for each signal will not be perfectly balanced. That is, a given signal will have more hold skew or more setup skew. The amount of the imbalance is called static skew. If a static delay is added to the signal or its strobe or clock, this static skew can be removed. The portion that remains is called dynamic skew.

7.7 Slew Calculation

7.7.1 Setup/hold slew rates

Slew rates for setup/hold calculation are calculated at the receiver (input slew rates) using the reference levels in Table 7-4. Slew rates are reported in units of V/ns.

Table 7-4 Reference Levels for Slew Calculation of Single-ended Signals

Timing constraint type	Transition direction	First reference level	Second reference level
Setup	Rising	V_{REF}	$V_{IH(ac)}$
Setup	Falling	V_{REF}	$V_{IL(ac)}$
Hold	Rising	$V_{IL(dc)}$	V_{REF}
Hold	Falling	$V_{IH(dc)}$	V_{REF}

In LPDDR2, the CKE signal has different thresholds than the other signals. CKE slews are calculated to the CKE thresholds. Table 7-5 lists the voltage levels used to define slews.

Table 7-5 Reference Levels for Slew Calculation of Differential Signals

Standard	Transition direction	First reference level	Second reference level	Source
PCDDR2	Rising	-250 mV	500 mV	General Note 2 to Table 42 in JESD79-2E
PCDDR2	Falling	250 mV	-500 mV	
PCDDR3	Rising	-200 mV	200 mV	Note 1 to Table 25 in JESD79-3D
PCDDR3	Falling	200 mV	-200 mV	
LPDDR2	Rising	$-V_{ID(dc)}$	$V_{ID(dc)}$	Table 77 in JESD209-2
LPDDR2	Falling	$V_{ID(dc)}$	$-V_{ID(dc)}$	

Figure 7-11 shows an example. This single-ended signal travels from its reference voltage to its ac level, 0.22 V above the reference, in 150 ps. The slew rate is therefore 1.47 V/ns.

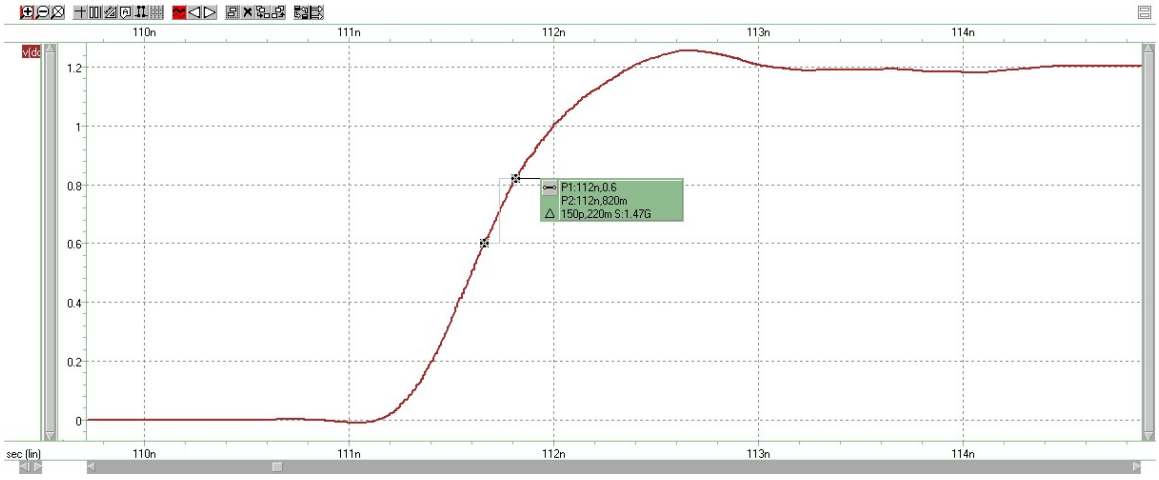

Figure 7-11 Example of slew calculation

In reporting the slew of the differential clock, the first several cycles should be skipped because the clock has not stabilized yet. The DRAM is not required to accept clock signals until after they have stabilized. In typical waveforms, ten cycles is enough for the clock to stabilize.

For signals that happen to be non-monotonic or have glitches near the threshold region, slews are computed as directed by the JEDEC specifications.

7.7.2 LPDDR4-style slew rate

For LPDDR4, slew rates are calculated using the upper and lower voltages of the input-valid window as the endpoints (VcIVW or VdIVW). There are no tangent-line calculations.

7.8 Pulse Width Calculation

To support checking of JEDEC's minimum-pulse-width constraint, a calculation of minimum pulse width is provided. The pulse-width calculation includes the pulses in preambles and postambles, if any are present.

Pulse widths are defined at Vref. Figure 7-12 shows an example waveform with four pulses noted. In this section of this waveform, the minimum pulse width was 271 ps.

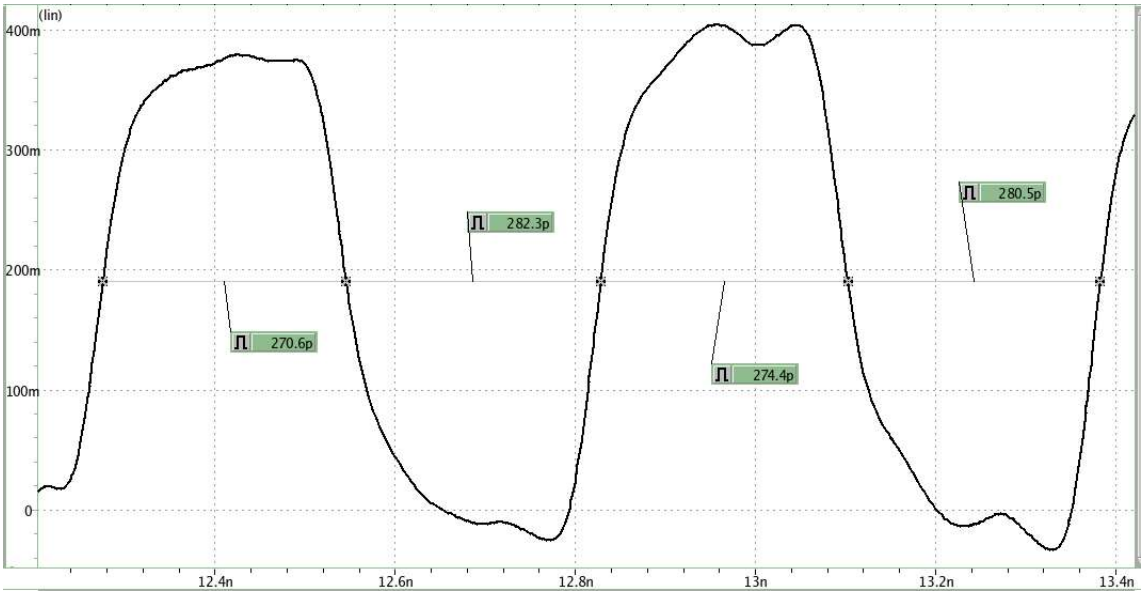

Figure 7-12 Example of pulse-width calculation

Peak-to-peak pulse-width distortion can be estimated as two times the difference between the ideal pulse width and the minimum pulse width.

7.9 Overshoot/Undershoot Calculation

Overshoot amplitudes and areas are reported consistent with the JEDEC specifications. Note that while JEDEC uses the term "undershoot" for occasions when signal voltages go below ground, this process uses the term "overshoot" for both directions.

Overshoots are reported separately for the DRAMs and for the memory controller, since the memory controllers have different overshoot requirements than the DRAMs do.

Overshoot area is computed with respect to the nominal VDD value set for the simulation. This might not be consistent with all of the JEDEC specs: the LPDDR4 spec calls for overshoot area to be computed referring to the maximum operating voltages. PSIG's method would be conservative in that case.

7.10 Skew Budgeting

The skew reported from the circuit simulations takes into account skew due to intersymbol interference, crosstalk in the board routing, and nominal length mismatches. An example of a nominal length mismatch is in the Clock-CA skew; the clock and CA routes cannot be the same length, and in fact have different numbers of loads.

There are other sources of skew that cannot be simulated before a layout is available, so they may be included in skew budget tables. These sources of skew may include the following:

- Crosstalk in memory controller package
- Length mismatch above and beyond nominal
- Return path discontinuities

7.10.1 Example skew budget

Table 7-6 shows an example of how skews may be budgeted in a prelayout or design-space-exploration investigation. The "Total" row is the linear sum of all the previous rows. The "Target" row lists the skew targets for each skew type. The cases where the total expected skew exceeds the target skew are shaded red.

Table 7-6 Skew budget for system timing analysis

Skew source	Expected skew [ps]									
	DQS-DQ Read		DQS-DQ Write		Clock-CA		Clock-CS/CKE		Clock-DQS	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
ISI, nominal length mismatch, and crosstalk in board routing	-73	78	-91	88	-451	-180	-328	-84	-190	-56
Extra setup/hold time due to slow slew rate			-8.0	0	0	0	0	0		
Crosstalk in SoC package	-5	5	-5	5	-5	5	-5	5	0	0
Crosstalk in DRAM package	0	0	0	0	0	0	0	0	0	0
Length mismatch in SoC package	-7.4	7.4	-7.4	7.4	-7.4	7.4	-7.4	7.4	-7.4	7.4
Length mismatch on main board	-6.4	6.4	-6.4	6.4	-6.4	6.4	-6.4	6.4	-6.4	6.4
Length mismatch on DIMM	-6.4	6.4	-6.4	6.4	-6.4	6.4	-6.4	6.4	-6.4	6.4
Return path discontinuities	0	0	0	0	-15	15	-15	15	0	0
Total:	**-98**	**103**	**-124**	**113**	**-491**	**-140**	**-368**	**-44**	**-210**	**-36**

Target:	-100	100	-100	100	-550	150	-550	150	-200	200

7.11 Period jitter

Period jitter is reported for the strobe signals: Clock and DQS. It is defined as the deviation of any single period from the ideal period. A period is defined as the time between consecutive rising edges.

Figure 7-13 shows an example of period calculation. One period of this waveform starts at 259 ns and ends at 261.5 ns, having a period of 2502.5 ps. The next period of this waveform starts at 261.5 ns and ends at 264 ns, and has a period of 2497.3 ps. The ideal period is 2500 ps.

The deviation of the first period is 2502.5 ps – 2500 ps = 2.5 ps. The deviation of the second period is 2500 ps – 2497.3 ps = 2.7 ps.

The period jitter in this section of the waveform is the larger value, 2.7 ps.

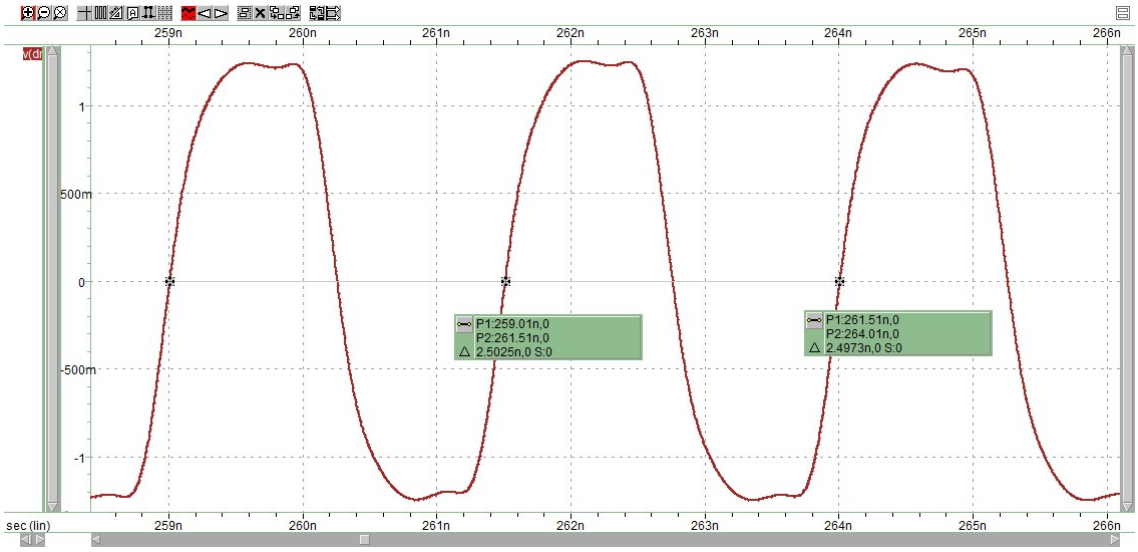

Figure 7-13 Example of period calculation

Peak-to-peak period jitter (also called peak-to-peak full-period jitter) is the difference between the longest and shortest periods in the simulation record.

Figure 7-14 shows another example. In this example the first period is 1000 ps long and the second period is 1000 ps long. The ideal clock period is also 1000 ps, so the period jitter is zero.

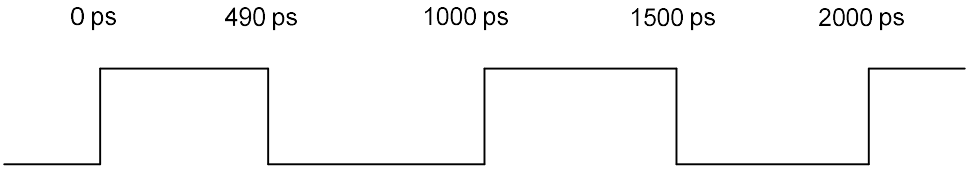

Figure 7-14 Example clock waveform

If we were to compute the peak-to-peak jitter of the falling edge, we would see that the first falling edge is 10 ps early and the second falling edge is at its ideal position. So the peak-to-peak jitter of the falling edges is 10 ps.

In transient simulation the first few edges of the strobe or clock have a lot of jitter. Calculation of period jitter for DDR should exclude the first several edges of the clock and, depending on the mode and the standard, may also exclude some of the first edges of the DQS burst.

7.12 Clock period error across multiple cycles

This parameter is computed with this formula:

$$tERR(nper) = \left(\sum_{j=i}^{i+n-1} tCK_j \right) - n * tCK(avg)$$

The error is calculated for all the clock periods in the simulation, excluding the startup periods at the beginning (which can be identified to MERiT with [clock_wait_cycles]). The range of periods checked sets the range for i.

Figure 7-15 shows a clock waveform from an example simulation. The average period is 625 ps. The markers are placed to show the actual lengths of the first few periods.

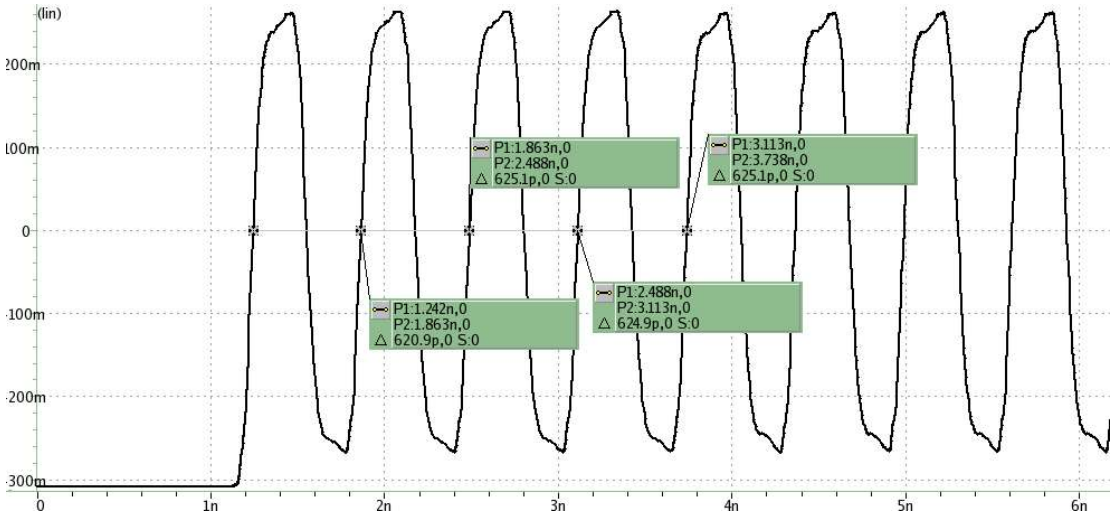

Figure 7-15 Example clock waveform with periods labeled

Table 7-7 lists numeric results from that simulation. The length of each period is extracted. In this example the first five periods are ignored, so i=1 is actually at period 6.

Table 7-7 Calculations for clock period error

Period index	tCK [s]	i	tERR(1per) [ps]	tERR(2per) [ps]	tERR(3per) [ps]
1	6.2086E-10				
2	6.2513E-10				
3	6.2492E-10				
4	6.2509E-10				
5	6.2433E-10				
6	6.2483E-10	1	-0.17		
7	6.2502E-10	2	0.02	-0.15	
8	6.2478E-10	3	-0.22	-0.19	-0.37
9	6.2794E-10	4	2.94	2.72	2.74
10	6.2202E-10	5	-2.98	-0.04	-0.26
11	6.2549E-10	6	0.49	-2.49	0.45
12	6.2483E-10	7	-0.17	0.32	-2.66
13	6.2503E-10	8	0.03	-0.14	0.35
14	6.2494E-10	9	-0.06	-0.03	-0.20
15	6.2501E-10	10	0.01	-0.05	-0.02
16	6.2494E-10	11	-0.06	-0.05	-0.11
17	6.2505E-10	12	0.05	-0.02	-0.01
18	6.2507E-10	13	0.07	0.11	0.05
19	6.2514E-10	14	0.14	0.21	0.25
20	6.2469E-10	15	-0.31	-0.17	-0.10
21	6.2525E-10	16	0.25	-0.06	0.09
22	6.2467E-10	17	-0.33	-0.07	-0.38
23	6.2493E-10	18	-0.07	-0.40	-0.14
24	6.2711E-10	19	2.11	2.03	1.71
25	6.2330E-10	20	-1.70	0.41	0.33
26	6.2574E-10	21	0.74	-0.96	1.15
27	6.2454E-10	22	-0.46	0.28	-1.42

28	6.2453E-10	23	-0.47	-0.93	-0.19
29	6.2534E-10	24	0.34	-0.13	-0.59
30	6.2483E-10	25	-0.17	0.17	-0.30
31	6.2517E-10	26	0.17	0.00	0.34
32	6.2869E-10	27	3.69	3.86	3.69
33	6.2167E-10	28	-3.33	0.36	0.53
		min	-3.33	-2.49	-2.66
		max	3.69	3.86	3.69

The period error is calculated for each cycle where it can be applied, and the minima and maxima are shown in the bottom two rows. For this example the results might be as listed in Table 7-8.

Table 7-8 Example results for clock period error

Parameter	Value
tERR(1per), min [ps]	-3.33
tERR(1per), max [ps]	3.69
tERR(2per), min [ps]	-2.49
tERR(2per), max [ps]	3.86
tERR(3per), min [ps]	-2.66
tERR(3per), max [ps]	3.69

7.13 N-UI jitter

N-UI jitter is like clock period error across multiple cycles, but it is generalized to work with the lengths of pulses rather than full periods, and it can be applied to any clock or strobe.

The N-UI jitter number basically gives the difference between the actual time taken for N pulses of the clock or strobe and the ideal time, N times the length of the ideal pulse.

7.14 DQS noise margin

For checking DQS noise margin, the “read burst followed by tristate” pattern is simulated, and the differential DQS waveform at the SoC receiver is plotted. The maximum value of v(DQS_diff) during the time that the DRAM driver is disabled is reported. In Figure 7-16, that value is about 30 mV.

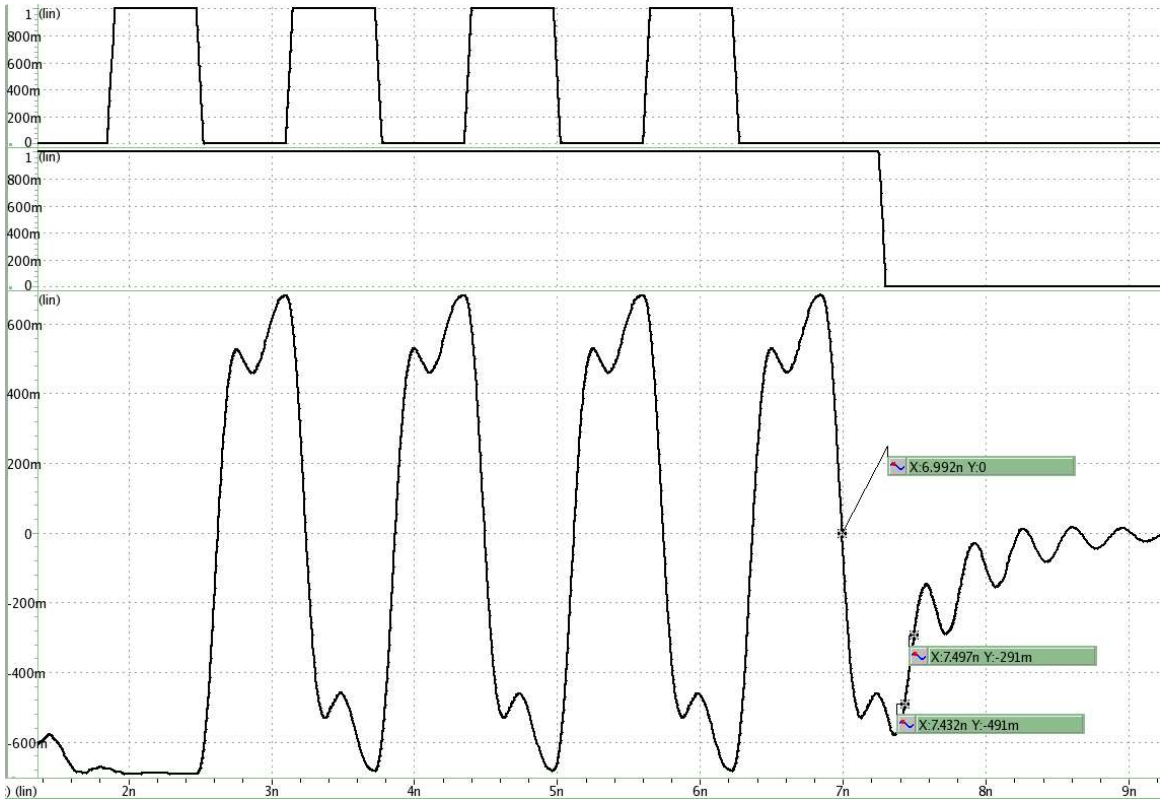

Figure 7-16 Waveforms from DQS noise margin check

Table 7-9 lists some constants involved in choosing the timing of when to disable the DRAM drivers. An optimization simulation is run, in which the DRAM component drives into a test load that includes a termination to mid-rail. The effective tRPST (read postamble time) is defined as the time that differential DQS stays below the threshold (-200 mV here) after the last DQS transition in the burst. The optimization simulation finds the shortest DQS postamble time that satisfies the goal postamble times in Table 7-9.

Table 7-9

Standard	Goal tRPST(min)	DQS thresholds to use in characterization [mV]
LPDDR2	0.33*tCK	-200
LPDDR3	0.30*tCK	-200

The goal postamble times are chosen based on conservative numbers from the JEDEC specs. In the case of LPDDR2, the tRPST(min) is calculated by subtracting the worst period jitter (per speed bin) from 0.43*tCK. Considering all the speed bins, 0.33*tCK was the shortest value. In the case of LPDDR3, there are two conflicting numbers for tRPST(min) in the spec; we chose the shorter one.

Figure 7-17 shows a simulated waveform for differential DQS at the test load. tRPST is computed as 999 ps in this case.

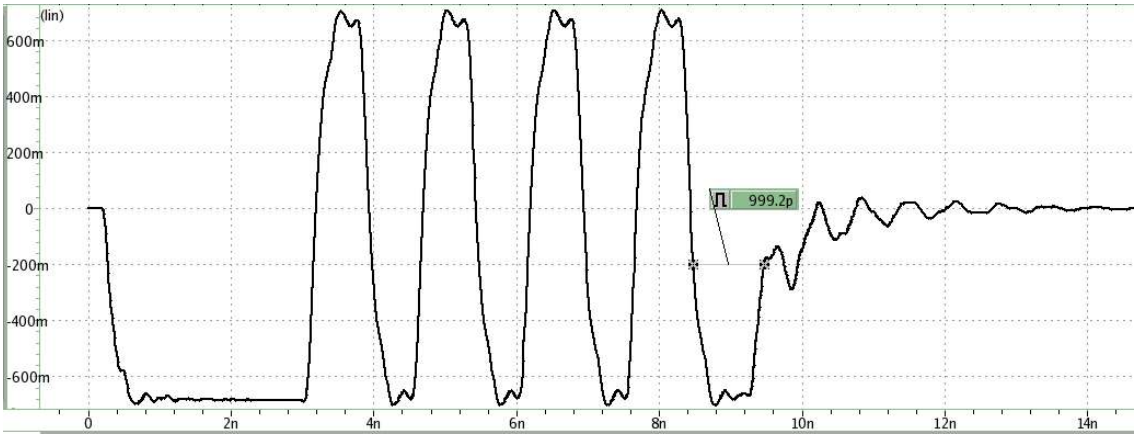

Figure 7-17 Characterization simulation for tRPST

This simple definition of tRPST is used instead of the slope-based method in the JEDEC spec because JEDEC's method is impractical: the slopes are not consistent enough.

7.15 Vcent_DQ and Vcent_CA

The Vcent voltage level for each net is calculated based on that net's eye diagram. The Vcent for the strobe group is computed as the average of the min and max Vcent values across the strobe group. The min and max Vcent values are of interest to check that the signal does not go out of the receiver's input range.

8 Targets

8.1 LPDDR4X

Table 8-1 lists targets that are used for generic LPDDR4X analyses at Qualcomm and how they may differ from JEDEC-spec guidance.

Table 8-1 Generic targets for LPDDR4X analyses

Item	Net type	JEDEC guidance	Relaxed for Qualcomm analyses
Max differential cross point error ratio [%]	DQS	10	20
	CK	12	20

The targets for differential cross-point error ratio have been relaxed because it has been found that systems don't always meet the strict JEDEC guidance.

9 Limitations

MSAK for LPDDR5 does not cover the "Link ECC" use case, in which RDQS is driven from SoC to DRAM.

10 Files

This document is located here (link), which can be accessed from go/msak.