

Technical Note

Design Guide – Dealing with DDR2/DDR3 Clock Jitter

Introduction

Prior to DDR2 technology, the expectation was that clock jitter specifications could be absorbed by the DRAM timing specifications. DDR2's faster clock rates and on-chip delay locked loop (DLL) changed all that, and industry-standard clock jitter specifications became a requirement for users and suppliers. Micron, however, actually started specifying clock jitter specifications with the release of DDR memory. Despite the fact that there seemed to be enough timing margin with DDR, the inclusion of the DLL begged for clock jitter guidance. Now, even though both DDR2 and DDR3 have clock jitter specifications, few DRAM users understand how to apply them or how to determine if their system clock violates the specification limits and what action to take if it does.

This design guide explores DDR2/DDR3 clock jitter specifications and provides guidance on how to apply them and how to deal with violations since many systems will unintentionally encounter them. (Note that statements made are equally applicable to DDR2 and DDR3 SDRAM, unless stated otherwise.)

Defining Clock Jitter

Clock jitter in DDR2/DDR3 systems is bound by clock jitter specifications in both positive and negative directions; the negative direction corresponds to a smaller clock period and the positive direction corresponds to a larger clock period. The specifications also limit the allowable clock jitter to Gaussian (or normal) distribution. If the clock jitter is not Gaussian in nature, then the clock violates the timing specifications.

What does Gaussian “in nature” mean? “In nature” is meant to convey that it is unlikely that the clock's profile will reveal a perfect Gaussian distribution; however, it should be close. How close? One way to answer that is to check for deterministic clock jitter. If deterministic clock jitter is observed, then it is likely not Gaussian enough.

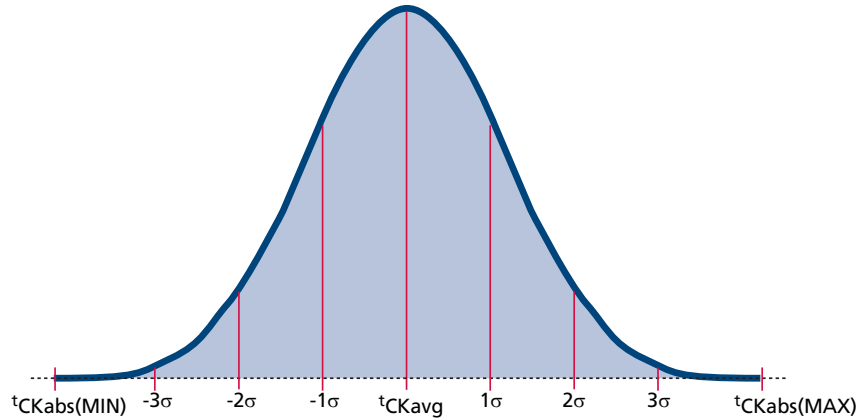
The two primary modes of operation are when the DLL is locking and when the DLL is locked. Each is associated with one set of concerns and criteria during initialization and another set of concerns and criteria during operation.

Random Clock Jitter

It is well understood that any clocked system will have some amount of random variability with its clock period. Even though good design practices can keep this skew to a minimum, there will always be some clock jitter. The benefit of restricting clock jitter to a Gaussian distribution is that the minimum and maximum clock pulse widths will only occur once in a while and not over back-to-back clock cycles (see Figure 1 below.)

Because clock jitter is random and can fit a Gaussian profile, $t_{ERRnper}$ specifications can be supported to minimize the effect on functionality. This is important because several timing parameters and commands require multiple clocks to occur.

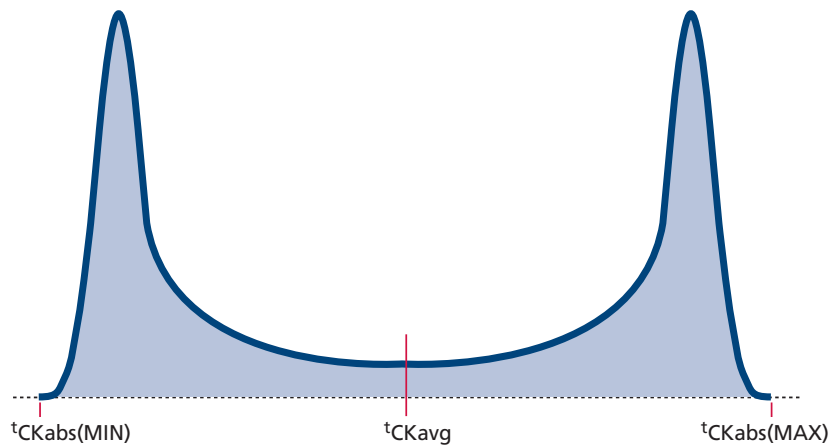
Figure 1: Standard Normal Distribution



Non-Gaussian Clock Jitter

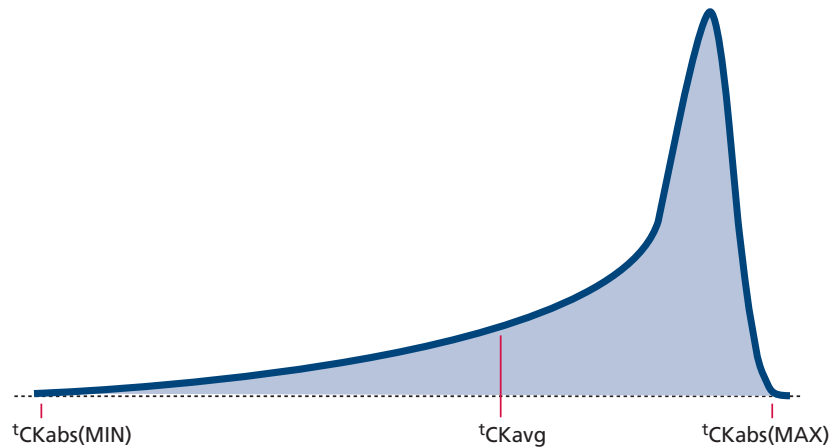
If consecutive clocks are allowed to be at or near $t_{CKabs(MIN)}$ and or $t_{CKabs(MAX)}$ in value, as would be the case with bimodal distribution (see Figure 2), then the DRAM would need a specification relaxation and/or the ability to allow less clock jitter; or manufacturers would have to deal with lower yields and higher costs. Even at that, extremely complex DLL locking circuitry would be required.

Figure 2: Bimodal Distribution



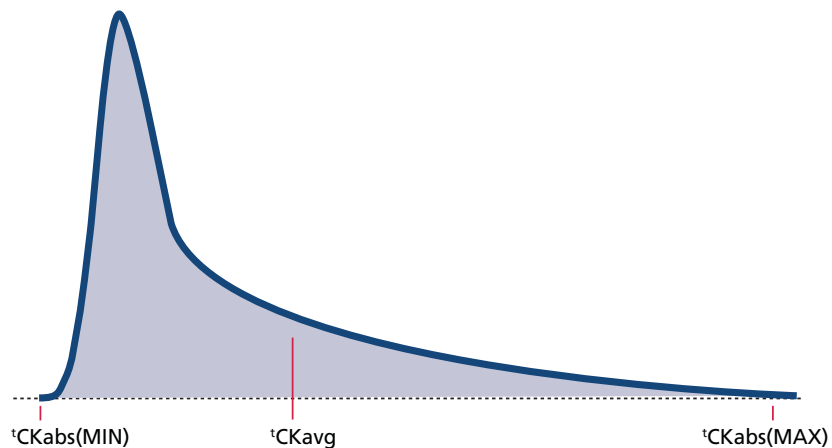
If the clock jitter is not Gaussian but it's close, technically, it would be a specification violation. However, in many cases, it will not adversely affect the DRAM's ability to function correctly. For example, if the DLL is locked and the clock jitter yields a distribution like the one in Figure 3, as long as the $t_{CK(abs)MIN}$ specification limit is not violated, the bulk of the non-Gaussian distribution is adding time to most of the clock pulses, relative to t_{CKavg} , and does not hurt the DRAM's ability to function correctly. The one caveat is that as the clock jitter distribution becomes less Gaussian, the more difficult it will be for the DLL to get a good lock.

Figure 3: Non-Gaussian Distribution with MAX Skew



Conversely, if the DLL is locked and the clock jitter yields a distribution like the one in Figure 4, even if the $t_{CK(abs)MIN}$ specification limit is not violated, the non-Gaussian distribution is losing time to most of the clock pulses relative to t_{CKavg} and could prevent the DRAM from functioning correctly.

Figure 4: Non-Gaussian Distribution with MIN Skew



The best way to determine the level of risk from a clock jitter specification violation (after first minimizing the clock jitter violation as much as possible, of course) is to work with the DRAM supplier. In most cases, the DRAM supplier should be able to help determine whether the existing design is relatively safe or that the clock jitter violation(s) need to be minimized or fixed.

Clock Jitter Specifications

Typically, clock generator specifications state the clock jitter limits in RMS terms, while DDR2/DDR3 SDRAM states clock jitter limits in absolute terms. These are not the same conditions. It is worth noting that random jitter is unbounded (which is why the issuer uses an RMS value), and DRAM clock jitter specifications are absolute values (because the memory supplier cannot test to an RMS limit.) This means that at some point, just about every memory clock will violate clock jitter specifications. Fortunately, in most cases these violations will be in the range of an SER event; that is, so small that it is not an issue. A clock jitter violation does not mean that the DRAM will necessarily fail.

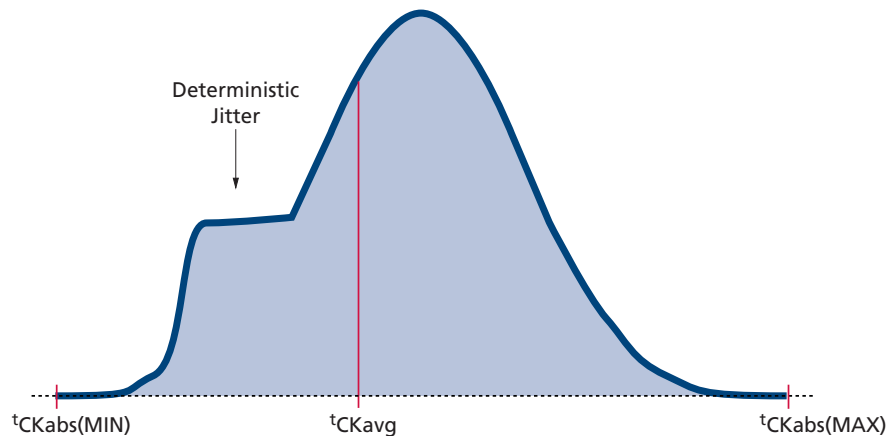
Certain facts, like the fact that clock jitter is unbounded and that the DRAMs' clock jitter specifications are absolute, are often overlooked. It is not unusual for users to measure clock jitter for a small period of time, determine the minimum and maximum clock pulses, and declare a clock jitter value. Yet, there is no mention of how many samples were taken or, more importantly, what the standard deviation was.

Until a statistically sound standard deviation value for the clock jitter is obtained, the results are incomplete and can be grossly misleading. Furthermore, the system needs a pass/fail bit error rate (BER) target before a solid risk assessment can be made. For example, with a 3ns DDR2 SDRAM (DDR2-667) that has a t_{CKavg} of 3.05ns and a clock period jitter (t_{JITper}) of $\pm 125ps$ (the specification limit is 125ps), are there clock jitter specification violations? If there are violations, is the system likely to see errors? Unless the $\pm 125ps$ clock jitter was measured over infinity in time, sooner or later the clock jitter will violate the DRAM's specifications. More importantly, it is impossible to determine the risk level of the clock jitter without knowing what the clock jitter standard deviation is and what the acceptable BER targets are.

Deterministic Clock Jitter

Deterministic clock jitter is predictable and reproducible (see Figure 5) and has specific causes. DDR2/DDR3 performance, and hence yields, would be adversely impacted if deterministic clock jitter was allowed in the specification. When the industry-standard clock jitter specifications were being defined, various system data indicated that the clock jitter should be random and fit a Gaussian distribution. Additionally, it was believed that users should be able to remove deterministic clock jitter from their systems; therefore, no form of deterministic clock jitter is allowed by specification.

Figure 5: Deterministic Clock Jitter



While any form of deterministic clock jitter is considered a specification violation, not all deterministic clock jitter will adversely affect DDR2/DDR3 performance. Once the DLL is locked, most forms of positive deterministic clock jitter should not adversely affect the DRAM's ability to function correctly, provided the clock rate is not near the maximum clock period and cycle-to-cycle jitter (t_{JITcc}) is not violated.

Conversely, if the DLL is locked, most forms of negative deterministic clock jitter could adversely affect the DRAM's ability to function correctly. Even if the $t_{CKabs(MIN)}$ specification limit is not violated, the negative deterministic clock jitter distribution could prevent the DRAM from functioning correctly.

DLL Locking vs. Locked

As previously mentioned, the two primary modes of DDR2/DDR3 operation are (1) when the DLL is locking and (2) when the DLL is locked, and each has different clock jitter concerns and, of course, different specifications. When analyzing clock jitter, it is important to investigate each mode of operation separately. That is to say, cycle-to-cycle clock jitter should be measured twice: once during DLL locking time and once during memory accessing. Using the value for one condition or the other will provide a faulty analysis.

DLL Locking

When the DRAM is powered up and initialized, the DLL must be set and locked. This means the internal delay loop must be locked to the external clock rate. It should be evident how sensitive this operation is to clock jitter. How can a DLL lock to a moving target? If the clock distribution was bimodal instead of Gaussian, what would the DLL lock to? How does it know where the nominal clock is? Gaussian-based clock jitter is a must for obtaining a good DLL lock.

Clock jitter specifications are tighter when the DLL is locking than after the DLL is locked. The tighter limits are necessary to ensure a good lock point. Fortunately, most systems have significantly less clock jitter during initialization, when the DLL is locking, than during normal operation. During initialization, power busing in the memory controller is fairly quiet. During intensive memory operations, however, power busing in the memory controller can get fairly noisy, which in turn can cause the clock generator to increase variation, resulting in increased clock jitter.

Although there are about 17 clock jitter specifications, the majority of them are either not applicable or they have a minimal effect on the DLL's ability to get a good lock. On a first order, as long as the cycle-to-cycle jitter is small (and maintains a Gaussian distribution), the DLL should be able to get a good lock. If deterministic clock jitter is present during the DLL locking period, it will be problematic because it is unlikely that the DLL will be able to find the nominal frequency.

DLL Locked

After the DLL is locked, clock jitter takes on a new light. First of all, the DLL becomes fairly immune to cycle-to-cycle clock jitter because the DLL filters out random clock jitter that is Gaussian since the internal loop delay remains constant. The variations in clock pulse widths due to clock jitter will pass through to the outputs, so to speak; however, the delay itself will not change due to cycle-to-cycle clock jitter.

It is interesting to note that during the DLL locking period, the DRAM is rather sensitive to cycle-to-cycle clock jitter; but after the DLL is locked, the DRAM is rather impervious to cycle-to-cycle clock jitter. Of course, this is assuming that the clock jitter has a good Gaussian distribution.

Clock jitter specifications that have minimal impact during the DLL locking stage are critical to the DRAM's functionality after the DLL is locked. Clock period jitter, ${}^t\text{JITper}$, and the duty cycle jitter, ${}^t\text{JITdty}$, are the two clock jitter specifications that play key roles after the DLL has been locked, when the DRAM is being accessed. The various ${}^t\text{ERRnper}$ jitter specifications play a lesser role. In fact, if the clock jitter is close to a Standard Normal distribution and the ${}^t\text{JITper}$ limits are not violated, the various ${}^t\text{ERRnper}$ jitter specifications are all but guaranteed to be satisfied.

DDR2/DDR3 Functionability

Up to this point, the discussion has focused on clock jitter in terms of a DRAM's functionability as opposed to it working correctly; there is a subtle but important difference. DDR2/DDR3 clock jitter specifications are applied to input timings only; output timings are stated without any clock jitter and any clock jitter effects must be added to them. Suffice it to say that there is a reasonably good explanation why this became the industry-standard methodology. Thus, clock jitter analysis needs to be separated between input timings concerns (will the device function correctly?) and output timing concerns (will the data eye be big enough?)

Input Timings

DDR2/DDR3 input timings incorporate the clock jitter limits. As long as the clock jitter specifications are satisfied, all the input timings limits can be used; they do not require additional derating to account for the clock jitter. Thus, functionability is tested to ensure it is not adversely affected by clock jitter as long as the clock jitter is within specification limits.

Going back to the earlier example of a 3ns DDR2 SDRAM (DDR2-667) with a $t_{CKavg}^{(MIN)}$ of 3ns and a t_{JITper} limit of $\pm 125ps$ results in a clock period as small as 2.875ps.

As far as the input timings are concerned, the maximum jitter values do not adversely affect DRAM functionability. Take the above example and assume it violated the $t_{JITper}^{(MAX)}$ limit of +125ps with a maximum of 250ps. That is a serious violation, but in terms of the input timings, the only penalty is that the clock period is extended from 3ns to 3.25ns.

Thus, clock jitter that is greater than the minimum clock jitter specification limits—and even clock jitter that exceeds the maximum clock jitter specification limits—ensures proper functionability. Clock jitter that violates one of the minimum clock jitter specification limits may not necessarily result in improper functionability. Unlike input timing limits that must be met to ensure proper functionability, most clock jitter violations against input timing limits can be neutralized, as will be discussed further on.

Considering that the clock jitter has a random distribution, it is likely that clock jitter specification limits will be violated in most cases. Neutralizing the clock jitter effects on the input timings will be required in many of these cases.

Output Timings

An often overlooked concern is that the output timings do not account for allowable clock jitter. Output timing specification limits are provided as if there is zero clock jitter. This means that the output timing specifications require that any clock jitter be derated from the output timing specification limits. The DDR2/DDR3 specifications provide a specific derating factor for each affected output timing specification. Generally speaking, two times the amount of clock jitter that goes into the device must be derated at the outputs. It behooves the system designer to not simply meet the clock jitter specification limits but to keep them to the absolute minimum.

For example, consider a 3ns DDR2 SDRAM (DDR2-667) that has an actual t_{JITper} of $-100ps$ and $+110ps$ versus a specification limit of $\pm 125ps$. Although the t_{AC} specification limit is $\pm 450ps$; after derating to account for the clock jitter, the derated t_{AC} specification limit is approximately $-650ps$ ($-450ps - 2 * +100ps$) and $+670ps$ ($+450ps - 2 * -110ps$). The

actual derating value is determined by the $t_{ERR5per}$ parameter and not twice t_{JITper} ; however, the two are very close in value. For example, the DDR2-667 has a t_{JITper} specification limit of $\pm 125ps$ and a $t_{ERR5per}$ specification limit of $\pm 250ps$.

It is worth noting that DDR3 requires the use of $t_{ERR10per}$ instead of $t_{ERR5per}$ and that the guideline is that two and one-half times the amount of whatever clock jitter goes into a DDR3 SDRAM must be derated at the outputs.

Clock Jitter Derating

DDR2/DDR3 input timings do not require derating as long as the clock jitter specifications are not violated. If the clock jitter specification limits are exceeded, the violations may be neutralized or the input timings may be derated to render the DRAM fully functional again. Even though the clock jitter specifications are violated, the DRAM functionality can be maintained, provided the violations can be adjusted for.

The one exception is cycle-to-cycle clock jitter during DLL locking; this condition cannot be neutralized against clock jitter. Fortunately, clock jitter is usually small during DLL locking time. That being said, the closer the cycle-to-cycle clock jitter distribution is to a Standard Normal distribution, the more robust the DLL will be in locking to the nominal clock rate.

DDR2/DDR3 output timings require derating if any clock jitter is present. In theory, this means every design's output timing should account for some form of derating when analyzing the timing budgets. Examples of each case are provided below and assume the DLL is in the locked state.

Clock Jitter Violations on Input Timing

When clock jitter specifications are violated, the DRAM's functionality is at risk because the input timing parameters are compromised. It is possible to offset the clock jitter by derating all the input timings by the amount of the clock jitter violation. Since the clock jitter distribution is Gaussian and the DLL is locked, the $t_{ERRnper}$ and t_{JITcc} specifications can generally be ignored. This leaves t_{JITper} and t_{JITdy} that require attention. When investigating input timing parameters for clock jitter violations, the positive violations are of no concern because the positive violations actually add margin to the DDR2/DDR3 input timing requirements. The negative clock jitter violations, on the other hand, steal time and must be accounted for to ensure proper functionality. Input derating/neutralizing clock jitter violations assume negative clock jitter violations, unless stated otherwise.

Input timings fall in to three classes: setup/hold parameters, time-based parameters (require a time value converted in to whole clocks), and n -clock period parameters (require a specific number of clocks to satisfy.) Input timing parameters that are a setup or hold specification are not sensitive to clock jitter because they are referenced to a clock edge that moves. Thus, clock jitter violations do not adversely affect parameters such as address setup (t_{IS}) and hold (t_{IH} .)

Time-based parameters and n -clock parameters are sensitive to clock jitter violations. Time-based parameters, such as write recovery (t_{WR}), can be derated by the amount that t_{JITper} violates the specification limit or it can be neutralized by reducing the clock rate by the amount that $t_{JITper(MIN)}$ violates the specification limit. Unlike time-based parameters, n -clock parameters, such as t_{CKE} , cannot be derated to account for clock jitter violations because they are specified in clock ticks; however, they can be neutralized by reducing the clock rate by the amount that $t_{JITper(MIN)}$ violates the specification limit.

Neutralizing Clock Jitter on Input Timing Example

To see how clock jitter violations can be neutralized by reducing the clock rate by the amount $t_{JITper}^{(MIN)}$ violates the specification limit, we can use the same 3ns DDR2 SDRAM (DDR2-667) example as before with the following specifications:

- $t_{JITper}^{(MIN)}$ limit is -125ps
- $t_{JITdy}^{(MIN)}$ limit is -125ps
- $t_{CKavg}^{(MIN)}$ limit is 3.0ns
- $t_{CKabs}^{(MIN)}$ limit is 2.875ns ($t_{CKavg} - t_{JITper}$)
- $t_{CHavg}^{(MIN)}$, $t_{CLavg}^{(MIN)}$ limit is 1.44ns ($0.48 * t_{CKavg}$)
- $t_{CHabs}^{(MIN)}$, $t_{CLabs}^{(MIN)}$ limit is 1.315ns ($0.48 * t_{CKavg} - t_{JITdy}$)

Assume the clock actually has $t_{JITper}^{(MIN)}$ of -225ps and $t_{JITdy}^{(MIN)}$ of -200ps. These values violate the clock jitter specifications by 100ps and 75ps, respectively. (DDR3 does not specify t_{JITdy} ; instead, it relies on meeting t_{CHabs} and t_{CLabs} limits). If the input clock rate (t_{CKavg}) is increased from 3ns to 3.1ns and the minimum half-period clock pulse is increased by 75ps, or until $t_{CHabs}^{(MIN)}$ and $t_{CLabs}^{(MIN)}$ are no less than 1.315ns, the clock jitter violations will have been and the DRAM can be expected to function correctly, with one caveat: Is the standard deviation larger than 10–20ps? If so, more work may need to be done. The importance of the clock jitter's standard deviation and how to apply it is addressed in the Clock Jitter and Statistics section.

Derating Output Timing Due to Clock jitter

DDR2/DDR3 output timings should always be derated to account for clock jitter because output timing specifications are applicable when there is zero clock jitter. The only difference between passing and failing clock jitter values is the amount that the output timings need to be derated by. Unlike input timings, both positive and negative clock jitter violations are of concern because the positive violations increase the level of uncertainty regarding the positive output timing requirements, as shown in Table 1.

Output timings that require derating fall into one of three types: DLL-derived, full-clock, and half-clock. DLL-derived parameters are significantly affected by clock jitter. The amount of clock jitter basically gets doubled at the output for DLL-derived parameters; DDR2 derating is $t_{ERR5per}$, or approximately $2 * t_{JITper}$, and DDR3 derating is $t_{ERR10per}$, or approximately $2.5 * t_{JITper}$. Thus, whatever clock jitter goes into the DRAM, the DLL-derived output parameters require at least double that in output derating.

Using the previous 3ns DDR2 SDRAM (DDR2-667) example, the t_{DQCK} specification limit is $\pm 400ps$, the t_{JITper} limit is $\pm 125ps$, and the $t_{ERR5per}$ limit is $\pm 250ps$. (Note: $t_{ERR5per}$ is two times t_{JITper} .) With a measured t_{JITper} of $\pm 100ps$, which is not a violation of the clock jitter specifications, t_{DQCK} would need to be derated to approximately $\pm 600ps$; the minimum is $-400ps - 200ps = -600ps$ and the maximum is $400ps - (-200ps) = 600ps$.

Full-clock output parameters require derating by t_{JITper} . Half-clock output parameters require derating by t_{JITdy} , which in the case of DDR3, would be the larger result of $0.47 * t_{CK} - t_{CHabs}$ or $0.47 * t_{CK} - t_{CLabs}$. Table 1 below summarizes which output parameters get derated and by which clock jitter specification. When developing the timing budget, the clock jitter specification limits should be used; but when system timings are being verified, the actual measured clock jitter values should be used.

Table 1: Output Timing Derating

Output Type	Parameter	DDR2 Derating Factor		DDR3 Derating Factor	
		Min ¹	Max ¹	Min ¹	Max ¹
DLL Derived	^t AC	^t ERR5per(MAX)	^t ERR5per(MIN)	na	na
	^t DQSCK	^t ERR5per(MAX)	^t ERR5per(MIN)	^t ERR10per(MAX)	^t ERR10per(MIN)
	^t LZDQS	^t ERR5per(MAX)	^t ERR5per(MIN)	^t ERR10per(MAX)	^t ERR10per(MIN)
	^t LZDQ	^t ERR5per(MAX)	^t ERR5per(MIN)	^t ERR10per(MAX)	^t ERR10per(MIN)
	^t HZ	na	^t ERR5per(MIN)	na	^t ERR10per(MIN)
	^t AON	^t ERR5per(MAX)	^t ERR5per(MIN)	^t ERR10per(MAX)	^t ERR10per(MIN)
	^t AOF	^t ERR5per(MAX) + ^t JITdty(MAX)	^t ERR5per(MIN) + ^t JITdty(MIN)	^t ERR10per(MAX) + ^t JITdty(MAX)	^t ERR10per(MIN) + ^t JITdty(MIN)
Full-Clock	^t RPRE	^t JITper(MAX)	^t JITper(MIN)	^t JITper(MAX)	^t JITper(MIN)
	^t QSH	na	na	^t JITper(MAX)	^t JITper(MIN)
	^t QSL	na	na	^t JITper(MAX)	^t JITper(MIN)
Half-Clock	^t RPST	^t JITdty(MAX)	^t JITdty(MIN)	^t JITdty(MAX) ²	^t JITdty(MIN) ²
	^t QH	^t JITdty(MAX)	na	^t JITdty(MAX) ²	na

- Notes:
1. Derating factor is subtracted from the specification limit.
 2. Some standards suggest using ^tJITper.

When the clock rate is adjusted to neutralize the effects that clock jitter violations have on input timing specifications, the amount that the clock period is increased by adds margin to the data-eye, offsetting the amount by which the output timings need to be derated. If the derated output timings do not allow the data-eye to close the timing budget, the clock period can be further increased until enough margin is acquired to close the timing.

Example of Derating Output Timing Due to Clock Jitter

Table 2 shows the output derating required for two conditions using the previous 3ns DDR2 SDRAM (DDR2-667) example and the formulas in Table 1 for both timing budget analysis (derate to specifications) and a case of measured clock jitter values. The timing budget analysis uses the specification limits of $\pm 125\text{ps}$ for t_{JITper} and t_{JITdy} and $\pm 250\text{ps}$ for t_{ERR5per} . The measured analysis uses passing clock jitter specification values of -50ps and $+75\text{ps}$ for t_{JITper} and t_{JITdy} and $\pm 150\text{ps}$ for t_{ERR5per} , but uses the parameter specification limit. The measured derating could likely have smaller values if actual measured values are used.

Table 2: DDR SDRAM Output Timing Derating

Output Type	Parameter	DDR2-667 Specification		Derated to Specification		Derated to Measured	
		Min	Max	Min	Max	Min	Max
DLL-Derived	t_{AC}	-450ps	450ps	-700ps	700ps	-600ps	600ps
	t_{DQCK}	-400ps	400ps	-650ps	650ps	-550ps	550ps
	t_{LZDQS}	-450ps	450ps	-700ps	700ps	-600ps	600ps
	t_{LZDQ}	-900ps	900ps	-1150ps	1150ps	-1050	1050ps
	t_{HZ}	na	450ps	na	700ps	na	600ps
	t_{AON}	-450ps	1150ps	-700ps	1400ps	-600ps	1300ps
	t_{AOF}	-450ps	1050ps	-825ps	1425ps	-675ps	1250ps
Full-Clock	t_{RPRE}	2700ps	3300ps	2575ps	3425ps	2625ps	3350ps
Half-Clock	t_{RPST}	1200ps	1800ps	1075ps	1925ps	1125ps	1850ps
	t_{QH}	1100ps	na	975ps	na	1025ps	na

After reviewing the results in Table 2, it should be readily apparent how important it is to keep clock jitter to an absolute minimum. Simply meeting the clock jitter specification limits can be costly to the output timings.

Clock Jitter and Statistics

As previously stated, the measured clock jitter values are not absolute, but the specification limits are. This presents a problem: is the measured clock jitter a passing value if it is not violating the absolute specification limit? It is unlikely since clock jitter is a random event and it is not possible to measure the worst-case clock jitter event. So, whether a measured value is passing or failing depends on several factors.

Recall that if the clock jitter specifications are violated, the clock jitter's adverse effect on the input timings can be neutralized by increasing the clock period. This suggests that the clock jitter value itself is not the issue; the issue is how much the input timing is adversely affected by the clock jitter.

Also as previously mentioned, until a statistically sound standard deviation (σ) value for the clock jitter is obtained, the clock jitter investigation is incomplete. A statistically sound standard deviation value depends on two factors: 1) the sample size used to measure the clock jitter; and 2) the system BER target.

When investigating clock jitter statistically, only the negative clock jitter only needs to be looked at. For device functionality concerns, the DRAM is adversely affected by negative clock jitter violations, not positive clock jitter violations. And, when the output timings are evaluated, the positive clock jitter results should mirror the negative clock

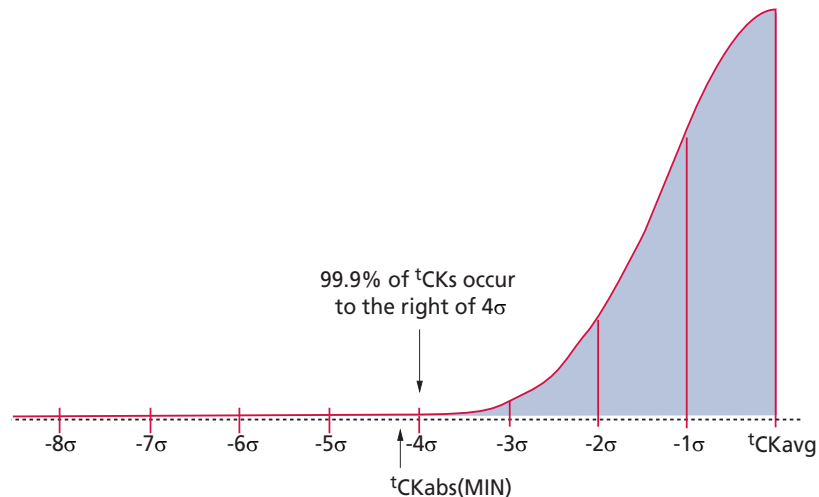
jitter results. Additionally, since the clock jitter is Gaussian, all the clock jitter values should track each other. Analyzing t_{JITper} and how it is compensated for generally ensures that the other clock jitter specifications will not be an issue.

Clock Jitter Sample Size and Standard Deviation

From past experience, most if not all clock jitter samples are evaluated without any regard for sigma. After inquiring, it was generally found that the clock jitter values were acquired with enough samples to yield a clock jitter value that was about 3 or 4 sigmas. For example, if $t_{JITper}(MIN)$ was measured to be $-120ps$, then a sigma in the range of 4 would not be unexpected. This would mean that sigma is about $30ps$. Experience has shown that until enough clock jitter samples are acquired to obtain a $t_{JITper}(MIN)$ value that is 6 sigma away from the nominal clock, or t_{CKavg} , not enough samples have been acquired to obtain a reasonably stable sigma.

Using the previous $3ns$ DDR2 SDRAM (DDR2-667) example with the measured $t_{JITper}(MIN)$ of $-120ps$ and $30ps$ sigma, it would appear the $t_{JITper}(MIN)$ limit of $-125ps$ is not violated and the system is good to go. Even then, considering that clock jitter is random and results in a Gaussian distribution, a 4-sigma clock jitter means almost all of the clock periods are within specification. Less than 0.1% of the clock pulse widths will be lower than specifications allow, as shown in Figure 6.

Figure 6: 4 Sigma t_{JITper} Example



BER Targets

Although the previous clock jitter analysis examples seem to provide a positive outcome; further study actually reveals the measured clock jitter is predicting a noticeable potential for failure. The reason it is a “potential failure rate” is because the clock jitter analysis is only identifying clock timing errors and not all clock errors result in a DDR2/DDR3 functional failure.

The analysis does not go far enough because it didn’t determine what the system BER target is and how well the clock jitter would respond to it. Applying DDR2-667 to clock jitter with $t_{JITper}(MIN)$ of $-120ps$ and a sigma of $30ps$, approximately one clock out of every 31,574 clocks will have one clock period less than the minimum allowed at DDR2-667, as shown in Table . This means the clock period will be too small once every $94,700ns$, or $94.7\mu s$. –

Table 3: Standard Normal Distribution Probabilities: BER at DDR2-667, 4 Sigma

%	Sigma	probability of occurrence - one in "x"	ns	sec	min	hr	day	Month	Year
1.59E+01	1	6.3	1.89E+01	1.89E-08	3.2E-10	5.3E-12	2.2E-13	7.4E-15	6.1E-16
2.28E+00	2	44	1.32E+02	1.32E-07	2.2E-09	3.7E-11	1.5E-12	5.1E-14	4.3E-15
1.35E-01	3	741	2.22E+03	2.22E-06	3.7E-08	6.2E-10	2.6E-11	8.7E-13	7.2E-14
3.17E-03	4	31,574	9.47E+04	9.47E-05	1.6E-06	2.6E-08	1.1E-09	3.7E-11	3.1E-12

If the initial clock jitter analysis stopped with the measured $t_{JITper}(MIN)$ of $-120ps$ and ignored the sigma, it would be the same as saying the system is allowed to have a BER of 4 sigma, resulting in a minimum clock period violation once every $94.7\mu s$.

The system BER for clock period violations due to clock jitter must be defined before a thorough clock jitter analysis can be completed. DDR2/DDR3 have a defined SER, which means the DRAM will fail once in awhile. Since the DDR2/DDR3 SER rate is so low, the defacto BER for SER events is extremely high and is ignored. Clock timing has also had a high defacto BER for clock jitter related violations; however, as clock speeds have increased, clock jitter has become a large enough percentage of the clock period that it should no longer be ignored.

All memory systems do not require the same BER for clock jitter, but surely something more than 4 sigma is required. Six sigma is often used as a good quality target. With the $t_{JITper}(MIN)$ 6 sigma measurement equal to the $t_{JITper}(MIN)$ specification limit, a probability of one clock period violation due to $t_{JITper}(MIN)$ will occur once every 3.04s, as seen in Table . Such a low BER is unlikely to satisfy most memory application needs.

Table 4: Standard Normal Distribution Probabilities: BER at DDR2-667, 6 Sigma

%	Sigma	Probability of Occurrence – 1 in "x"	ns	sec	min	hr	day	Month	Year
1.59E+01	1	6.3	1.89E+01	1.89E-08	3.2E-10	5.3E-12	2.2E-13	7.4E-15	6.1E-16
2.28E+00	2	44	1.32E+02	1.32E-07	2.2E-09	3.7E-11	1.5E-12	5.1E-14	4.3E-15
1.35E-01	3	741	2.22E+03	2.22E-06	3.7E-08	6.2E-10	2.6E-11	8.7E-13	7.2E-14
3.17E-03	4	31,574	9.47E+04	9.47E-05	1.6E-06	2.6E-08	1.1E-09	3.7E-11	3.1E-12
2.87E-05	5	3,488,556	1.05E+07	1.05E-02	1.7E-04	2.9E-06	1.2E-07	4.1E-09	3.4E-10
9.87E-08	6	1,013,594,635	3.04E+09	3.04E+00	5.1E-02	8.4E-04	3.5E-05	1.2E-06	9.9E-08

BER targets need to be much higher. BER targets in the range of 10 to 11 sigma should satisfy most memory application needs. (See Table 5.) A BER target of 10 sigma translates to a t_{JITper} sigma of 9–12ps for DDR2 and 7–10ps for DDR3.

Table 5: BER at DDR2-667 with 10,11 Sigma Targets

%	Sigma	Probability of Occurance – 1 in "x"	ns	sec	min	hr	day	Month	Year
1.59E+01	1	6.3	1.89E+01	1.89E-08	3.2E-10	5.3E-12	2.2E-13	7.4E-15	6.1E-16
2.28E+00	2	44	1.32E+02	1.32E-07	2.2E-09	3.7E-11	1.5E-12	5.1E-14	4.3E-15
1.35E-01	3	741	2.22E+03	2.22E-06	3.7E-08	6.2E-10	2.6E-11	8.7E-13	7.2E-14
3.17E-03	4	31,574	9.47E+04	9.47E-05	1.6E-06	2.6E-08	1.1E-09	3.7E-11	3.1E-12
2.87E-05	5	3,488,556	1.05E+07	1.05E-02	1.7E-04	2.9E-06	1.2E-07	4.1E-09	3.4E-10
9.87E-08	6	1,013,594,635	3.04E+09	3.04E+00	5.1E-02	8.4E-04	3.5E-05	1.2E-06	9.9E-08
1.28E-10	7	781,332,343,402	2.34E+12	2.34E+03	3.9E+01	6.5E-01	2.7E-02	9.1E-04	7.6E-05
8.66E-12	8	11,547,941,474,051	3.46E+13	3.46E+04	5.8E+02	9.6E+00	4.0E-01	1.4E-02	1.1E-03
5.30E-13	9	188,802,792,827,740	5.66E+14	5.66E+05	9.4E+03	1.6E+02	6.6E+00	2.2E-01	1.8E-02
2.95E-14	10	3,391,460,775,180,010	1.02E+16	1.02E+07	1.7E+05	2.8E+03	1.2E+02	4.0E+00	3.3E-01
1.50E-15	11	66,531,501,179,552,800	2.00E+17	2.00E+08	3.3E+06	5.5E+04	2.3E+03	7.8E+01	6.5E+00

Take the previous example of $t_{JITper}(MIN)$ of -120ps with a sigma of 30ps. A BER of 4 sigma or 6 sigma is not acceptable for almost any memory system. A BER of 10 sigma may be acceptable since the clock violation is only once every four months of continuous clocking. At 10 sigma, the clock period can be expected to be violated by 175ps (10 * 30ps - 125ps). So, instead of the initial clock jitter analysis suggesting the clock jitter was within specifications, it should show that the clock period t_{CKavg} should be increased 175ps to ensure a satisfactory BER for clock jitter effect and acceptable DDR2/DDR3 functionality.

Conclusion

DDR2/DDR3 clock jitter is commonly misunderstood. It is extremely important to analyze clock jitter both when the DLL is locking and when it is locked. When the DLL is locking, the cycle-to-cycle clock jitter must be kept very low. Once the DLL is locked, the DRAM is fairly insensitive to cycle-to-cycle clock jitter, but the clock period jitter and half-period clock jitter need to be analyzed.

It is extremely important to obtain the clock period clock jitter's sigma when measuring clock jitter. Without knowing the sigma, it is impossible to know if the measured jitter value is acceptable. Once the clock jitter is measured and a statistically sound standard deviation is obtained, clock jitter violations can be neutralized; that is, the clock jitter may violate specifications but it can still be allowed if the timings are adjusted to account for the additional error.



8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900

prodmktg@micron.com www.micron.com Customer Comment Line: 800-932-4992

Micron, the M logo, and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners.