

Successful DDR2 Design

Mentor Graphics highlights design issues and solutions for DDR2, the latest trend in memory design.

by Steve McKinney
HyperLynx Technical Marketing Engineer
Mentor Graphics
steven_mckinney@mentor.com

The introduction of the first SDRAM interface, in 1997, marked the dawn of the high-speed memory interface age. Since then, designs have migrated through SDR (single data rate), DDR (double data rate), and now DDR2 memory interfaces to sustain increasing bandwidth needs in products such as graphics accelerators and high-speed routers. As a result of its high-bandwidth capabilities, DDR and DDR2 technology is used in nearly every sector of the electronics design industry – from computers and networking to consumer electronics and military applications.

DDR technology introduced the concept of “clocking” data in on both a rising and falling edge of a strobe signal in a memory interface. This provided a 2x bandwidth improvement over an SDR interface with the same clock speed. This, in addition to faster clock frequencies, allowed a single-channel DDR400 interface with a 200 MHz clock to support up to 3.2 GB/s, a 3x improvement over the fastest SDR interface. DDR2 also provided an additional 2x improvement in bandwidth over its DDR predecessor by doubling the maximum clock frequency to 400 MHz. Table 1 shows how the progression from SDR to DDR and DDR2 has allowed today’s systems to maintain their upward growth path.

SDR		DDR				DDR2			
PC100	PC133	DDR - 200	DDR - 266	DDR - 333	DDR - 400	DDR2 - 400	DDR2 - 533	DDR2 - 667	DDR2 - 800
0.8	1.1	1.6	2.1	2.7	3.2	3.2	4.266	5.33	6.4
Single Channel Bandwidth (GB/s)									

Table 1 – The progression from SDR to DDR and DDR2 has allowed today’s systems to maintain their upward growth path. Speed grades and bit rates are shown for each memory interface.

With any high-speed interface, as supported operating frequencies increase it becomes progressively more difficult to meet signal integrity and timing requirements at the receivers. Clock periods become shorter, reducing timing budgets to a point where you are designing systems with only picoseconds of setup or hold margins. In addition to these tighter timing budgets, signals tend to deteriorate because faster edge rates are needed to meet these tight timing parameters. As edge rates get faster, effects like overshoot, reflections, and crosstalk become more significant problems on the interface, which results in a negative impact on your timing budget. DDR2 is no exception, though the JEDEC standards committee has created several new features to aid in dealing with the adverse effects that reduce system reliability.

Some of the most significant changes incorporated into DDR2 include on-die termination for data nets, differential strobe signals, and signal slew rate derating for both data and address/command signals. Taking full advantage of these new features will help enable you to design a robust memory interface that will meet both your signal integrity and timing goals.

On-Die Termination

The addition of on-die termination (ODT) has provided an extra knob with which to dial in and improve signal integrity on the DDR2 interface. ODT is a dynamic termination built into the SDRAM chip and memory controller. It can be enabled or disabled depending on addressing conditions and whether a read or write operation is being performed, as shown in Figure 1. In addition to being able to turn termination off or on, ODT also offers the flexibility of different termi-

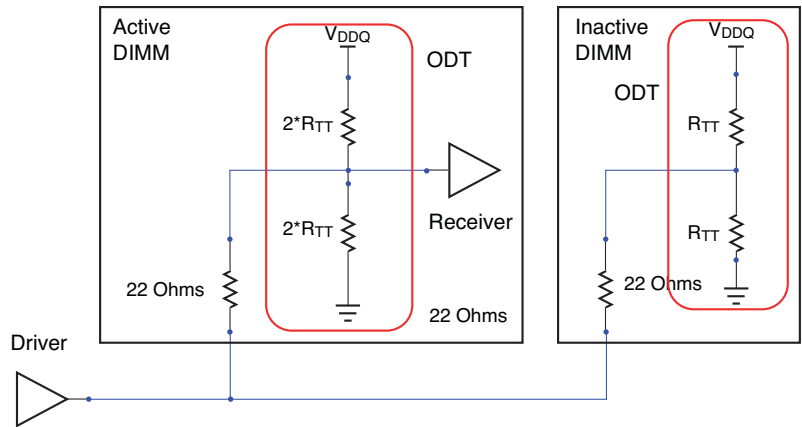


Figure 1 – An example of ODT settings for a write operation in a 2 DIMM module system where $R_{TT} = 150$ Ohms.

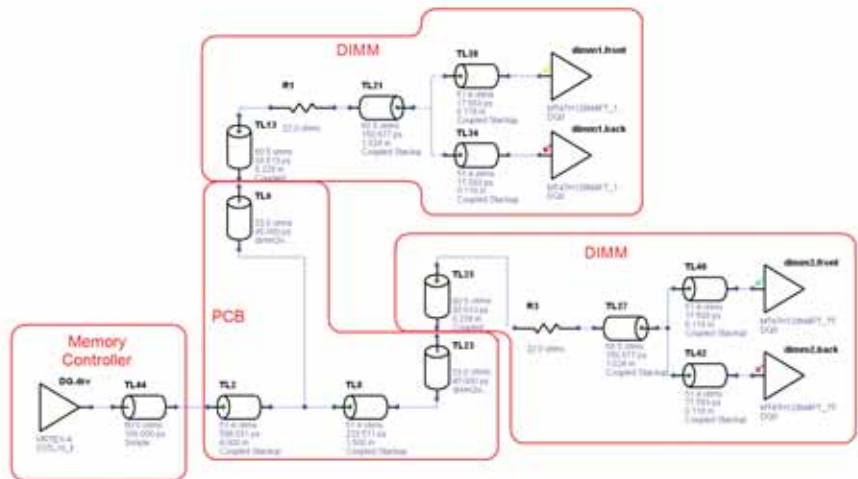


Figure 2 – The HyperLynx free-form schematic editor shows a pre-layout topology of an unbuffered 2 DIMM module system. Transmission line lengths on the DIMM are from the JEDEC DDR2 unbuffered DIMM specification.

nation values, allowing you to choose an optimal solution for your specific design.

It is important to investigate the effects of ODT on your received signals, and you can easily do this by using a signal integrity software tool like Mentor Graphics’ HyperLynx product. Consider the example design shown in Figure 2, which shows a DDR2-533 interface (266 MHz) with two

unbuffered DIMM modules and ODT settings of 150 Ohms at each DIMM. You can simulate the effects of using different ODT settings and determine which settings would work best for this DDR2 design before committing to a specific board layout or creating a prototype.

With the 150 Ohm ODT settings, Figure 3 shows significant signal degrada-

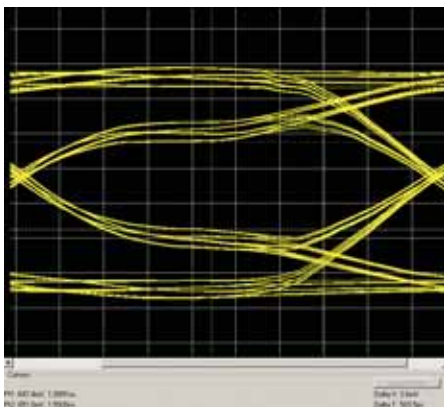


Figure 3 – The results of a received signal at the first DIMM in eye diagram form. Here, ODT settings of 150 Ohms are being used at both DIMM modules during a write operation. The results show there is an eye opening of approximately 450 ps outside of the VinAC switching thresholds.

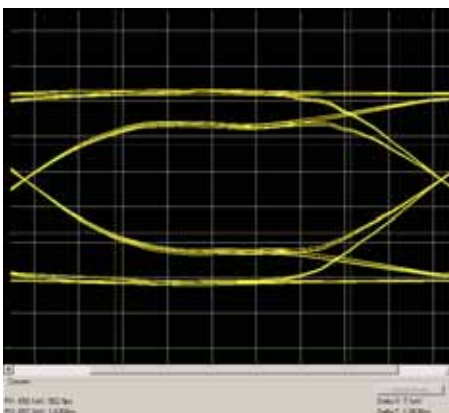


Figure 4 – This waveform shows a significant improvement in the eye aperture with a new ODT setting. Here, the ODT setting is 150 Ohms at the first DIMM and 75 Ohms at the second DIMM. The signal is valid for 1.064 ns with the new settings, which is an increase of 614 ps from the previous ODT settings.

tion at the receiver, resulting in eye closure. The eye shows what the signal looks like for all bit transitions of a pseudo-random (PRBS) bitstream, which resembles the data that you might see in a DDR2 write transaction. Making some simple measurements of the eye where it is valid outside the VinhAC and VinlAC thresholds, you can see that there is roughly a 450 ps window of valid signal at the first DIMM module.

It is appropriate to try to improve this eye aperture (opening) at the first DIMM if possible, and changing the ODT setting is one of the options available for this. To improve the signal quality at the first

DIMM, you must change the ODT value at the second DIMM. Setting the ODT at the second DIMM to 75 Ohms and re-running the simulation, Figure 4 shows more than a 100 percent increase in the eye aperture at the first DIMM, resulting in a 1.06 ns eye opening. As you can see, being able to dynamically change ODT is a powerful capability to improve signal quality on the DDR2 interface.

With respect to a DDR interface, ODT allows you to remove the source termination, normally placed at the memory controller, from the board. In addition, the pull-up termination to VTT at the end of the data bus is no longer necessary. This reduces component cost and significantly improves the layout of the board. By removing these terminations, you may be able to reduce layer count and remove unwanted vias on the signals used for layer transitions at the terminations.

Signal Slew Rate Derating

A challenging aspect of any DDR2 design is meeting the setup and hold time requirements of the receivers. This is especially true for the address bus, which tends to have significantly heavier loading conditions than the data bus, resulting in fairly slow edge rates. These slower edge rates can consume a fairly large portion of your timing budget, preventing you from meeting your setup and hold time requirements.

To enable you to meet the setup and hold requirements on address and data

buses, DDR2’s developers implemented a fairly advanced and relatively new timing concept to improve timing on the interface: “signal slew rate derating.” Slew rate derating provides you with a more accurate picture of system-level timing on the DDR2 interface by taking into account the basic physics of the transistors at the receiver.

For DDR2, when any memory vendor defines the setup and hold times for their component, they use an input signal that has a 1.0V/ns input slew rate. What if the signals in your design have faster or slower slew rates than 1.0V/ns? Does it make sense to still meet that same setup and hold requirement defined at 1.0V/ns? Not really. This disparity drove the need for slew rate derating on the signals specific to your design.

To clearly understand slew rate derating, let’s consider how a transistor works. It takes a certain amount of charge to build up at the gate of the transistor before it switches high or low. Consider the 1.0V/ns slew rate input waveform between the switching region, Vref to Vin(h/l)AC, used to define the setup and hold times. You can define a charge area under this 1.0V/ns curve that would be equivalent to the charge it takes to cause the transistor to switch. If you have a signal that has a slew rate faster than 1.0V/ns, say 2.0V/ns, it transitions through the switching region much faster and effectively improves your timing margin. You’ve added some amount of timing margin into your system, but that was with the assumption of using the stan-

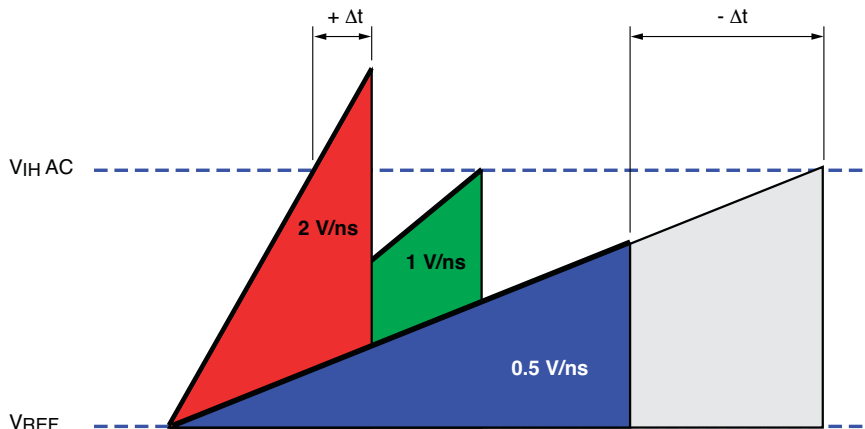


Figure 5 – A 1V/ns signal has a defined charge area under the signal between Vref and VinhAC. A 2V/ns signal would require a + Δt change in time to achieve the same charge area as the 1V/ns signal. A 0.5V/ns signal would require a - Δt change in time to achieve the same charge area as the 1V/ns signal. This change in time provides a clearer picture of the timing requirements needed for the receiver to switch.

standard setup and hold times defined at 1.0V/ns. In reality, you haven't allowed enough time for the transistor to reach the charge potential necessary to switch, so there is some uncertainty that is not being accounted for in your system timing budget. To guarantee that your receiver has enough charge built up to switch, you have to allow more time to pass so that sufficient charge can accumulate at the gate.

Once the signal has reached a charge area equivalent to the 1.0V/ns curve between the switching regions, you can safely say that you have a valid received signal. You must now look at the time difference between reaching the V_{inAC} switching threshold and the amount of time it took for the 2.0V/ns to reach an equivalent charge area, and then add that time difference into your timing budget, as shown in Figure 5.

Conversely, if you consider a much slower slew rate, such as 0.1V/ns, it would take a very long time to reach the switching threshold. You may never meet the setup and hold requirements in your timing budget with that slow of a slew rate through the transition region. This could cause you to overly constrain the design of your system, or potentially limit the con-

figuration and operating speed that you can reliably support. But again, if you consider the charge potential at the gate with this slow slew rate, you would be able to subtract some time out of your budget (as much as 1.42 ns under certain conditions) because the signal reached an equivalent charge area earlier than when it crossed the V_{inAC} threshold.

To assist you in meeting these timing goals, the memory vendors took this slew rate information into account and have constructed a derating table included in the DDR2 JEDEC specification (JESD79-2B on www.jedec.com). By using signal derating, you are now considering how the transistors at the receiver respond to charge building at their gates in your timing budgets. Although this adds a level of complexity to your analysis, it gives you more flexibility in meeting your timing goals, while also providing you with higher visibility into the actual timing of your system.

Determining Slew Rate

To properly use the derating tables, it is important to know how to measure the slew rate on a signal. Let's look at an example of a slew rate measurement for the rising edge of a signal under a setup condition.

The first step in performing signal derating is to find a nominal slew rate of the signal in the transition region between the V_{ref} and $V_{in(h/l)AC}$ threshold. That nominal slew rate line is defined in the JEDEC specification as the points of the received waveform and V_{ref} and V_{inAC} for a rising edge, as shown in Figure 6.

It would be a daunting task to manually measure each one of your signal edges to determine a nominal slew rate for use in the derating tables toward derating each signal. To assist with this process, HyperLynx simulation software includes built-in measurement capabilities designed specifically for DDR2 slew rate measurements. This can reduce your development cycle and take the guesswork out of trying to perform signal derating. The HyperLynx oscilloscope will automatically measure each of the edge transitions on the received waveform, reporting back the minimum and maximum slew rate values, which can then be used in the JEDEC derating tables. The scope also displays the nominal slew rate for each edge transition, providing confidence that the correct measurements are being made (see Figure 7).

The nominal slew rate is acceptable for use in the derating tables as long as the

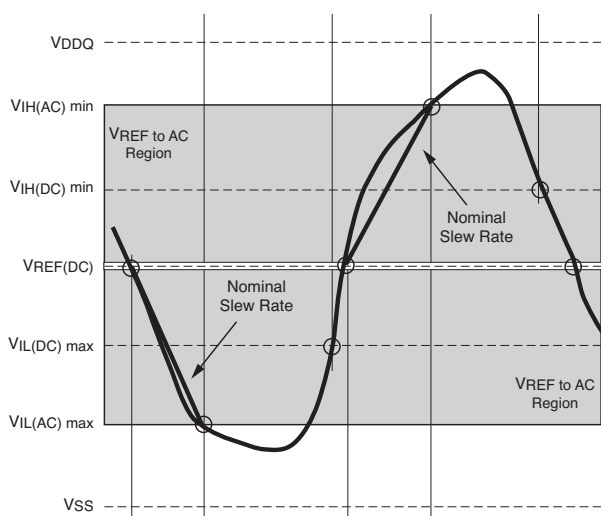


Figure 6 – The waveform illustrates how a nominal slew rate is defined for a signal when performing a derating in a setup condition. The waveform is taken from the DDR2 JEDEC specification (JESD79-2B).

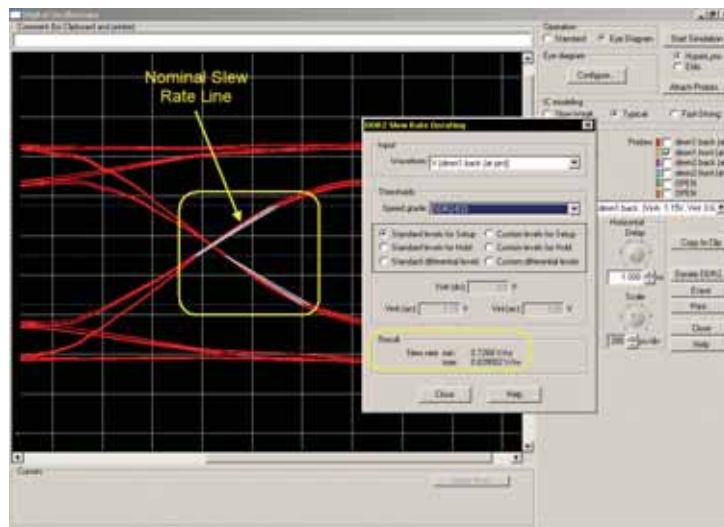


Figure 7 – The HyperLynx oscilloscope shows an automated measurement of the nominal slew rate for every edge in an eye diagram with the DDR2 slew rate derating feature. The measurement provides the minimum and maximum slew rates that can then be used in the DDR2 derating tables in the JEDEC specification.

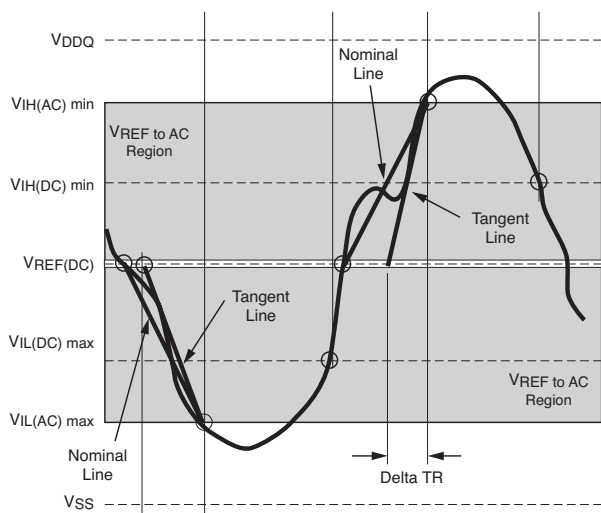


Figure 8 – This waveform, taken from the DDR2 JEDEC specification, shows how a tangent line must be found if any of the signal crosses the nominal slew rate line. The slew rate of this tangent line would then be used in the DDR2 derating tables.

received signal meets the condition of always being above (for the rising edge) or below (for the falling edge) the nominal slew rate line for a setup condition. If the signal does not have clean edges – possibly having some non-monotonicity or “shelf”-type effect that crosses the nominal slew rate line – you must define a new slew rate. This new slew rate is a tangent line on the received waveform that intersects with V_{inhAC} and the received waveform, as shown in Figure 8. The slew rate

of this new tangent line now becomes your slew rate for signal derating.

You can see in the example that if there is an aberration on the signal edge that would require you to find this new tangent line slew rate, HyperLynx automatically performs this check for you. If necessary, the oscilloscope creates the tangent line, which becomes part of the minimum and maximum slew rate results. As Figure 9 shows, the HyperLynx oscilloscope also displays all of the tangent lines,

making it easier to identify whether this condition is occurring.

For a hold condition, you perform a slightly different measurement for the slow rate. Instead of measuring from V_{ref} to the V_{inAC} threshold, you measure from V_{inDC} to V_{ref} to determine the nominal slew rate (shown in Figure 10). The same conditions regarding the nominal slew rate line and the inspection of the signal to determine the necessity for a tangent line for a new slew rate hold true here as well.

Conclusion

With the new addition of ODT, you’ve seen how dynamic on-chip termination can vastly improve signal quality. Performing signal derating per the DDR2 SDRAM specification has also shown that you can add as much as 1.42 ns back into your timing budget, giving you more flexibility in your PCB design and providing you with a better understanding of system timing.

Equipped with the right tools and an understanding of underlying technology, you will be able to move your designs from DDR to DDR2 in a reasonably pain-free process – realizing the added performance benefits and component-count reductions promised by DDR2. ●●

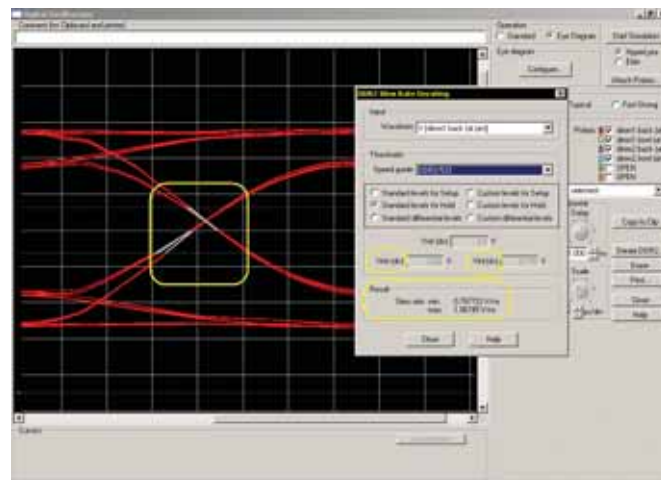
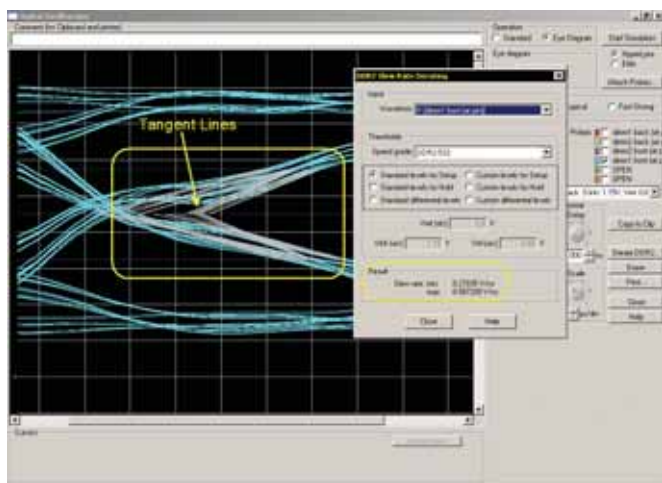


Figure 9 – The HyperLynx oscilloscope shows how the tangent line is automatically determined for you in the DDR2 slew rate derating feature. The slew rate lines in the display indicate that they are tangent lines because they no longer intersect with the received signal and V_{ref} intersection. The oscilloscope determines the slew rate of these new tangent lines for you and reports the minimum and maximum slew rates to be used in the derating tables.

Figure 10 – The oscilloscope shows how a derating for a hold condition is being performed on the received signal. The DC thresholds are used in place of the AC switching thresholds, which are noted in the DDR2 derating dialog.