



Modeling System Signal Integrity Uncertainty Considerations

Authors

Ravindra Gali

High-Speed I/O Applications
Engineering, Intel® Corporation

Zhi Wong

High-Speed I/O Applications
Engineering, Intel Corporation

Navid Azizi

Software Engineering
Intel Corporation

John Oh

High-Speed I/O Applications
Engineering, Intel Corporation

Arun VR

Memory I/O Applications Engineering,
Intel Corporation

Abstract

This white paper describes signal integrity (SI) mechanisms that cause system-level timing uncertainty and how these mechanisms are modeled in the Intel® Quartus® Prime software Timing Analyzer to achieve timing closure for external memory interface designs.

By using the Intel Quartus Prime software to achieve timing closure for external memory interfaces, a designer does not need to allocate a separate SI timing budget to account for simultaneous switching output (SSO), simultaneous switching input (SSI), intersymbol interference (ISI), and board-level crosstalk for flip-chip device families such as Stratix® IV and Arria® II FPGAs for typical user implementation of external memory interfaces following good board design practices.

Introduction

The widening performance gap between FPGAs, microprocessors, and memory devices, along with the growth of memory-intensive applications, are driving the need for faster memory technologies. This push to higher bandwidths has been accompanied by an increase in the signal count and the signaling rates of FPGAs and memory devices. In order to attain faster bandwidths, device makers continue to reduce the supply voltage.

Initially, industry-standard DIMMs operated at 5 V. However, due to improvements in DRAM storage density, the operating voltage was decreased to 3.3 V (SDR), then to 2.5V (DDR), 1.8 V (DDR2), 1.5 V (DDR3), and 1.35 V (DDR3) to allow the memory to run faster and consume less power. Plans are currently underway for DDR4 chips, which are expected to run at voltages between 1.2 V and 1.0 V.

Because of this reduction in operating voltage and timing budgets, there is a higher probability that an error may occur if the designer does not pay sufficient attention to the system design, as via breakout layers, board trace spacing, pin assignment, and power delivery network design all have a direct impact on the amount of timing uncertainty seen by the receiver.

Figure 1 illustrates the industry trend of increasing peak bandwidth while comparing various SDRAM technologies.

Table of Contents

Abstract	1
Introduction	1
Source Synchronous Timing.....	2
SI Mechanisms and Timing Uncertainties.....	3
Intel Quartus Prime Software Timing Model Assumptions	6
Conclusion.....	7
Where to Get More Information ..	7

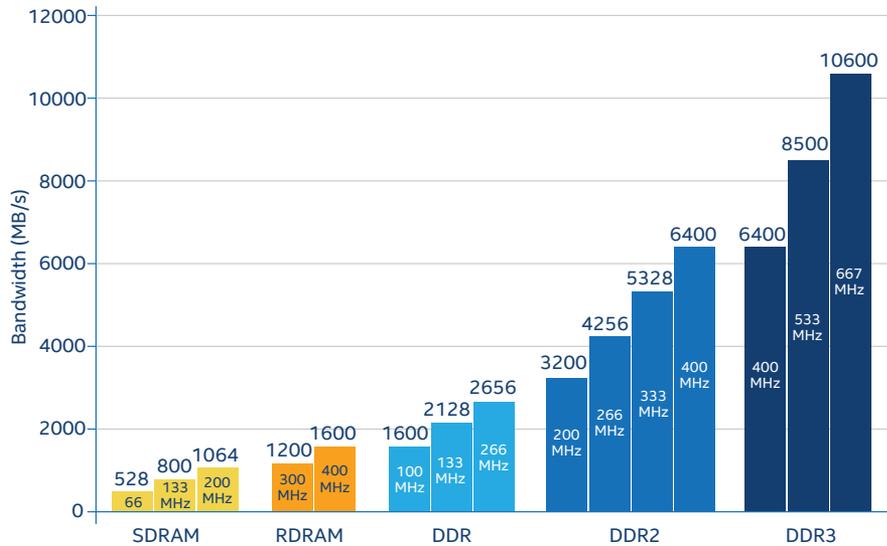


Figure 1. Peak Bandwidth Comparison of Various Memory Technologies

Source Synchronous Timing

In source synchronous interfaces, the source of the clock is the same device as the source of the data. Mainstream memory interfaces, such as DDR, DDR2, DDR3, RDRAM II, and QDR II are all source synchronous. In DDR, DDR2, and DDR3, a bidirectional clock, or data strobe (DQS/DQS#), is used for both read and write operations, while in RDRAM II and QDR II, unidirectional clocks such as DQ/DQ#/QK/QK# and K/K#/CQ/CQ# (respectively) are used. Figure 2 shows a block diagram of a basic source-synchronous interface.

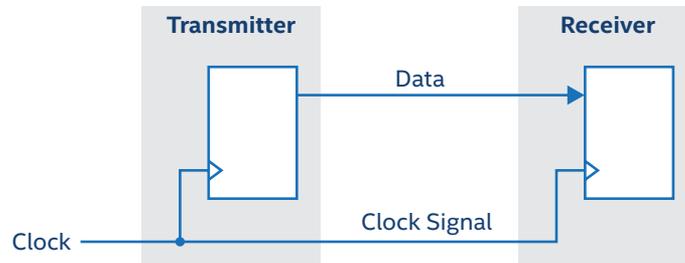


Figure 2. Source Synchronous Interfaces

In DDR applications, the data strobe is edge aligned during a read operation (a data transfer from the memory device to the FPGA) and center aligned during a write operation (a data transfer from the FPGA to the memory device). When a strobe is edge aligned with the data, the receiving device shifts the strobe as necessary to capture the data. However, in the center-aligned example, the receiving device directly uses the shifted clock to capture the data.

Figure 3 shows both edge-aligned and center-aligned data transfers.

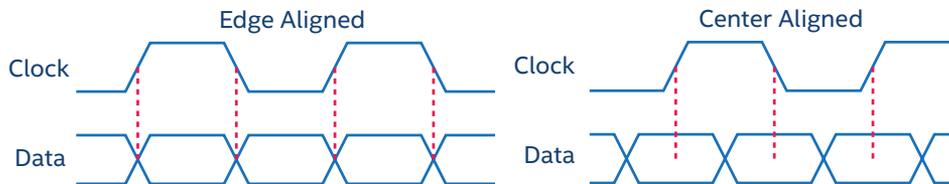


Figure 3. Edge Aligned and Center Aligned Data Transfers

In source-synchronous SDR interfaces, one edge of the clock, typically the rising edge, transfers the data. The time required to transmit one bit, known as the unit interval (UI), is equal to the period of the clock. In source-synchronous DDR interfaces, data is transferred on both edges of the clock, as shown in Figure 4. The UI is equal to half the period of the clock, assuming a 50/50 duty cycle.

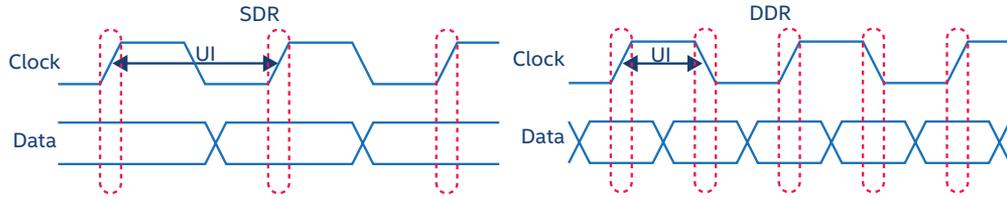


Figure 4. SDR and DDR UI Definitions

Timing margins for chip-to-chip data transfers are defined by Equation 1:

Equation 1.

$$\langle \text{Margin} \rangle = \langle \text{Bit Period (UI)} \rangle - \langle \text{Transmitter Uncertainties} \rangle - \langle \text{Receiver Requirements} \rangle - \langle t_{\text{EXT}} \rangle$$

Where:

1. Transmitter uncertainties include the timing difference between the fastest and slowest output edges on data signals, tCO variation, clock skew, and jitter. Transmitter channel-to-channel skew (TCCS) accounts for the transmitter uncertainties.
2. The receiver requirements consist of a period of time during which the data must be valid to capture it correctly. The receiver sampling window (SW) accounts for all the receiver requirements.
3. t_{EXT} specifies the board level skew across the data and clock traces. This is the maximum board trace variation allowed between any two signal traces

SI Mechanisms and Timing Uncertainties

The amount of push-out and pull-in for a given design due to simultaneous switching noise (SSN) on the outputs and inputs (SSO and SSI) depends on the choices made during the layout of the PCB. The key parameters responsible for the SI timing uncertainty include the following:

- PCB via length
- PCB power distribution network (PDN) design
- I/O buffer drive strength and slew rate
- Board trace crosstalk
- ISI
- Voltage reference (VREF)/termination voltage (VTT) variations
- Receiver I/O termination

When describing SSN in a system, it is useful to define the following terms:

- Victim pin is the pin of interest.
- Aggressor pins are pins other than the pin of interest that are transitioning and causing noise to be injected onto the victim pin.
- SSN is a noise voltage induced onto a victim I/O pin due to the switching behavior of other aggressor I/O pins in the device. The SSN results in both voltage and timing noise on the victim signal.

Figure 5 shows the two types of timing variations caused by SSO noise. Timing push-out is caused when the victim signal is switching in the same direction as the aggressor signals (Case A and C). Timing pull-in is caused when the victim signal is switching in the opposite direction as the aggressor signals (Case B and D)

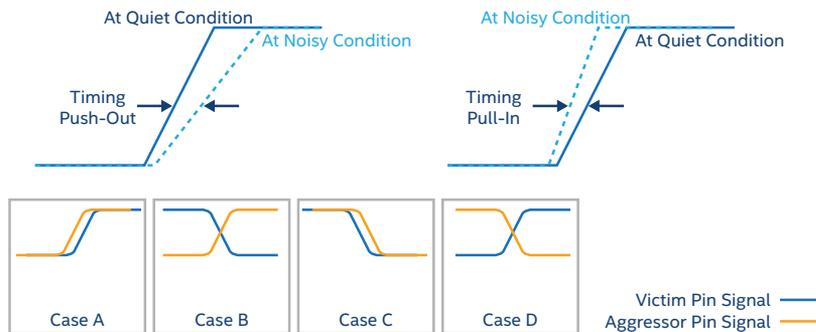


Figure 4. Timing Push-Out and Pull-In Due to SSO and SSI

The SSN seen is due to two physical mechanisms:

- Mutual inductive coupling
- Delta-I noise in the PDN

Inductive coupling is often the dominant mechanism for SSN, and is governed by Equation 2.

Equation 2.

$$V = M \times di / dt$$

Where:

1. M is mutual inductive coupling.
2. di / dt is the derivative of current over time.

Inductive coupling occurs when current from one conductor (aggressor) generates a magnetic field that is coupled to another conductor (victim) and generates a voltage across it. This effect grows with the number of switching outputs as:

$$V1 = M12 \times di2 / dt + M13 \times di3 / dt + ...$$

Therefore, the larger the number of simultaneously switching buffers, the larger the SSN due to mutual inductance.

Most inductive crosstalk occurs in the vertical structures rather than in the horizontal transmission line structures. Examples of vertical coupling structures include C4 solder bumps, package vias, solder balls (package pins), PCB vias, and pins in a DIMM connector. The magnitude of inductive coupling is proportional to the parallel length of the aggressor and victim signals. All vertical structures contribute some amount of inductive coupling. However, most of the coupling occurs at the interface between the FPGA package and the PCB in the PCB break-out via field, where the parallel path is the longest between aggressors and victims.

The other dominant source of inductive coupling is the via field region under the DIMM or the discreet memory device. Noise is inductively coupled from the aggressor to the victim conductors during the aggressor rise and fall time and is not coupled at any other time.

The value of the mutual inductance, M, which affects the amount to which the different vias are coupled, is a function of the self inductance (length), L, of each via and the coupling, k, between the vias. The coupling is, among other things, a function of the distance between the vias, thus causing vias that are closer together to have a larger mutual inductance between them. The designer must pay attention to the via break out during layout to minimize the amount of coupling.

Figure 6 shows the important components of the various coupling mechanisms on a memory system topology.

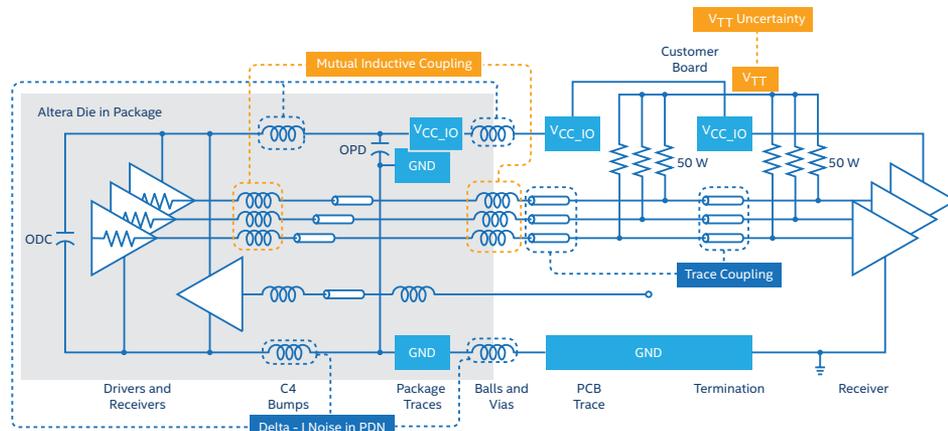


Figure 6. Noise Coupling Mechanisms

Delta-I noise in the PDN is caused when multiple output drivers switch simultaneously and induce voltage changes in the chip and package PDN. This noise manifests as a voltage drop on the power rail and a voltage spike on local GND relative to the system GND. These changes in voltage are related to the amount of loop inductance present in the PDN and the amount of current sunk by each switching output, determined by Equation 3.

Equation 3.

$$V = L \times di / dt$$

Loop inductance in the PDN is comprised of the inductance of the on-chip PDN, the inductance associated with the package plane, vias and balls, the inductance associated with the PWR and GND vias in the PCB breakout region, and the loop inductance of the PCB planes. The larger the inductance in the PDN, the larger the change in voltage. Furthermore, the larger the number of outputs switching at the same time, the larger the value of di/dt and therefore, a larger value of PDN noise.

Similar to signal vias, the longer the lengths of the PWR and GND vias, the higher the PCB loop's inductance contribution to the overall PDN inductance.

In addition, and similar to inductive coupling, delta-I noise only occurs during the signal transition, as this is the only time where the current changes as a function of time. Delta-I noise does not occur in time frames where the driver current is constant because there is no di/dt to generate the noise.

The di/dt of a switching I/O depends on the I/O buffer's drive strength and the slew rate setting enabled by the buffer. Stratix IV and Arria II FPGAs offer a variety of drive strengths for each supported I/O standard. The I/O buffer drive strength of a given driver is a measurement of how much current the driver launches on a given load. It can also determine the largest load that can be driven at a certain speed, without affecting the integrity of the transmitted signal. In other words, a stronger driver is able to drive larger loads and longer transmission lines.

However, it is not always a good idea to simply choose the strongest driver because it is able to drive larger loads and longer transmission lines. Stronger drivers launch larger currents, and larger currents imply larger crosstalk, timing pull-out and pull-in due to SSN, and power consumption. A stronger driver might provide a larger noise margin but also generates a larger noise that impacts timing. Because choosing the right driver directly affects the quality of the signal, it is important to choose the minimum drive strength able to drive the load connected to the output of the FPGA.

Figure 7 shows the drive strength effects on the output signal when using a transistor-to-transistor logic (TTL) standard that toggles from rail to rail. SSTL and HSTL I/O standards behave differently because of the presence of pull-up resistors.

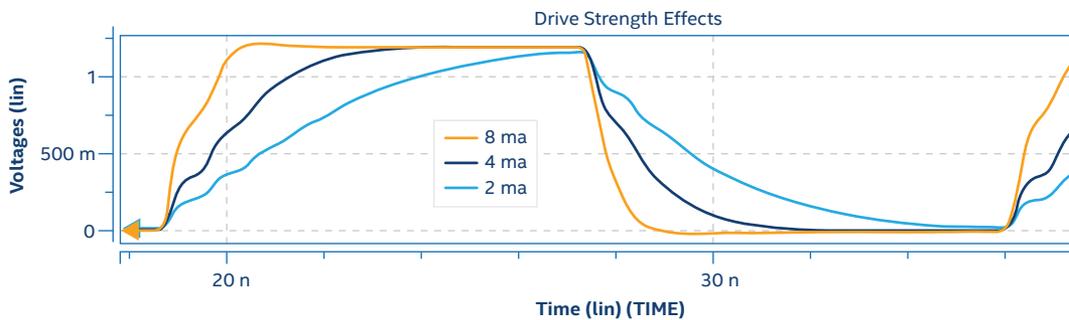


Figure 7. I/O Drive Strength Impact on the Output Signal

The I/O buffer slew rate determines the maximum rate of change of the output signal. In other words, it determines the speed of the rising and falling times of the output signal. Stratix IV and Arria II FPGAs have different slew rate settings that allow the designer to modify the duration of the rise and fall times.

The drive strength specifies how much current the driver sources and sinks; the slew rate specifies how fast the buffer sources and sinks the current. Together, these two settings determine the rise and fall times of the output signal. The rise and fall times are set by the process technology.

Figure 8 shows the rising edge of the output signal under four different settings. The designer can choose the one that is optimal for the design based on the timing noise trade off.

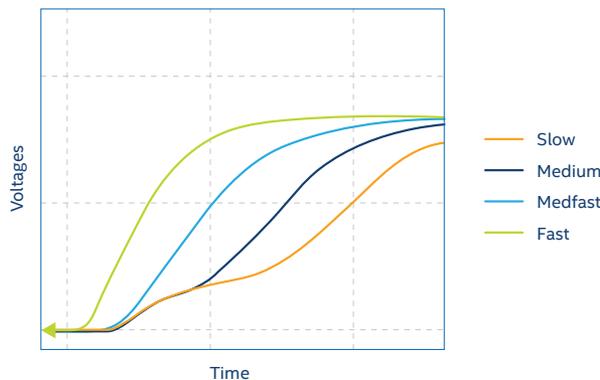


Figure 8. I/O Slew Rate Impact on the Output Signal

Trace-to-trace coupling can result in board-level crosstalk, causing a timing pull-in or push-out on the victim signal. The crosstalk results in a change in the effective characteristic impedance and the propagation velocity of the trace. Additionally, it can induce noise voltage onto the victim trace. The amount of crosstalk seen on the victim trace depends on the number of toggling aggressors, the aggressor data pattern, the air gap separation between the victim and aggressor traces, and the toggling rate of the aggressor signals.

Trace-to-trace coupling is caused by board real estate constraints when fanning out traces to the routing layers underneath the PCB via breakout region. After breakout, the air gap between the traces should be increased to minimize coupling. A good rule is to have a $3H$ air gap between the traces, where H is the dielectric height between the trace and the nearest GND plane. Minimize H so that the trace couples strongly to the GND reference plane and less to the adjacent signals. During layout, route with short parallel sections and minimize long coupled sections between nets.

The traces on a PCB are bandwidth limited and behave like a low-pass filter. The low-pass filtering smears the transmitted signal, over time causing the effect of a bit period (UI) to spread across the adjacent bit periods when a sequence of data bits is transmitted (ISI).

ISI is pattern dependent and can result in a timing uncertainty known as pattern-dependent jitter or data-dependent jitter. The skin effect of a conductor and the dielectric loss is responsible for ISI. Reflections from poorly terminated loads can also be a source of ISI. As frequency increases, dielectric loss is the dominant factor in high-frequency attenuation because its effect is proportional to the frequency, where the skin effect is proportional to the square root of frequency.

All PCB laminate materials have a specific dielectric constant and a loss tangent value. Materials with a high loss tangent often see a deterioration of the signal with frequency. Low-cost materials such as FR-4 have a high loss tangent, which results in a large attenuation of the signal at high frequency. To minimize ISI, design the PCB using a dielectric material with a lower loss tangent value based on the application requirements. Dielectric materials with a lower loss tangent cost more than materials with higher loss tangent.

Timing uncertainty is also caused by noise on the VREF or VTT power rail, offset of the VTT relative to the VREF, drift of VREF or VTT over voltage and temperature, and an external component mismatch. Stratix IV and Arria II FPGAs have calibration circuits to ensure that the strobe signal stays in the center of the data valid window by calibrating for voltage (V) and temperature (T) over time.

The choice of receiver I/O termination can also result in system uncertainty because non-optimal receiver termination may result in the signal being reflected back and forth onto the transmission line, which can cause degradation in the signal edge rate seen at the receiver. Choose the optimal on-die termination (ODT) value based on the characteristic impedance of the traces on the PCB.

Intel Quartus Prime Software Timing Model Assumptions

The Intel Quartus Prime software accounts for the timing uncertainty from many of the SI mechanisms when analyzing timing for external memory interfaces. This feature in the timing model applies to designs using Stratix IV and Arria II FPGAs that use flip-chip technology for the package in the Intel Quartus Prime software.

For these families, the timing model assigns a timing uncertainty parameter due to SSO and SSI based on mechanisms that can influence timing push-out and pull-in. The timing model makes certain assumptions for PCB via length, PDN design, I/O buffer drive strength and slew rate, board trace crosstalk, ISI, VREF/VTT variations, and receiver I/O termination to reflect a typical memory interface application for the analysis. The timing uncertainty values are based on simulations and system-level characterization for the assumed parameters.

On a typical mainstream memory interface, a data signal strobe is associated with a number of data bits, usually eight, but can vary from four to 36 bits. When the FPGA writes to the memory device, time uncertainties include contributions from the numerous internal FPGA circuits including the following:

- Location of the DQ and DQS output pins
- Width of the DQ group
- PLL clock uncertainties, including phase jitter between different output taps used to center-align the DQS with respect to the DQ pins
- Clock skew across the DQ output pins and between the DQ and DQS output pins
- Package skew on the DQ and DQS output pins
- Push-out and pull-in on the output pins due to multiple DQ and DQs pins switching simultaneously at the same time (SSO)

Conclusion

Though the Intel Quartus Prime software takes into account the timing uncertainty due to various SI effects, such as SSO, SSI, ISI, and crosstalk, for both read and write paths, the amount of uncertainty that the Intel Quartus Prime software assumes is based on a typical user implementation for external memory interfaces following good board design practices. Any variations, such as designing the PCB with very deep signal vias, very deep power and GND vias, minimal trace-to-trace spacing, and using a high-loss tangent dielectric material for board design, lead to a higher amount of uncertainty.

In situations where a PCB design may deviate significantly from best practices and the typical application assumed in the Intel Quartus Prime software timing model, Intel recommends that designers complete further analysis in simulation using the appropriate package, PCB, and I/O models. In most cases, the assumptions and techniques the Intel Quartus Prime software timing model uses for timing closure for external memory interfaces lead to an accurate assessment of the interface performance.

Where to Get More Information

- Signal Integrity Center: www.altera.com/support/support-resources/support-centers/signal-power-integrity.htm

