

Data Strobe Timing of DDR2 using a Statistical Random Sampling Technique

Rashed Zafar Bhatti

EE- Systems Department
University of Southern California
Marina del Rey, CA-90292, USA
bhatti@isi.edu

Monty Denneau

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
denneau@us.ibm.com

Jeff Draper

Information Sciences Institute
University of Southern California
Marina del Rey, CA-90292, USA
draper@isi.edu

Abstract— This paper presents a new way to tackle critical bus cycle timing issues related to DDR/DDR2 bus operations using a statistical random sampling technique. The technique allows a pure standard cell based design which is inherently area, power and design time efficient compared to existing solutions proposed in the literature. The proposed design employs a statistical random sampling technique to measure and correct the duty cycle of a clock to produce source synchronous signals and to adjust the phase of the incoming strobe to correctly capture data. The proposed circuits are used to interface Samsung K4T51163QB_D5 DDR2 chips to a massively parallel processing logic ASIC chip, targeted to IBM Cu-08 90 nm technology. The proposed design is a fully digital solution based on standard cell components and does not require any custom designed component. This makes it extremely design time efficient and portable across most ASIC and FPGA technologies.

I. INTRODUCTION

RAMBUS [4] and Double Data Rate (DDR) [5] memory technologies were introduced to achieve high data throughputs [6] and reduce the ever increasing performance gap between digital logic and memory subsystems. These memory technologies utilize source-synchronous double data rate techniques to achieve higher data bandwidth. In DDR/DDR2 a group of data bits is sent over parallel wires along with a strobe. The timing of the strobe with respect to the data is very critical and differs for read and write bus operations. Fig. 1 shows a typical read and write timing of DDR2 with a read latency (RL) of 5 cycles and burst length of four. Notice that during the write operation the edges of the strobe have to reach the memory centered within the burst of data bits to maximize jitter tolerance and timing margin. Whereas during read cycles, edges of both data and strobe are launched from the memory chip simultaneously, thus leaving the responsibility of adjusting the phase of the strobe to capture the data correctly to the receiving logic. Besides this synchronization issue, when writing to DDR/DDR2 it is important to have the timing of individual data bits and strobe equalized to let the DRAM correctly capture the data[5]. Achieving this interface characteristic

requires either a clock of double the frequency that of the DDR/DDR2 or a clock with balanced duty cycle (50%) on the transmitting side. Conventional DDR controllers employ PLLs and DLLs to correct the phase of the incoming strobe relative to data and launch the data and strobe with balanced timing. These circuits involve analogue or mixed signal components which are inherently resource heavy in terms of silicon area and power dissipation. Most of the previously proposed all digital solutions [11][12] either produce less accurate timing compensation or use some custom designed digital components, which leaves the design technologic specific or platform dependent. Moreover most of these approaches have heavy patent restrictions on them which prohibit low-budget ventures from employing these techniques. Compared to these approaches, the proposed design employs a pure standard cell ASIC design approach which keeps the overall system area, power and design-time efficient and portable across varying technologies. The proposed design employs a statistical random sampling technique to measure and correct the duty cycle of clock [1] to produce source synchronous signals and to adjust the relative phase [2] of the incoming strobe to correctly capture the data. To better understand the problem, Section II provides a quick review of source synchronous I/O systems and optimization equations. Section III explains the ideological and theoretical basis of the proposed design. Section IV provides an insight of circuit level implementation. Section V briefly analyzes the proposed design and section VI concludes the paper.

II. BACKGROUND OF SOURCE SYNCHRONOUS SYSTEM

Source-synchronous signaling used in RAMBUS and

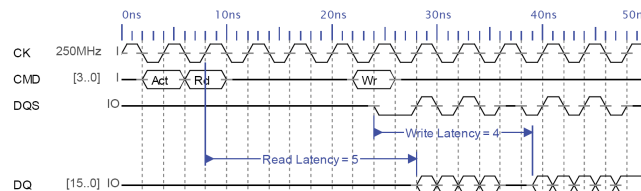


Figure 1. A typical read and write cycle timing of DDR2.

DDR/DDR2 memory systems is a standard technique for high-speed parallel bus interfaces in digital systems [8]. A typical source synchronous channel employs a PLL on the transmission side to provide a balanced and stable synchronization clock to launch the data and strobe. In contrast to conventional common clock signaling [13], a source synchronous bus provides a sampling strobe in synchronization with the data. A separate channel carries a reference strobe, whose phase is adjusted at the receiver by a delay locked loop (DLL) to sample at the middle of the data eye. Compared to conventional common clock signaling [13], in this technique absolute signal propagating delays (flight times) are omitted from the timing equations because both data and strobe are sourced from the same transmitter, and a carefully designed printed circuit board equalizes the propagation delays. All delay terms are converted to differential delays which are represented relative to the sampling edge of the strobe. Fig. 2 shows the setup and hold timing of a typical source synchronous bus. The basic source synchronous bus timing optimization equations [6][7] for DDR/DDR2 in which data is transmitted at both edges of the strobe are given as follows:

$$Tvb_{\min} > Tsu_{\text{receiver}} + (t_{f\text{-data}} - t_{f\text{-strobe}})_{\max} \quad (1)$$

$$Tva_{\min} > Th_{\text{receiver}} + (t_{f\text{-data}} - t_{f\text{-strobe}})_{\min} \quad (2)$$

$$(Tvb + Tva) < 0.5T_{\text{cycle}} \quad (3)$$

Where Tvb and Tva are the minimum time the signal is required to be valid at the receiving components before and after the sampling edge of the strobe respectively. Tsu and Th represent the setup and hold time respectively. Times of flight or propagation delays of data and strobe are represented by t_f . The difference term in the above equations comes from the timing uncertainties, and it has a dynamic and a static component [7]. The static component comes from the mismatched parameters like impedance and length of the two channels; it is usually called skew. Conventionally it is compensated with a DLL on the receiving end. The main sources of the dynamic part of timing uncertainties are signal jitter, crosstalk, ambient noise and intersymbol interference (ISI) [9]. The usual approach for tolerating these effects is to include adequate timing margin, provided all other techniques to minimize these effects are already employed. It is obvious from (3) that positioning the edges of the strobe within the eye of the data and balanced duty cycle of the strobe is very critical to optimize data rate and timing margin of a source synchronous signaling system.

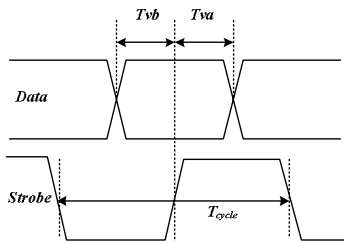


Figure 2. Setup and hold timing of a typical source synchronous bus.

III. THEORETICAL SUBSTRATUM OF PROPOSED DESIGN

Statistical random sampling has long been used to quantitatively estimate a particular attribute of a given large population through sampling some correlated observable phenomenon randomly. We applied this technique in this design over high-speed on-chip signals in two manners: (1) to measure and correct the duty cycle of the system clock to produce equalized data and strobe for write cycles of source synchronous DDR/DDR2 bus, and(2) to measure and adjust the path delay of the strobe relative to the data to align the capturing edge of the strobe in the middle of the data eye. In this technique a random clock [3] is used, which is generated by a digitally controlled ring oscillator fed with pseudo-random numbers generated from a Linear Feedback Shift Register (LFSR). The capricious behavioral characteristics of a ring oscillator together with pseudo-random numbers generated by a LFSR produce a functional random clock for random observation of on-chip signals.

A. Duty Cycle Measurement

To measure the duty cycle of the system clock, its state is repeatedly captured and recorded at random instants of time with the help of a random clock [3]. The duty cycle is directly related to the probability of capturing a logic high (one) in a particular random observation. A large data sample of premeditated size is gathered and the ratio of the number of ones in the sample corresponds to the duty cycle of the clock under measurement. The following section shows how this measurement technique is used to correct the duty cycle of the system clock. The accuracy and confidence level can easily be controlled with the size of the collected sample [1].

B. Relative Phase Measurement

The measurement of the relative phase of two periodic signals with the statistical random sampling technique requires simultaneous observations of the two signals at random instants of time. For this, the states of the two signals are captured with a random clock and if the leading signal is captured as high and lagging signal is captured as low the observation is counted. If the cycle time of the two signals under observation is T_{cycle} and t_A is the time for which two signals overlap such that the leading signal is high and the lagging signal is low, then for a large enough sample size the ratio of number of counted observations to the total observations becomes quite equal to the ratio t_A/T_{cycle} . This is due to the fact that the joint probability of capturing the two signals in the particular state (when first is high and second is low) is equal to the ratio t_A/T_{cycle} . The observed ratio can be mapped to relative phase using the equation: phase $\varphi = 2\pi t_A/T_{\text{cycle}}$. Similar to the duty cycle measurement, the accuracy and confidence level of the relative phase measurement can easily be controlled with the sample size.

IV. CIRCUIT LEVEL IMPLEMENTATION

As per the JEDEC standard [5] DDR2 SDRAMs use a differential clock (CK) input to latch the address and command signals. The data (DQ) and strobe (DQS) are

bidirectional busses which are required to be turned around to switch between read and write cycles. During write cycles the controller is expected to send the data with edges of the strobe centered within the data. Conventional DDR controllers employ special DLL based circuits which tightly control this timing. To avoid resource heavy, specialized DLLs and keep the design purely standard cell based, the proposed design employs a clock double the frequency of CK with a balanced duty cycle of 50%. The data is launched at the positive edges of this clock and transitions of the strobe are launched at the negative edges. In this way the edges of the source synchronous strobe automatically remain aligned within the center of the data.

A. Local Duty Cycle Correction of the Clock

In the proposed approach explained above, local correction of the duty cycle is a timing problem to be addressed, because in practice a noticeable degradation in duty cycle can be observed at the terminal ends of the clock tree, even if the clock is generated with a perfectly stable and accurate oscillator or PLL. This phenomenon occurs due to the slight mismatch in the drive strengths of pull-up and pull-down networks of the CMOS gates/buffers and non-uniformity in the distribution of wiring capacitances. Thus, a local duty cycle correction circuit is required to fix this problem. Fig. 3 shows the proposed circuit of duty cycle corrector (DCC). To locally correct duty cycle of the clock signal, the proposed circuit delays the input clock with a digitally controlled delay line, the delayed and original input are ORed or ANDed to stretch or chop the asymmetric clock signal respectively, to produce a balanced output duty cycle as shown in Fig. 4. The random sampling unit (RSU) [1] provides an accurate measurement of duty cycle of the output of the DCC which is fed back to adjust the delay line.

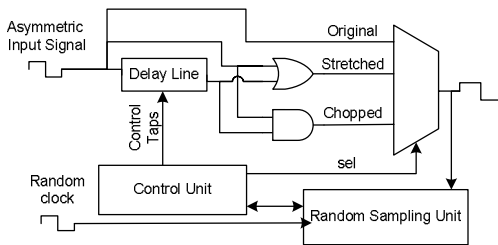


Figure 3. Block diagram of DCC circuit.

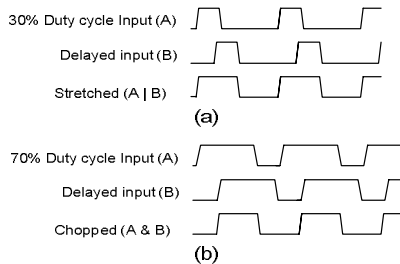


Figure 4. Duty Cycle Correction.

B. Strobe Delay Adjustment

The second critical timing issue in the circuit level implementation of a DDR2 interface appears during the read cycle. Since DDR2 provides simultaneous transitions of the data and strobe [5], the path delay of the strobe has to be adjusted to maximize the timing margin and correctly capture the data. Similar to the duty cycle problem, to avoid DLLs this problem is also handled using the statistical random sampling technique [2]. Fig. 5 shows both data and strobe are received through well-matched channels. The path delay of the strobe is adjusted using a digitally controlled delay-line to have strobe edges in the middle of the data. To accurately setup the delay line, a balanced clock is fed to the delay line using a multiplexer. The original and delayed signals are fed to an RSU [2] to measure their relative timing by simultaneously capturing the state of the two signals at random instants of time. As shown in the Fig. 6 the required phase delay corresponds to the region “A”. The joint probability of capturing the two signals with region code 10 directly corresponds to the phase delay to be measured and corrected. Using this technique the proposed design can be targeted to any standard cell technology without using any resource heavy component like DLLs or PLLs for data strobe timing and alignment.

V. ANALYSIS OF THE PROPOSED DESIGN

The DDR/DDR2 bus interface design described in this paper is a unique design as compared to conventional designs in many ways. Timing skew is compensated during the configuration stage at boot time; timing margins are maximized to tolerate dynamic timing uncertainties like jitter, crosstalk etc. In this section we analyze different aspects of the design.

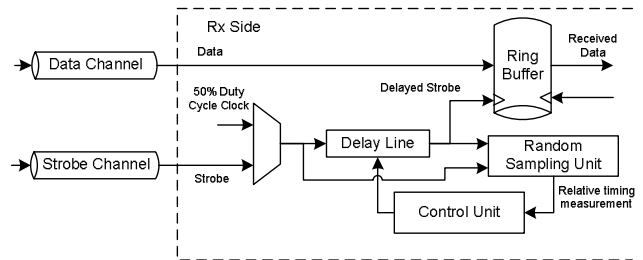


Figure 5. DDR/DDR2 strobe delay adjustment circuit.

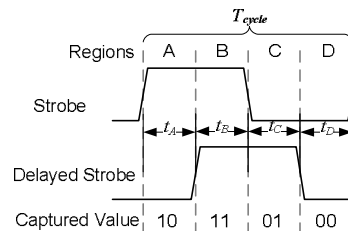


Figure 6. Delayed strobe timing.

A. Design Time efficiency

The proposed design is built with well-characterized library components instead of custom designed components that may involve a long and tedious characterization phase before they can be used in actual systems. The pure standard cell based ASIC design flow makes the overall design and verification time much shorter. The design time efficiency of the approach reduces the time to market of the final product.

B. Portability of Design

The proposed DDR/DDR2 interface design is strictly a standard cell based design that does not require any custom design component. This characteristic makes this design practically portable to any standard cell ASIC or FPGA technology. The requirement of resource heavy components like a PLL or DLL for frequency synthesis and data/strobe alignment is negated by using a standard cell based RSU[1][2] that provides comparable performance when used with an appropriate sample size. The trade-off here is the configuration time at startup of the interface. If the system has to run for a considerable time before it is restarted, then the setup time of a few seconds becomes insignificant. This startup configuration time can be significantly reduced by storing the configuration corresponding to the native PCB and socket in some non-volatile memory and loading this information into configuration registers as a part of subsequent boot-up sequence.

C. Flexibility of Bus Timing Specification

The strobe can easily be placed to any specific time within the data, which means if the setup and hold time of the bus protocol are changed because of some technical reason the proposed interface can adapt to the new timing specifications. Moreover the interface can easily adapt to any potential future changes in the SDRAM interface timing standard.

D. Implementation and Timing Verification

The proposed technique is employed to interface Samsung K4T51163QB_D5 DDR2 chips to a massively parallel processing logic ASIC chip, targeted to IBM Cu-08 (90 nm technology). To handle signal integrity issues 1.8V SSTLs (stub series-terminated logic) of the IBM Cu-08 IO library are used. For functional and timing verification a very accurate simulation model by Denali Inc. [10] is employed. Extensive simulations were run to prove the effectiveness of the proposed technique. The measurement and correction results obtained manifested a very close consonance with the expected theoretically result. The RSU based delay loop is an order of magnitude smaller in terms of silicon area compared to typical double data rate delay lines for DDR/DDR2 like DDRDL_9FS of IBM Cu-08 (90 nm technology).

VI. CONCLUSION

This paper addressed a very important issue related to DDR/DDR2 bus timing in a unique way by employing statistical random sampling. The proposed design technique negates a requirement of DLLs or PLLs to control the timing of the source synchronous DDR/DDR2 bus. This approach considerably reduces the silicon area of the overall design. The proposed design not only reduces circuit complexities but also reduces the perpetually consumed power by power-hungry components like PLLs or DLLs used in conventional DDR/DDR2 controllers. The proposed design is a fully digital solution based on standard cells and does not require any custom designed component. This makes it extremely design time efficient and portable across most ASIC and FPGA technologies.

REFERENCES

- [1] R. Bhatti, M. Denneau, J. Draper "Duty Cycle Measurement and Correction Using a Random Sampling Technique", Proceedings of the 48th IEEE Midwest Symposium on Circuits and Systems, August 2005.
- [2] R. Bhatti, M. Denneau, J. Draper "Phase Measurement and Adjustment of Digital Signals Using Random Sampling Technique", Proceedings of the IEEE International Symposium on Circuits and Systems, May 2006.
- [3] R. Bhatti, K. Chugg, J. Draper "Standard Cell based Pseudo-Random Clock Generator for Statistical Random Sampling of Digital Signals", Proceedings of the 50th IEEE Midwest Symposium on Circuits and Systems, August 2007.
- [4] C. C. Huang, K. S. Oh, S. Rajan, "The Interconnect Design and Analysis of RAMBUS Memory Channel," Proceedings of ASME IPACK 2001, IPACK2001-15531, July 2001.
- [5] JEDEC Standard DDR2 SDRAM Specification, JESD79-2C (Revision of ESD79-2B) MAY 2006.
- [6] T. Arabi; J. Jones, G. Taylor; D. Riendeau, "Modeling, simulation, and design methodology of the interconnect and packaging of an ultra-high speed source synchronous bus"; IEEE 7th topical Meeting on Electrical Performance of Electronic Packaging, Oct. 1998
- [7] N. Pham; M. Cases; J. Bandyopadhyay; "Design, modeling and simulation methodology for source synchronous DDR memory subsystems"; Electronic Components and Technology Conference May 2000
- [8] G. Balamurugan; N. Shanbhag; "Modeling and mitigation of jitter in multiGbps source-synchronous I/O links"; Proceedings of International Conference on Computer Design, Oct. 2003
- [9] A. Wirick, S. Ulrich, N. Pham, M. Cases, D. N. de Araujo, "Design and modeling challenges for DDR II memory subsystems", Electrical Performance of Electronic Packaging, 2003, Page(s):229 – 232
- [10] Denali Software, Inc. <http://www.denali.com>.
- [11] Ching-Che Chung; Pao-Lung Chen; Chen-Yi Lee; "An All-Digital Delay-Locked Loop for DDR SDRAM Controller Applications"; International Symposium on VLSI Design, Automation and Test, April 2006
- [12] You-Jen Wang; Shao-Ku Kao; Shen-Iuan Liu; "All-digital delay-locked loop/pulsewidth-control loop with adjustable duty cycles"; IEEE Journal of Solid-State Circuits, Volume 41, Issue 6, June 2006
- [13] W.J. Dally and J.W. Poulton, "Digital Systems Engineering", Cambridge University Press, 1998.