27.5 Phase-Tolerant Latency Control for a Combination 512Mb 2.0Gb/s/pin GDDR3 and 2.5Gb/s/pin GDDR4 SDRAM

Brian Johnson[1], Brent Keeth[1], Feng Lin[1], Hua Zheng[2]

[1]Micron Technology, Boise, ID, [2]Micron Technology, Shanghai, China

High-speed DRAM interface standards create significant design challenges, especially in the multi-GHz realm. As unit intervals shrink, read latency control becomes one of the more difficult problems to solve. High data rate SDRAM specifications require the use of DLL circuits to generate read clocks, which, in turn, creates an indeterminate and time-varying phase boundary between the command/address (C/A) clock and the read data clock domains. This clock domain crossing must be managed to guarantee that read latency is determinate under all conditions. A similar latency problem exists for write data because of the arbitrary phase relationship between write data capture circuits and the command decoder circuit. In this paper, circuit techniques are described to manage both read and write latency in a high data rate GDDR3/GDDR4 capable device. These techniques are incorporated in a 512Mb SDRAM operating up to 2Gb/s/pin in GDDR3 mode and up to 2.5Gb/s/pin in GDDR4 mode.

Figure 27.5.1 is a simplified block diagram of the circuits involved in latency control. The input delay, denoted by tI, represents the delay-matched distribution of the input C/A signals and system clock distribution to the primary capture latches. The output delay (tO) represents the output path delay from the data retiming latches to the output pad. The timing diagram shown in Fig. 27.5.2 depicts the methodology developed for the GDDR3/4 device to achieve deterministic read timing specified by the programmed CAS latency (CL) value.

The pseudo-differential signaling employed on a GDDR3/4 device necessitates additional gain stages following the input receivers. Slew-rate control circuits [1] are also needed in the predriver circuits. These circuit features increase maximum delay and delay variation, causing the DLL I/O model delay to be much greater than the system clock period. The total phase difference between the system clock and read clock is expressed as, $\Phi c = 2\pi(tI + tO)/tCK$ rad.

Figure 27.5.2 shows the simulated range of delay values for total I/O delay. Because the phase difference between the read clock and the system clock spans such a large range, it becomes difficult to pick the correct corresponding clock edge to enable the output data path so that deterministic read latency is achieved. The read latency control circuits in Fig. 27.5.1 are implemented to track phase variations between the read clock and system clock and determine the correct corresponding clock edge from which to enable the output path relative to a decoded read command. For a combination GDDR3/4 device, a wide range of CL is necessary because of the frequency range over which the device operates. This method differs from a previous implementation [2] in that a larger range of programmable CL values is supported without additional circuitry. Other advantages include: (1) predictive alignment of output data to correct clock phase for a multi-clock-phase data output path, (2) elimination of read clock routing near the C/A decode circuits, (3) a reduced risk of replica delay mismatch and timing failure since a true I/O replica delay is used only once during initialization. Figure 27.5.3 shows the read latency control circuits used to accomplish correct read latency timing. These circuits control the tICL delay shown in Fig. 27.5.4.

Upon DLL_LOCK going high, a reset signal is forwarded from the read clock domain through a replica DLL I/O model to a gray-code counter synchronized to the C/A clock domain. The reset signal emerges from the delay model correctly aligned to the C/A clock because of DLL phase detector (PD) action. For an imaginary CL=0 case, the read clock domain gray code counter would actually start prior to the counter in the C/A clock domain resulting in the read clock domain counter leading the C/A clock driven counter by tI+tO.

For the CL=12 case shown in Fig. 27.5.4, there is one stage of pipeline overhead (SP). The initialization FSM block in Fig. 27.5.3, synchronizes the reset signal sent to the C/A clock domain and then waits CL-SP read clock cycles before releasing the reset to the read clock domain counter. The result of this initialization is that corresponding count values generated in the read clock domain, relative to the C/A clock domain, are delayed by tICL, which is expressed as tICL=(CL-SP)*tCK-(tI+tO).

Subsequent to the initialization procedure outlined above, the two gray code counters shown in Fig. 27.5.3 provide an index for corresponding clock edges by establishing the discrete, programmed portion of tICL ((CL-SP)*tCK) while tracking phase variation (-(tI+tO)) between the C/A and read clock domains.

When a read command is decoded, the count value from the C/A clock domain is written into a FIFO. The output latency (tFWL) of the FIFO must be less than tICL. The FIFO depth (FD) is approximated as a function of column cycle delay (CCD), expressed in system clock cycles, and the maximum allowable CL, such that FD = tICLmax/(tCK*CCD) rounded up to the next integer value. For the GDDR3 case, CCD = 2 with maximum tICL occurring at CL = 12, tI + tO = 1.6ns and tCK = 1ns. Note that tICLmax occurs at the fastest operating corner. Applying the equation results in FD = 5.

The completion of a read cycle assumes that column address access delay (tCAA in Fig. 27.5.2) occurs within the timing boundary established by the corresponding clock edge in the read clock domain. Synchronizing the output enable across four byte groups requires careful interconnect design for both clocks and enable signals. Clock frequency division at the expense of logic complexity was implemented to solve these timing issues.

Write latency control is a simple problem compared to read latency control because the phase difference between system clock and write strobe (WDQS) is specified at a maximum of ±1/4 clock period. A more difficult problem for write latency control is the distribution of a common write timing signal shown in Fig. 27.5.5. The GDDR3/4 device has physically large separations between byte groups making it difficult to distribute timing signals common to all I/O, such as the write timing signal. The result is that interconnect delay can easily exceed one clock period. To improve timing margin for write latency prediction, a DLL implicitly locks the continuous C/A clock to the burst WDQS clock through input delay matching and clock tree distribution modeling. First, the DLL-generated clock aligns with the C/A clock so that the write timing signal is easily transferred from the C/A clock domain to the DLL clock domain. The DLL output clock can be frequency divided to allow more timing margin for signal distribution. The write timing signal aligns to the captured data and retires the correct data to the array.

Using these read and write latency techniques, an 88mm^2 512Mb×32 GDDR3/GDDR4 device, as shown in the die micrograph, is fabricated in a 1.5V 3M 95nm CMOS process. Figure 27.5.6 is a measured tCK-Vdd shmoo data showing device performance reaching beyond 2Gbps/pin in GDDR3 operation for page fast write and read vectors that produce 2ns column cycle times (data failures).

References:

[1] T. Matano, Y. Takai, T. Takahashi, et al., "A 1-Gb/s/pin 512-Mb DDRII SDRAM Using A Digital DLL and A Slew-Rate-Controlled Output Buffer," *IEEE J. Solid-State Circuits*, pp. 762-768, vol. 38, no. 5, May, 2003.

[2] S.-B. Lee, S.-J. Jang, J.-S. Kwak, et al., "A 1.6-Gb/s/pin Double Data Rate SDRAM with Wave-Pipelined CAS Latency Control," *IEEE J. Solid-State Circuits,* pp. 223-232, vol. 40, no. 1, Jan., 2005.

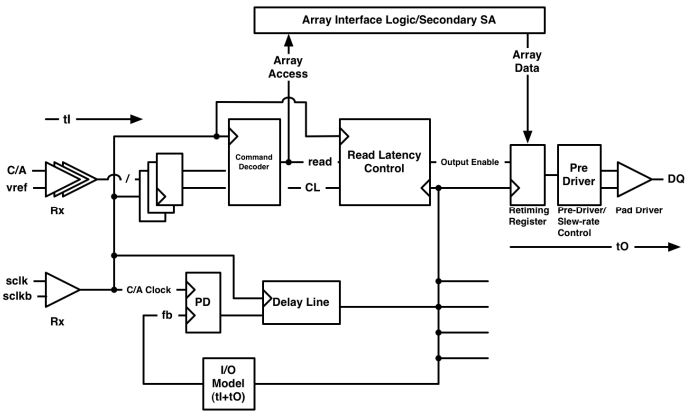

Figure 27.5.1: Read latency control context.

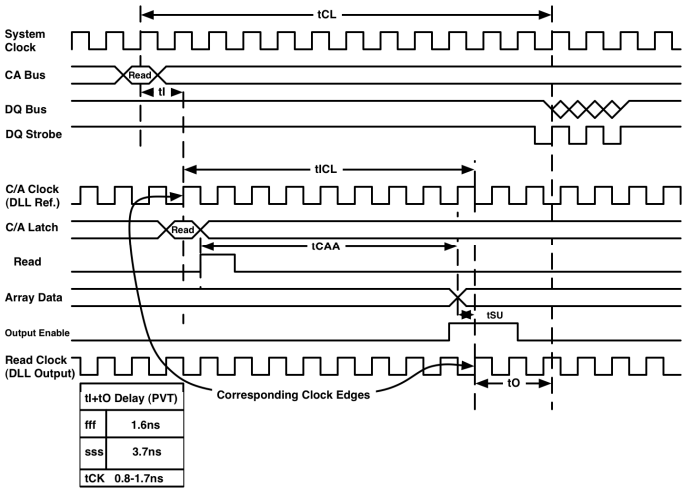

Figure 27.5.2: Read latency timing.

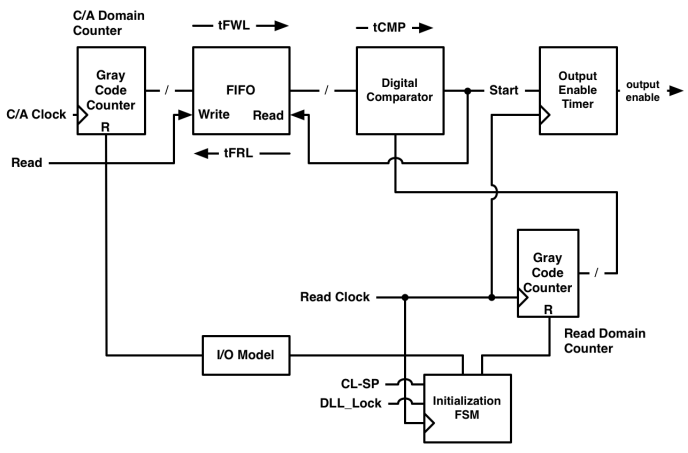

Figure 27.5.3: Read latency control.

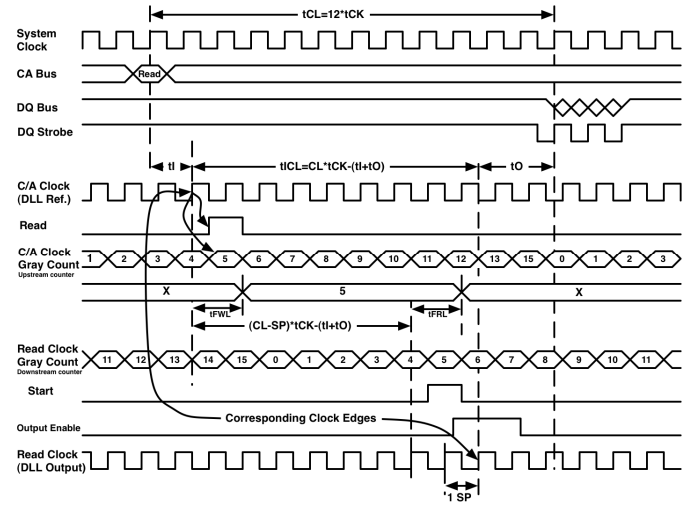

Figure 27.5.4: Read latency control timing diagram.

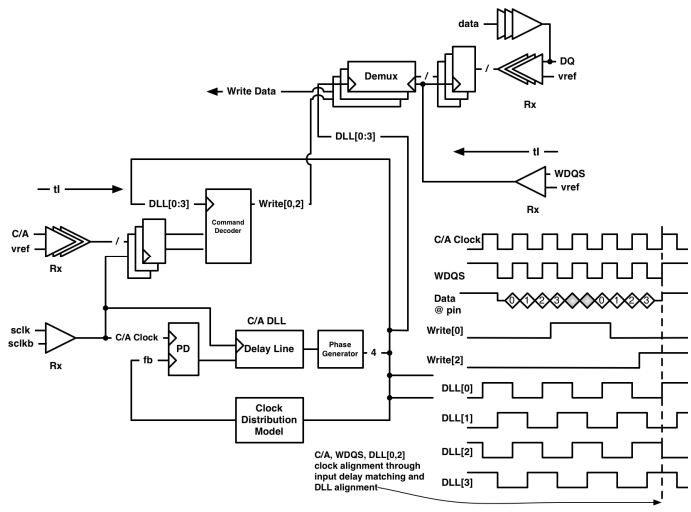

Figure 27.5.5: C/A DLL write latency control.

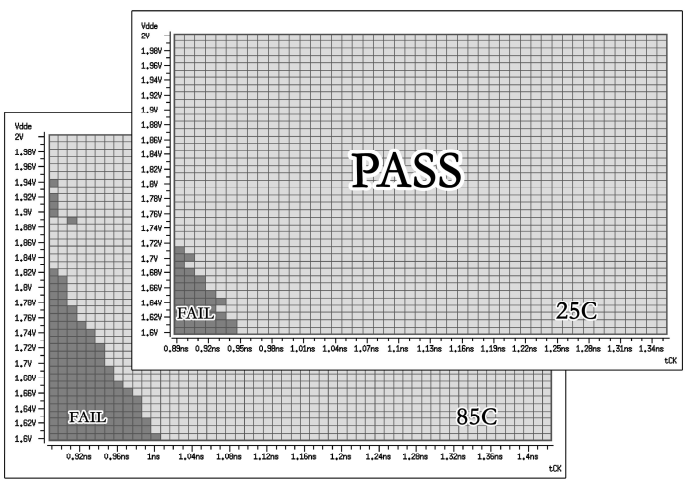

Figure 27.5.6: tCK vs. VDD Schmoo GDDR3, BL=4, T=25C/85C, CL=12.

Continued on Page 617

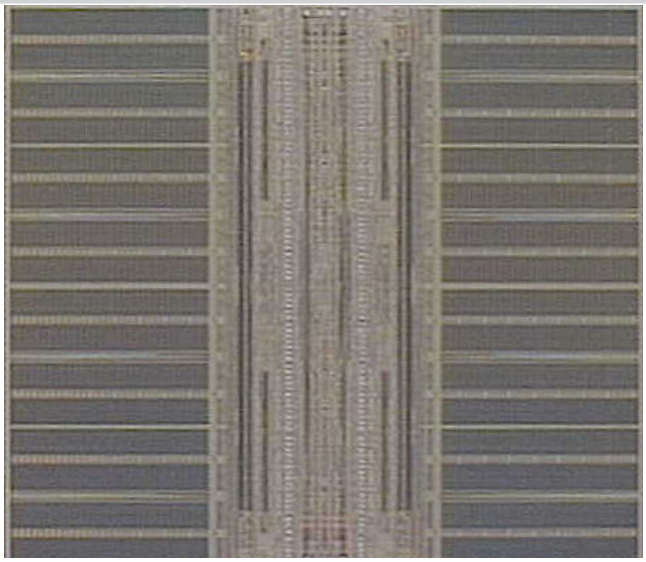

Figure 27.5.7: Die micrograph.

