A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM

Feng Lin, Jason Miller, Aaron Schoenfeld, Manny Ma, and R. Jacob Baker

***Abstract*—This paper describes a register-controlled symmetrical delay-locked loop (RSDLL) for use in a high-frequency double-data-rate DRAM. The RSDLL inserts an optimum delay between the clock input buffer and the clock output buffer, making the DRAM output data change simultaneously with the rising or falling edges of the input clock. This RSDLL is shown to be insensitive to variations in temperature, power-supply voltage, and process after being fabricated in 0.21-μm CMOS technology. The measured rms jitter is below 50 ps when the operating frequency is in the range of 125–250 MHz.**

***Index Terms*—Delay-locked loops, double-data rate, DRAM.**

I. Introduction

In synchronous DRAM, the output data strobe (DQS) should be locked to the data outputs (DQ outputs) for high-speed performance. The clock-access and output-hold times of conventional DRAM designs are determined by the delay time of the internal circuits such as the clock input and output buffers. Variations in temperature and process shifts will change the access time and make the valid data window small. To optimize and stabilize the clock-access and output-hold times, an internal register-controlled delay-locked loop (RDLL) [1], [2] has been used to adjust the time difference between the output and input clock signals in SDRAM. Since the RDLL is an all-digital design, it provides robust operation over all process corners. Another solution to the timing constraints found in SDRAM was given in [3] with the synchronous mirror delay (SMD). Compared to RDLL, SMD does not provide as tight of locking but has the advantage that the time to acquire lock between the input and output clocks is only two clock cycles. As the clock speeds used in DRAM continue to increase, the skew becomes the dominating concern, outweighing the disadvantage of the added time to acquire lock needed in an RDLL.

This paper describes a modified register-controlled symmetrical delay-locked loop (RSDLL) used to meet the requirements of double-data-rate (DDR) SDRAM (read/write accesses occur on both rising and falling edges of the clock). Here, "symmetrical" means that the delay line used in the DLL has the same delay whether a high-to-low or a low-to-high logic signal is propagating along the line. The data output timing diagram of a DDR SDRAM is shown in Fig. 1. The RSDLL is used to increase the valid output data window and diminish the undefined t_{DSDQ} by synchronizing both rising and falling edges of the DQS signal with the output data DQ.

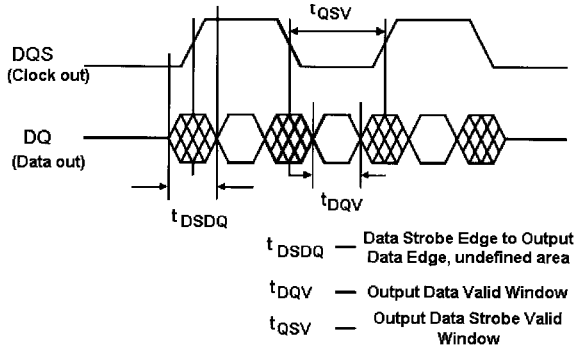

Fig. 1. Data timing chart for DDR DRAM.

The target specifications for the DLL described in this paper are:

1) robust operation eliminating the need for postproduction tuning (something required in an analog implementation);
2) operating frequency ranging from 143 (286 Mb/s/pin) to 250 MHz (500 Mb/s/pin);
3) tight synchronization (skew less than 5% of the cycle time) between the output clock and data on both rising and falling edges of the output clock;
4) low skew between the input and output clocks (with low, $<$5% duty cycle distortion);
5) power-supply-voltage operating range from 2.5 to 3.5 V;
6) portability for ease of use in other processes.

II. RSDLL Architecture

Fig. 2 shows the block diagram of the RSDLL. The replica input buffer dummy delay in the feedback path is used to match the delay of the input clock buffer. The phase detector (PD) is used to compare the relative timing of the edges of the input clock signal and the feedback clock signal, which comes through the delay line, controlled by the shift register. The outputs of the PD, shift-right and shift-left, are used to control the shift register. In the simplest case, one bit of the shift register is high. This single bit is used to select a point of entry for CLKIn in the symmetrical delay line (more on this later). When the rising edge of the input clock is within the rising edges of the output clock and one unit delay of the output clock, both outputs of the PD, shift-right and shift-left, go to logic LOW and the loop is locked. The basic operation of the PD is shown in Fig. 3. The resolution of this RSDLL is determined by the size of a unit delay used in the delay

Manuscript received September 3, 1998; revised November 2, 1998. This work was supported by Micron Technology, Inc.

F. Lin and R. J. Baker are with the Microelectronics Research Center, University of Idaho, Boise, ID 83712 USA (e-mail: danlin@uidaho.edu).

J. Miller, A. Schoenfeld, and M. Ma are with Micron Technology, Inc., Boise, ID 83707-0006 USA.

Publisher Item Identifier S 0018-9200(99)02438-5.

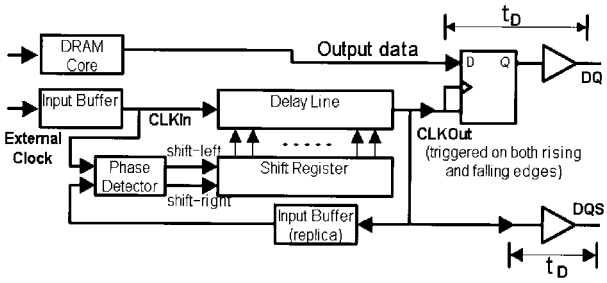

Fig. 2. Block diagram of RSDLL.

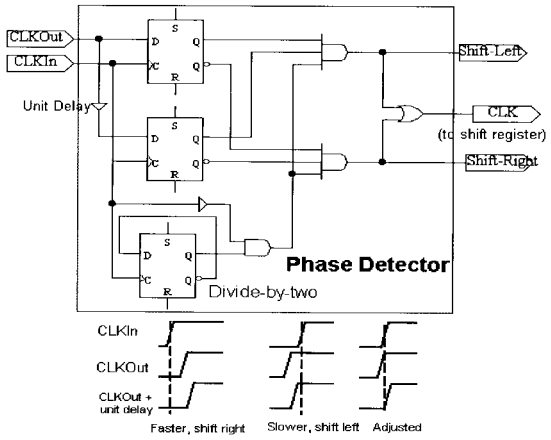

Fig. 3. Phase detector used in RSDLL.

line. The locking range is determined by the number of delay stages used in the symmetrical delay line. Since the DLL circuit inserts an optimum delay time between CLKIn and CLKOut, making the output clock change simultaneously with the next rising edge of the input clock, the minimum operating frequency to which the RSDLL can lock is the reciprocal of the product of the number of stages in the symmetrical delay line with the delay per stage. Adding more delay stages will increase the locking range of the RSDLL at the cost of increased layout area.

III. Circuit Implementation

A. Basic Delay Element

Instead of using an AND gate as the unit-delay stage (NAND + inverter), as was done in [1], we used a NAND-gate-based delay element. The implementation of a three-stage delay line is shown in Fig. 4. The problem when using a NAND + inverter as the basic delay element is that the propagation delay through the unit delay resulting from a HIGH-to-LOW transition is not equal to the delay of a LOW-to-HIGH transition ($t_{\mathrm{PHL}} \neq t_{\mathrm{PLH}}$). Further, this delay varies from one run to another. If the skew between t_{PHL} and t_{PLH} is 50 ps, for example, the total skew of the falling edges through ten stages will be 0.5 ns. Because of this skew, the NAND + inverter delay element cannot be used in a DDR DRAM. In our modified symmetrical delay element, another NAND gate is used instead of an inverter (two NAND gates per delay stage). This scheme guarantees that $t_{\mathrm{PHL}} = t_{\mathrm{PLH}}$ independent of process variations, since while one NAND switches from a HIGH to LOW, the other

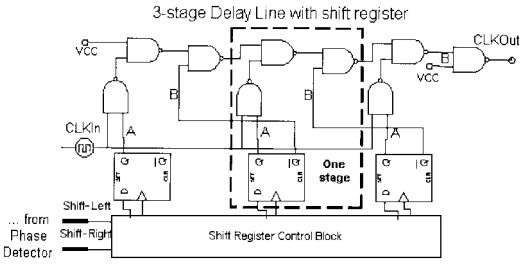

Fig. 4. Symmetrical delay element used in RSDLL.

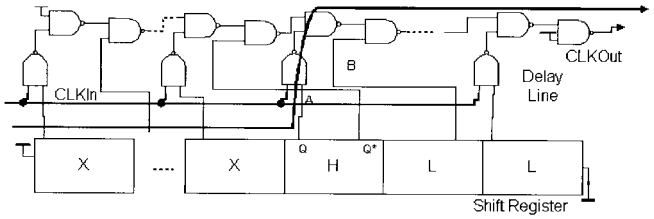

Fig. 5. Delay line and shift register for RSDLL.

switches from LOW to HIGH. An added benefit of the two-NAND delay element is that two point-of-entry control signals are now available. Both are used by the shift register to solve the possible problem caused by the power-up ambiguity in the shift register.

B. Control Mechanism of the Shift Register

As shown in Figs. 4 and 5, the input clock is a common input to every delay stage. The shift register is used to select a different tap of the delay line (the point of entry for the input clock signal into the symmetrical delay line). The complementary outputs of each register cell are used to select the different tap: Q is connected directly to the input A of a delay element, and Q^* is connected to the previous stage of input B. From right to left, the first LOW-to-HIGH transition in the shift register sets the point of entry into the delay line. The input clock will pass through the tap with a high logic state in the corresponding position of the shift register. Since the Q^* of this tap is equal to a LOW, it will disable the previous stages; therefore, it does not matter what the previous states of the shift register are (shown as "don't cares," X, in Fig. 5). This control mechanism guarantees that only one path is selected. This scheme also eliminates power-up concerns since the selected tap is simply the first, from the right, LOW–HIGH transition in the shift register.

C. Phase Detector

To stabilize the movement in the shift register, after making a decision, the phase detector will wait at least two clock cycles before making another decision (Fig. 3). A divide by two was included in the phase detector so that every other decision, resulting from comparing the rising edges of the external clock and the feedback clock, was used. This will provide enough time for the shift register to operate and the output waveform to stabilize before another decision by the PD is implemented. The unwanted side effect of this delay is an increase in the lock time. The shift register is clocked by combining the

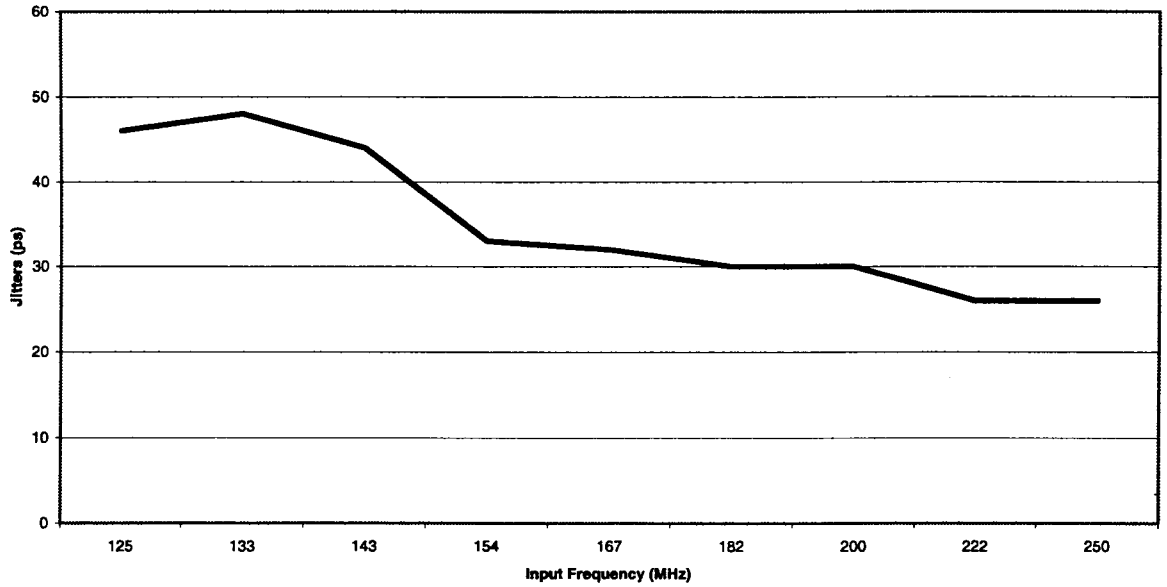

Fig. 6. Measured rms jitter versus input frequency.

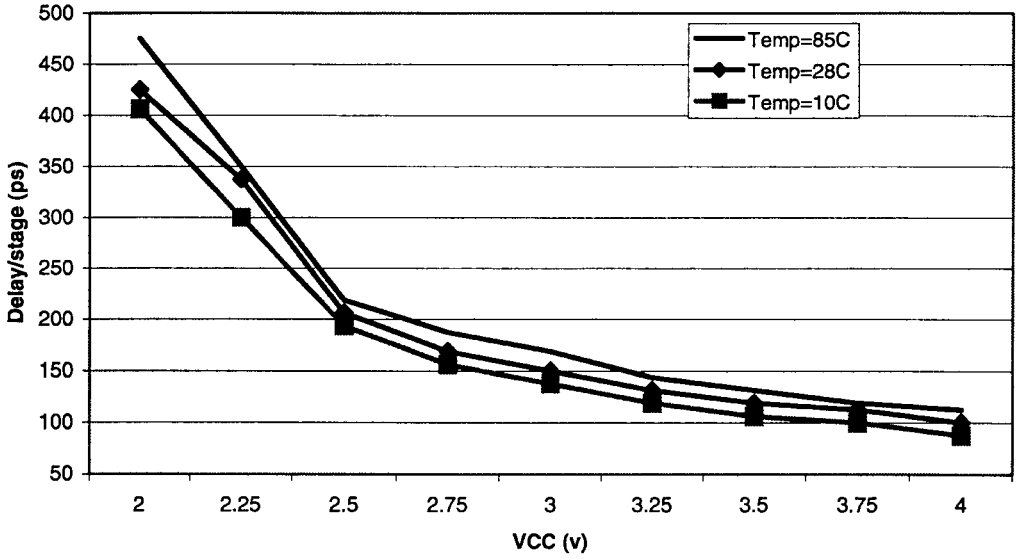

Fig. 7. Measured delay per stage versus VCC and temperature.

shift-left and shift-right signals. The power consumption will decrease when there are no shift-left or -right signals and the loop is locked. Another concern with the phase-detector design is the design of the flip-flops (FF's). To minimize the static phase error, very fast FF's should be used, ideally with zero setup time. Also, the metastability of the flip-flops becomes a concern as the loop becomes locked. This together with possible noise contributions and the need to wait, as discussed above, before implementing a shift-right or -left may increase the desirability of adding additional filtering in the phase detector. Some possibilities include increasing the divider ratio used in the phase detector or using a shift register in the phase detector to determine when a number—say, four—shift-rights or -lefts have occurred. For the present design, we were forced to use a divide by two in the phase detector because of lock time requirements.

IV. Experimental Results

The RSDLL was fabricated in a 0.21-μm, four-poly, double-metal CMOS technology (a DRAM process). We used a 48-stage delay line with an operation frequency of 125–250 MHz. The maximum operating frequency was limited by delays external to the DLL such as the input buffer and interconnect. There was no noticeable static phase error on either rising or falling edges. Fig. 6 shows the resulting rms jitter versus input frequency. One sigma of jitter over the 125–250-MHz frequency range was below 50 ps. The peak-to-peak jitter over this frequency range was below 100 ps. The measured delay per stage versus VCC and temperature is shown in Fig. 7. Note that the 150-ps typical delay of a unit-delay element was very close to the rise and fall times on-chip of the clock signals and represents a practical minimum resolution of a DLL for use in a DDR DRAM fabricated in a 0.21-μm process. The power

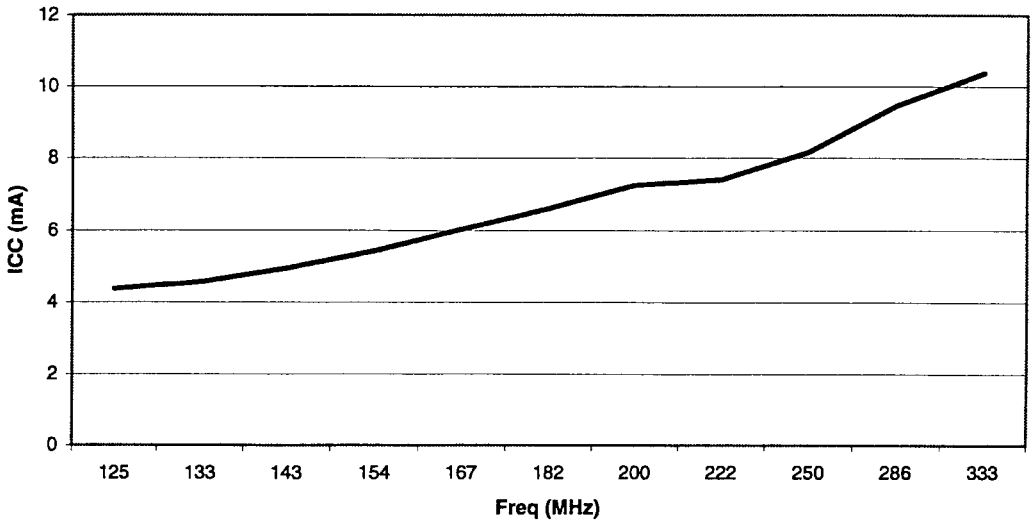

Fig. 8. Measured ICC (DLL current consumption) versus input frequency.

consumption (current draw of the DLL when VCC $= 2.8$ V) of the prototype RSDLL is illustrated in Fig. 8. We found that the power consumption was mainly determined by the dynamic power dissipation of the symmetrical delay line. Our NAND delays in this test chip were implemented with 10/0.21-μm NMOS and 20/0.21-μm PMOS. By reducing the widths of both the NMOS and PMOS transistors, the power dissipation can be greatly reduced without a speed or resolution penalty (with the added benefit of reduced layout size).

V. Conclusions

The concept of a register-controlled symmetrical delay-locked loop has been presented. The modified symmetrical delay element makes the RSDLL useful in DDR DRAM's. Experimental results verify that this RSDLL is stable against temperature, process, and power-supply variations.

Further development of the RSDLL will include investigations into reducing power consumption, implementing phase-locked loops where the symmetrical delay is used as part of a purely digital registered-controlled oscillator, and developing two-loop architectures where coarse loops (resolutions on the order of 100 ps) are used with fine loops (resolutions on the order of 10 ps [2]) for wide tuning range and small static phase errors.

References

[1] A. Hatakeyama, H. Mochizuki, T. Aikawa, M. Takita, Y. Ishii, H. Tsuboi, S.-Y. Fujioka, S. Yamaguchi, M. Koga, Y. Serizawa, K. Nishimura, K. Kawabata, Y. Okajima, M. Kawano, H. Kojima, K. Mizutani, T. Anezaki, M. Hasegawa, and M. Taguchi, "A 256-Mb SDRAM using a register-controlled digital DLL," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1728–1732, Nov. 1997.

[2] S. Eto, M. Matsumiya, M. Takita, Y. Ishii, T. Nakamura, K. Kawabata, H. Kano, A. Kitamoto, T. Ikeda, T. Koga, M. Higashiro, Y. Serizawa, K. Itabashi, O. Tsuboi, Y. Yokoyama, and M. Taguchi, "A 1Gb SDRAM with ground level precharged bitline and non-boosted 2.1V word line," in *ISSCC Dig. Tech. Papers*, Feb. 1998, pp. 82–83.

[3] T. Saeki, Y. Nakaoka, M. Fujita, A. Tanaka, K. Nagata, K. Sakakibara, T. Matano, Y. Hoshino, K. Miyano, S. Isa, S. Nakazawa, E. Kakehashi, J. M. Drynan, M. Komuro, T. Fukase, H. Iwasaki, M. Takenaka, J. Sekine, M. Igeta, N. Nakanishi, T. Itani, K. Yoshida, H. Yoshino, S. Hashimoto, T. Yoshii, M. Ichinose, T. Imura, M. Uziie, S. Kikuchi, K. Koyama, Y. Fukuzo, and T. Okuda, "A 2.5-ns clock access 250-MHz, 256-Mb SDRAM with synchronous mirror delay," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1656–1665, Nov. 1996.