A Self-Adaptive and PVT Insensitive Clock Distribution Network Design for High-Speed Memory Interfaces

Feng (Dan) Lin, Senior Member of Technical Staff
MS 1-356, DRAM R&D
Micron Technology, Inc.
Boise, ID 83707
flin@micron.com

Brent Keeth, Fellow
DRAM R&D
Micron Technology, Inc.
Boise, ID 83707

***Abstract*—A clock distribution network (CDN) insensitive to process, voltage, and temperature (PVT) variations is presented in this paper. Unlike a traditional source-synchronous interface, the CDN uses a current-mode logic (CML) divider and sense-amp-based data receiver for data capture and deserialization. The proposed input path extends its operating range beyond 4-Gb/s/pin without the need for retraining. A unique self-adaptive bias generator based on a Bandgap reference is also disclosed. Simulation data based on the CDN shows a 40% reduction in timing sensitivity for a 100mV supply voltage change and an 85°C temperature change at 4-Gb/s using a 3-metal, 50-nm DRAM process. Design considerations are also addressed based on power, performance, and complexity.**

Keywords-clock distribution network (CDN); current-mode logic (CML); Bandgap reference; deserializer; training; timing skew; voltage and temperature (VT) sensitivity; source synchronous; memory interface

I. INTRODUCTION

As the data rate for I/O interfaces pushes well into the gigahertz range, designing a memory interface, that includes data transceivers, deserializers and clock distribution networks (CDNs), becomes a challenge. Traditionally, memory interfaces favor a source-synchronous relationship between data and clock (or strobe). A dedicated CDN is required to deliver multiple copies of the clock to different locations across the die for both data capture and deserialization.

A technique called matched routing [1, 2], shown in Fig. 1, has been used to maintain source-synchronicity of the internal clocks and data all the way from the device pins to the capture latches. Notice that both Logical Effort [3] and point-to-point matching are utilized based upon the fan-out and loading. Only one data path is illustrated in Fig. 1. Matched routing adds extra circuits for each data path and increases power and area consumption. With single-ended signaling for the data receiver and the lengthened data path, duty-cycle distortion is a major concern for this approach. At higher speeds and for increased data bus width (pin count), matched routing may not be practical.

An alternative scheme employs adjustable input timing and capture latches co-located with the data input pads. The latches are generally built from sense-amp style circuits to maximize their gain and input sensitivity and minimize setup and hold. CDN delay is either backed out through training by the memory controller or by using an internal PLL/DLL [1, 4]. This approach simplifies the data input path, but places a greater burden on clock path design. Since adding a PLL/DLL would greatly increase the complexity, latency, and power of the clock path, we will focus our CDN design on the controller training based method.

The goal of training is to optimize capture timing at the input data latches. The memory controller accomplishes this by delaying or advancing the clock until the latches operate in the center of the data eye. Although process variation and static timing offsets can be tuned out by initial training, delay variations due to voltage and temperature (VT) drift may still exist. Retraining may be necessary if these delay variations are too great, further complicating the link interface design.

This paper addresses these issues by presenting a VT insensitive CDN design. Section II lists different CDN topologies based on current-mode logic (CML) or CMOS buffers. Impacts from VT variations are compared along with other design considerations, such as power and complexity. Section III describes the proposed VT-insensitive CDN with a self-adaptive bias generator. Simulation results are also presented. Section IV outlines conclusions.

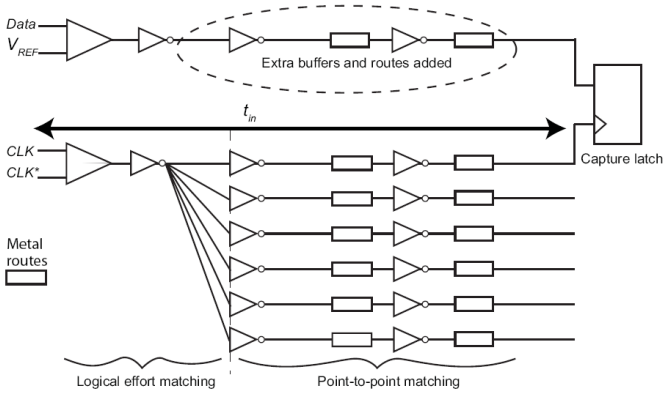

Figure 1. Example of Matched Routings

II. CDN TOPOLOGIES

A. *CML versus CMOS*

Conventionally, a CDN is made up from simple CMOS inverters. The inverters are sized to drive certain loads over a specific distance at predetermined rise and fall times. Despite its elegance and simplicity, a CMOS-based CDN injects switching noise into power supplies and is more susceptible to supply voltage variation. In [4], a voltage regulator was applied to the clock path circuits to mitigate supply sensitivity.

Current-mode logic (CML) [1], on the other hand, appears superb because of high supply noise immunity. Sporting differential circuit elements and constant current consumption, the CML-based CDN is, however, more complicated and consumes more static power than its CMOS counterpart. A comparison [1] for delay of a two-stage CDN is shown in Fig. 2. Over process and voltage corners, the CMOS delay variation is six times greater than that of the CML CDN. However, the CML tree consumes three times more current than the CMOS tree. Similar 0.26ps/°C temperature sensitivity is recorded for a given process.

Voltage sensitivity factor (α) is a good metric for comparison of competing CDN designs. The α factor is defined as

$$\alpha = (\Delta T / T) / (\Delta V / V) \quad (1)$$

where T is the total propagation delay at a given supply voltage (V) and changes in delay and voltage are depicted as ΔT and ΔV respectively. For example, if α is calculated using the data from Fig. 2, (across a VCC range of 1.25V to 1.35V at the typical corner) we find that α_{CML} is approximately 0.1, while α_{CMOS} approaches 1. Conversion circuits that translate the small-swing CML signals to full CMOS signals may increase the total α_{CML} up to 0.5. While a CML design exhibits superior voltage sensitivity, it comes at the cost of higher power. A mixture of CML and CMOS elements may actually produce a CDN design with balanced power and timing performance.

B. *Clock Dividers*

In order to improve signal quality and reduce jitter and ISI for high-speed clock distribution, a clock divider is generally applied [2, 4, and 5]. The four 90-degree phase-shifted clocks generated by clock dividers run at a quarter of the data rate. For 4-Gb/s data rate, the internal divided clocks operate at only 1GHz. This technique greatly improves the achievable speed for an existing DRAM process. Use of 4-phase ¼ rate clocks subsequently simplifies design of the capture latches and deserializers.

One drawback to including a CMOS divider in the clock path is increased VT sensitivity. Therefore, a CML latch based divider, shown in Fig. 3, is instead used in this design. The *CK* and *CK#* are the differential full-speed clocks, and *Nbias* is the CML bias voltage. While the added CML divider helps to reduce voltage sensitivity for the CDN, it does produce higher temperature sensitivity, which must be mitigated.

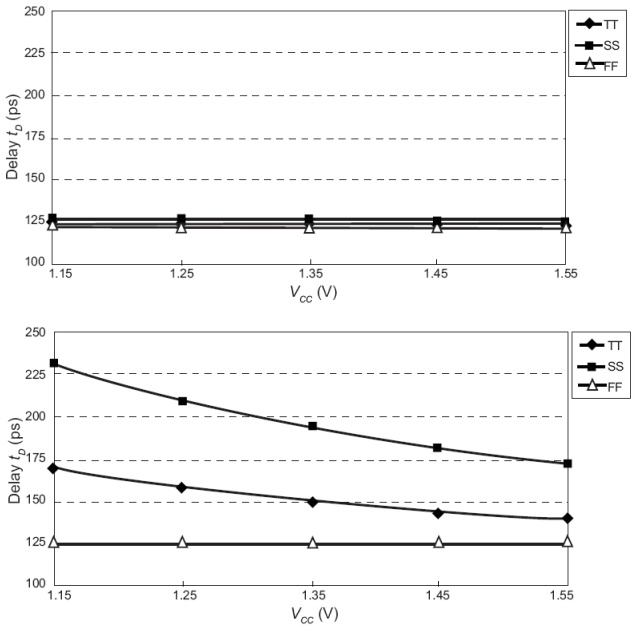

Figure 2. Propogation delay comparison across process and voltage corners at 95°C, 1GHz for two-stage CML (top) and CMOS (bottom) CDNs

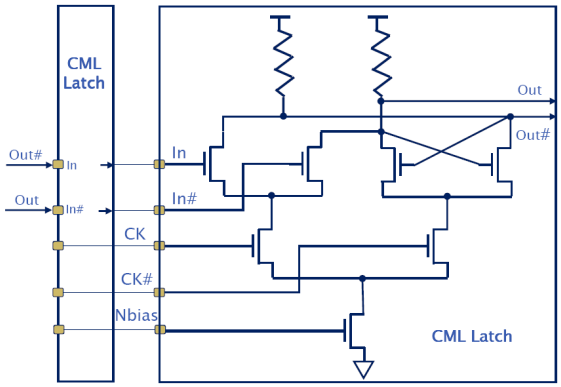

Figure 3. Circuit diagram for a CML latch-based divider

C. *Proposed CDN Topology*

A proposed CDN topology is shown in Fig. 4. The clock receiver (Rx) is a CML based differential receiver with hooks for duty-cycle correction (DCC). The outputs of the Rx are fed into a CML divider for 4-phase ¼ rate clock generation. After conversion from CML to CMOS (via C2C), the 4-phase clocks are distributed through CMOS inverters to data capture latches and deserializers.

With a conventional 2-phase CML CDN, there is no divider in the data capture path (most critical timing path), but there is still a need for an additional 4-phase clock tree for subsequent deserialization. Simulations were run to measure average current for both of the described configurations at 4-Gb/s, 1.35V, 85°C and a typical process corner. The 2-phase CML CDN produced the shortest delay at 252ps versus 372ps for the 4-phase CMOS CDN. Average current is 30mA and 10mA for the 2-phase CML and the proposed 4-phase CMOS CDN respectively.

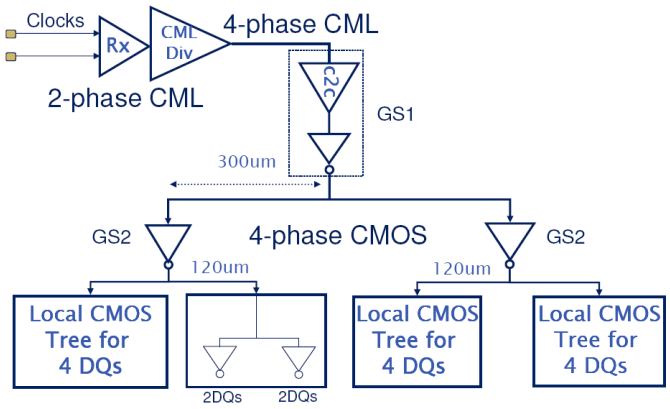

Figure 4. A proposed 16-DQ mixed-mode 4-phase CDN

III. VOLTAGE AND TEMPERATURE SENSITIVITY

As mentioned before, we try to minimize VT sensitivity in the CDN to avoid the need for either retraining or on-die timing circuits, i.e., PLL/DLL. By mixing CML and CMOS circuits in the proposed CDN, we reduce supply sensitivity while minimizing power consumption. However, temperature sensitivity is degraded because of the CML divider. Since CML and CMOS buffers have similar positive temperature coefficients (0.26ps/°C from the previous example), what opportunity is there to improve the temperature sensitivity?

Let's first look at how we generate the bias voltage (e.g., *Nbias* in Fig. 3) for the CML circuits. To achieve a near constant delay, *Nbias* must be adjusted to provide constant current within the CML elements. With a constant current flowing through the load resistor, as shown in Fig. 3, the output swing of the CML also remains constant. Generation of a constant current reference is best accomplished using Bandgap Reference (BGR) based circuitry. An example configuration is shown in Fig. 5 that employs a simple current mirror and diode load referenced to a BGR. The resulting *Vinit* exhibits a negative temperature coefficient that tracks threshold voltage behavior over temperature. To cancel the positive temperature coefficient evident in the CMOS buffers, we explored using the fixed bias voltage scenario depicted in Fig. 5. A resistor serves as the current mirror load for fixed bias generation. For a given process, a fixed bias voltage (*VA*) can be selected and applied to the CML circuits. A plot of temperature sensitivity across three different processes for the proposed 4-phase CDN is shown in Fig. 6. Overall, a 70% reduction in temperature sensitivity was achieved through the use of fixed bias voltages.

The remaining question is how to select the fixed bias voltage across different process corners? A straightforward solution is exploiting the fixed swing bias circuit initially as a reference. A block diagram for a proposed self-adaptive bias generator is shown in Fig. 7. The self-adaptive process starts with the BGR initialization. After the BGR is started and settled, the temperature-compensated current is applied to both fixed-swing and fixed-bias (shown as an Adjust Network in Fig. 7) generators. The process info is captured by the fixed-swing generator and represented by the *Vinit*.

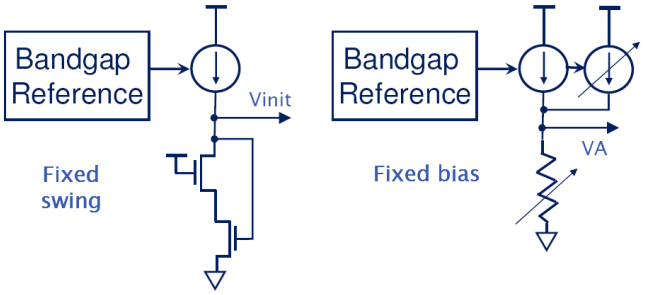

Figure 5. CML bias generation: fixed swing versus fixed bias

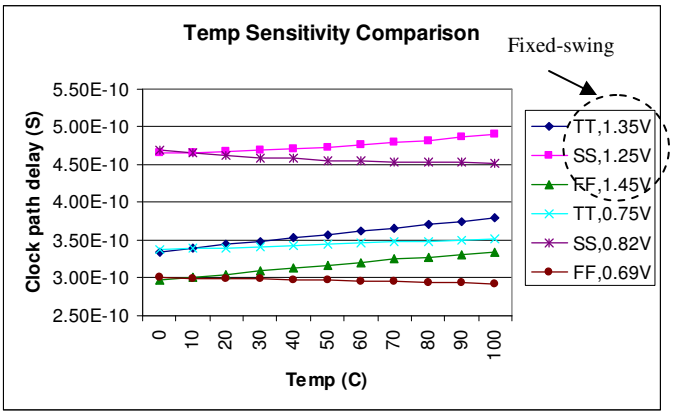

Figure 6. Temp sensitivity of the CDN across processes and voltages with two difference bias schemes: fixed swing or fixed bias

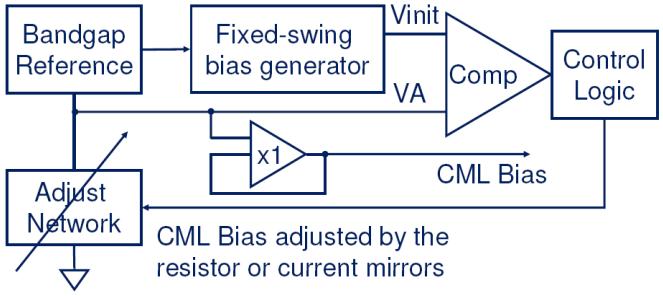

Figure 7. Block diagram of a self-adaptive bias generator

One implementation of the control logic sets the fixed bias value *VA* to its minimum initially. The range of the *VA* can be determined by sweeping the process corners. After comparing *Vinit* and *VA* via a comparator, the results are fed into the control logic to determine if the *VA* needs to be adjusted or not. When the *VA* is greater than the *Vinit*, the calibrating process is completed. A unit gain buffer is inserted between the *VA* and the final CML bias voltage. The fixed-swing generator, comparator, and control logic can be idled or shut off to save power after the calibration.

A flow chart of the self-adaptive calibration is shown in Fig. 8. For a given device on a given process, the calibration only needs to be run once during the power-up initialization. Three-bit binary weighted current sources are used for adjustment (Fig. 5). A 3-bit UP counter selects the value of the current. The step size is around 20mV with a 10kΩ resistor.

The calibration takes a couple hundred cycles, depending on the process corner and clock frequency. Calibration can be part of the power-up sequence.

Simulation results based upon the proposed CDN are summarized in Table I and Table II. Both results for the fixed-swing and fixed-bias generators are listed for comparison. Using typical models (TT) from a 3-metal 50nm DRAM process, the fixed bias is set around 0.77V, where the CML bias for the fixed-swing generator can vary between 0.71 to 0.78V. The timing improvement over a 100mV and 85°C VT change is about 40%. For a temperature change only at 1.45V, from Table II, a 68% improvement is achieved.

Fig. 9 compares the temperature sensitivity across different bias voltages, as well as the fixed-swing bias. One observation is that the slope of the curve can be adjusted either negatively or positively depend on the bias voltage. Another observation is that lower temperatures produce wider variations. It is preferable to calibrate at lower temperatures.

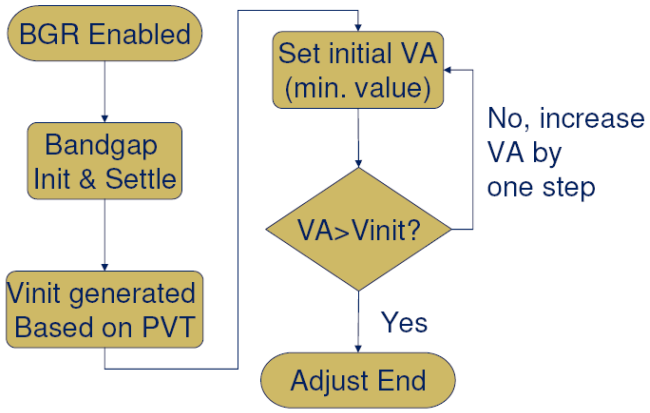

Figure 8. Flow chart of the self-adaptive process

TABLE I. SIMULATION RESULTS ACROSS VOLTAGE & TEMPERATURE

TT, 4Gpbs	Propagation delay of the clock distribution network		
	Voltage, Temperature	*Fixed-swing (ps)*	*Fixed-bias (ps)*
1	1.45V, 0°C	378	383
2	1.45V, 85°C	409	373
3	1.35V, 0°C	394	402
4	1.35V, 85°C	425	385

TABLE II. SIMULATION RESULTS ACROSS TEMPERATURE

TT, 4Gpbs	Propagation delay of the clock distribution network		
	Voltage, Temperature	*Fixed-swing (ps)*	*Fixed-bias (ps)*
1	1.45V, 0°C	378	383
2	1.45V, 85°C	409	373
Delta	0V, 85°C	31	10

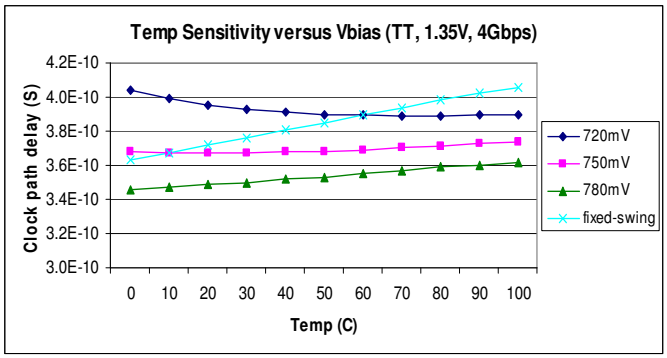

Figure 9. Temperature sensitivity versus different bias votages

For low-voltage operations, the improvement using the proposed VT insensitive CDN is not as significant as that at higher voltages. At 1.35V, only 45% improvement is recorded over 85°C temperature change, as compared to 68% at 1.45V. For low-power low-voltage memory applications, voltage sensitivity is dominant factor for the timing budget. A clock distribution design with more CML components may be more appropriate in this case.

IV. CONCLUSION

To obtain high-speed operating for memory interfaces, a multi-phase, voltage and temperature insensitive clock distribution network (CDN) is proposed. Design consideration for different CDN topologies were analyzed with a focus on power and performance. Current-mode logic is utilized to mitigate supply sensitivity while a unique self-adaptive bias generator reduces temperature dependency. The proposed CDN can be easily applied to output timing path as well.

ACKNOWLEDGMENT

The authors like to thank Jason Brown, Seong-hoon Lee, Brian Johnson, Tim Hollis, and Dragos Dimitriu for valuable discussions.

REFERENCES

[1] B. Keeth, R. J. Baker, B. Johnson, F. Lin, *DRAM Circuit Design - Fundamental and High-Speed Topics*. Wiley-IEEE Press, 2007.

[2] F. Lin, R. Royer, B. Johnson, B. Keeth, "A wide-range mixed-mode DLL for a combination 512 Mb 2.0 Gb/s/pin GDDR3 and 2.5 Gb/s/pin GDDR4 SDRAM", *IEEE J. Solid-State Circuits*, vol. 43, No. 3, pp. 631-641, March 2008.

[3] I. E. Sutherland, R. F. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. San Diego, CA: Academic Press, 1999.

[4] S. Bae, Y. Sohn, K. Park, K. Kim, D. Chung, J. Kim, S. Kim, M. Park, J. Lee, S. Bang, H. Lee, I. Park, J. Kim, D. Kim, H. Kim, Y. Shin, C. Park, G. Moon, K. Yeom, K. Kim, J. Lee, H. Yang, S. Jang, J. Choi, Y. Jun, K. Kim, "A 60nm 6Gb/s/pin GDDR5 Graphics DRAM with Multifaceted Clocking and ISI/SSN-Reduction Techniques", in *IEEE ISSCC Dig. Tech. Papers*, 2008, pp. 278-279

[5] K. Lee, J. Cho, B. Choi, G. Lee, H. Jung, W. Lee, K. Park, Y. Joo, J. Cha, Y. Choi, P. B. Moran, and J. Ahn, "A 1.5-V 3.2 Gb/s/pin Graphic DDR4 SDRAM With Dual-Clock System, Four-Phase Input Strobing, and Low-Jitter Fully Analog DLL", *IEEE J. Solid-State Circuits*, vol. 42, no. 11, pp. 2369-2377, Nov. 2007