

A Tree-Topology Multiplexer for Multiphase Clock System

Hungwen Lu, Chauchin Su, *Member, IEEE*, and Chien-Nan Jimmy Liu, *Member, IEEE*

Abstract—This paper proposes a tree-topology multiplexer (MUX) that employs a multiphase low-frequency clock rather than a high-frequency clock. Analysis and simulation results show that the proposed design can achieve higher bandwidth and be less sensitive to process variations than the conventional single-stage MUX. In order to verify the feasibility, this proposed design is integrated with a multiphase phase-locked loop and a low-voltage differential signaling driver in a 0.18- μm CMOS technology. Measured results indicate that the proposed design can operate up to 7 gigabits/s under 0.3-UI jitter limitation.

Index Terms—I/O, multiplexer, MUX, serdes, serializer.

I. INTRODUCTION

SERIALIZED data transmission systems are usually adopted when the ratio of the on-chip data bandwidth to the off-chip I/O pin count becomes large. Multiplexers (MUX) and demultiplexers (DEMUX) are applied to convert parallel low-speed data into serial high-speed data or vice versa. Conventionally, there are tree-type [1] and single-stage [2] MUX architectures.

A tree-type MUX, as shown in Fig. 1, is composed of multiple 2–1 MUX cells organized in a tree structure. It requires a high-frequency clock for the final stages. The frequency is half the data rate. The clock is then divided to control the successive stages. At each stage, D-type flip-flops (DFFs) are used to latch the data temporarily in order to let two input data be out of phase. It guarantees sufficient setup time and hold time for the output switch to achieve high bandwidth. However, the bandwidth demands on clock buffers and registers result in extra power consumption and circuit area.

A single-stage MUX, as shown in Fig. 2, is composed of multiple open-drain NAND cells. It is driven by a low-speed multiphase clock. As a result, its area and power consumption are lower than that of a tree-type MUX. However, due to its large parasitic loading at the output node, the speed is also lower.

A multiphase clock generator is usually implemented by a multistage ring oscillator (OSC), whereas a high-frequency clock generator is normally implemented by an LC-tank OSC.

Manuscript received December 1, 2006; revised February 26, 2008. First published June 6, 2008; current version published February 4, 2009. This work was supported in part by the National Science Council under Contract NSC95-2221-E-009-328, by the Industrial Technology Research Institute, and by the Ministry of Economic Affairs under Contract MOEA95-EC-17-A-01-S1-037 of Taiwan. This paper was recommended by Associate Editor M. Stan.

H. Lu and C.-N. J. Liu are with the Department of Electrical Engineering, National Central University, Zhongli 32001, Taiwan (e-mail: s9521011@cc.ncu.edu.tw).

C. Su is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan.

Digital Object Identifier 10.1109/TCSI.2008.926578

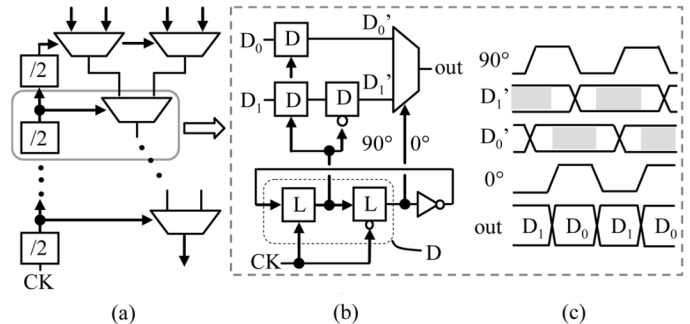


Fig. 1. (a) Tree-type MUX schematic. (b) 2–1 MUX cell. (c) Timing diagram of 2–1 MUX cell.

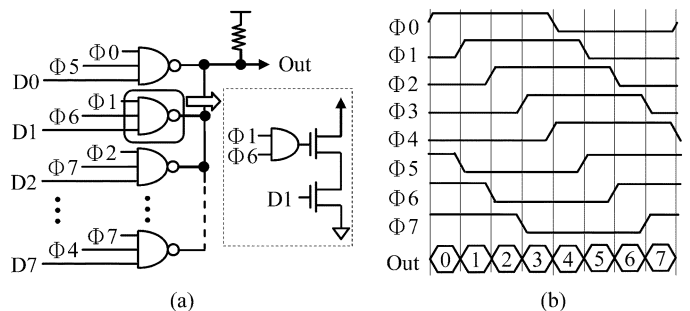


Fig. 2. (a) Single-stage MUX schematic and (b) its timing diagram.

Multiphase clock generators are likely to have wider frequency ranges than high-frequency clock generators [3], [4] do. Low-cost and wide-range transceivers can be implemented by using multiphase clock generators [5]–[7]. However, as stated earlier, the speed limitation is the main drawback.

In this paper, we propose a multiphase-clock-based tree-topology MUX in order to achieve high speed and low power at the same time. The same 2–1 MUXs are used as MUX cells and clock deskew module to eliminate the skew between data paths and clock paths. Without retiming DFFs, the area overhead and power consumption can be reduced.

This paper is organized as follows. Section II describes the proposed MUX architecture and its detailed operations. Section III analyzes the proposed MUX and compares its jitter performance with that of a single-stage MUX mathematically and simulationwise. Section IV shows the chip implementation and measured results. Finally, Section V concludes this paper.

II. PROPOSED MUX

Fig. 3 shows the proposed MUX structure and its timing diagram. The structure is similar to a tree-type MUX with multiple 2–1 MUX organized in a binary tree structure. We have to note that no retiming DFF exists in the proposed MUX. The MUX is

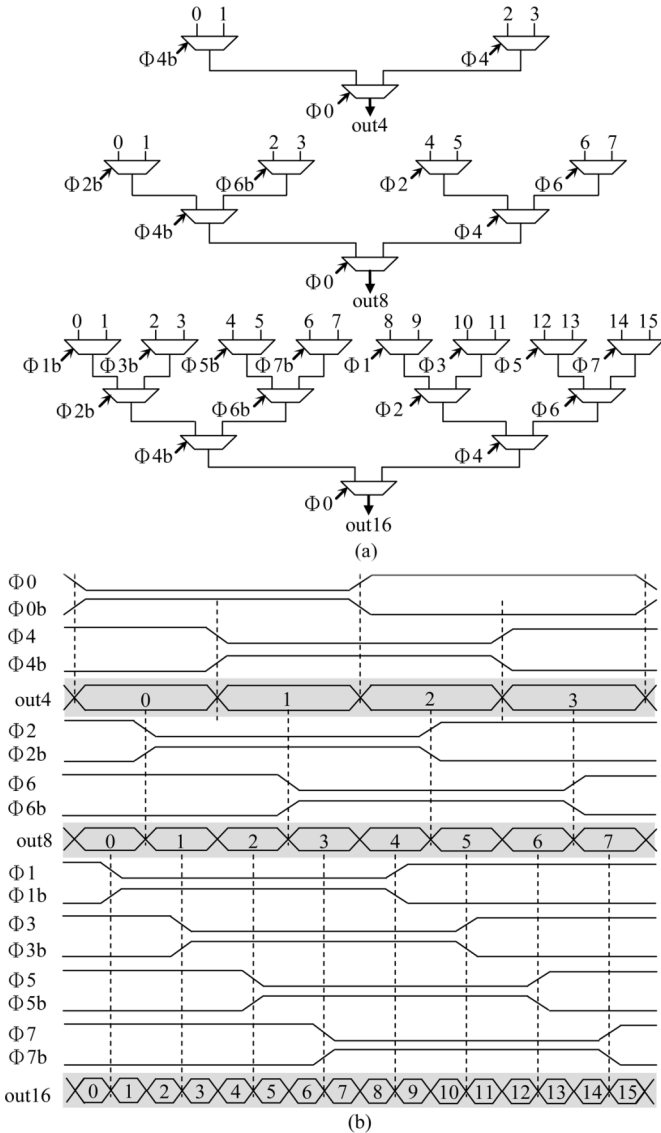


Fig. 3. (a) Proposed MUX schematic and (b) its timing diagram.

not controlled by a high-frequency clock and its divided clocks. It is controlled by different clock phases organized regularly. The first stage is controlled by $\Phi 0$ which outputs data at 0° and 180° . The second stage is controlled by the phases between 0° and 180° of the first stage, namely, 90° and 270° . Again, the third-stage controls are in the middle of the second stage, or 45° , 135° , 225° , and 315° . Consequently, the fourth stage is controlled by 22.5° , 67.5° , 112.5° , 157.5° , 202.5° , 247.5° , 292.5° , and 337.5° .

The major distinguishing feature is the implementation of low-speed multiphase clocks for the tree-type MUX. The parasitic parameters at each stage are minimized by multiplexing only two inputs, so it achieves high bandwidth. Unlike that of the single-stage MUX, the performance of the tree-type MUX remains the same regardless of the number of inputs. The frequency of intersymbol interference (ISI) remains unchanged due to constant output parasitic effects. Note that a single-stage MUX deteriorates as the number of inputs increases.

Although the proposed tree-type structure solves the speed limitation and alleviates the jitter problem, it still has several

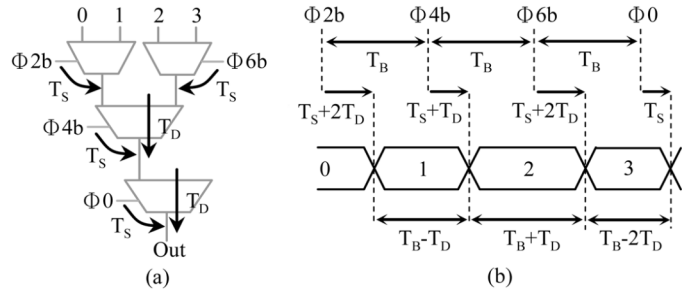


Fig. 4. (a) Propagation delay mismatch and (b) unequal bit period.

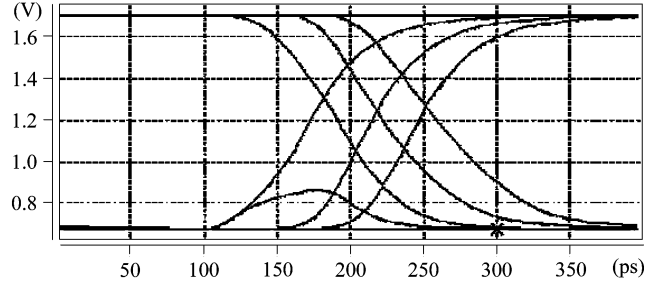


Fig. 5. Output eye diagram while regarding the propagation delay mismatch.

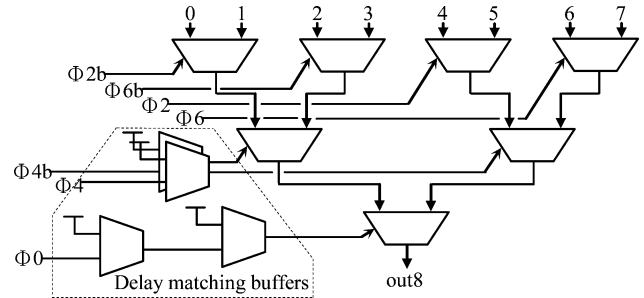


Fig. 6. Proposed MUX schematic with delay-matching buffers.

drawbacks. The delay path mismatch creates deterministic jitter, as shown in Fig. 4. T_D and T_S denote the delays for the data and control inputs for the MUX, respectively. Therefore, the data have different delay phases to the output, depending on their control. For example, the delay of $D1$ or the edge of $\Phi 2b$ is $T_S + 2T_D$, while the delay of $D0$ or the edge of $\Phi 0$ is T_S . This mismatch is transformed into a data period variation. For the 8–1 MUX in Fig. 4(b), the data periods are $T_B \pm T_D$ and $T_B \pm 2T_D$. Here, T_B is the data period, and the delay skew is $2T_D$. For a general $N-1$ MUX, the maximal skew can be derived as $(\log_2 N - 1) \times T_D$. Fig. 5 shows the jitter caused by such a period variation.

In order to solve this delay mismatch problem, delay-matching buffers are inserted to match the delay, as shown in Fig. 6. The delay-matching buffers are exactly the same as 2–1 MUX cells being used in the data path. Its purpose is to balance the skew of T_D in each stage of the data path. By letting clocks go through the same MUX, the skews are compensated. Since the tree-type MUX and delay-matching buffers are identical, the design is less sensitive to process, voltage, and temperature variations. It will be verified in the analysis and simulation later in this paper.

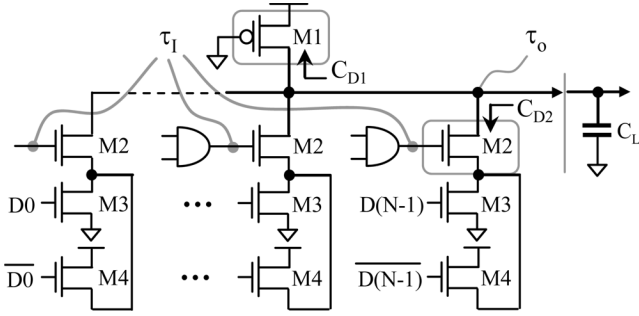


Fig. 7. Circuit model of the single-stage MUX.

III. TIMING JITTER ANALYSIS

The bandwidth of the MUX is determined by the jitter performance in addition to the 3-dB bandwidth of the MUX cell. The sources of deterministic jitter include process variations, simultaneous switching noise (SSN), and ISI. Process variation causes mismatch between control phases. The SSN caused by the large current change during the transition generates power supply noise. ISI becomes significant when the data transition time is close to or larger than the data period.

In order to compare the bandwidth, the jitter performances of the proposed MUX and the single-stage MUX are analyzed under the influence of process variation and ISI.

A. Jitter Caused by Process Variation

Process variation affects many aspects of a circuit. Among them, the performances of transistors and their associated parasitic capacitances are closely related to the jitter performance. For a 2–1 MUX cell, its driven node can be modeled as a simple one-pole system. Let T_1 denote the delay time for a signal to reach 50% of its amplitude in a one-pole system. Then, T_1 can be derived as follows:

$$1 - e^{-T_1/RC} = 0.5 \Rightarrow T_1 = RC \cdot \ln 2. \quad (1)$$

The delay is linearly proportional to the RC time constant. R is the channel turn-on resistance of the driving transistor, and C is the total loading capacitance. Since R and C are changed under process variation, the delay time variation can be derived as follows:

$$\Delta T_1 = [(R + \Delta R)(C + \Delta C) - R \cdot C] \ln 2 \approx [\Delta R \cdot C] \ln 2. \quad (2)$$

In (2), ΔT_1 can be regarded as jitter for the following reasons. For a MUX, data pass through different paths. The variations on the path delays create timing jitter. According to the statistical analysis of process variations, the variation on the channel resistance greatly exceeds that on the total parasitic capacitance. Therefore, it is concluded that ΔT_1 is dominated by ΔR and C . Therefore, as shown in Fig. 7, for a conventional single-stage MUX, the jitter is derived as

$$\Delta T_{1,Con} = \Delta R(N \cdot C_{D2} + C_{D1} + C_L) \ln 2. \quad (3)$$

C_{D1} , C_{D2} , and C_L are the parasitic capacitances of the pull-up PMOS, the pull-down NMOS, and the output load, respectively.

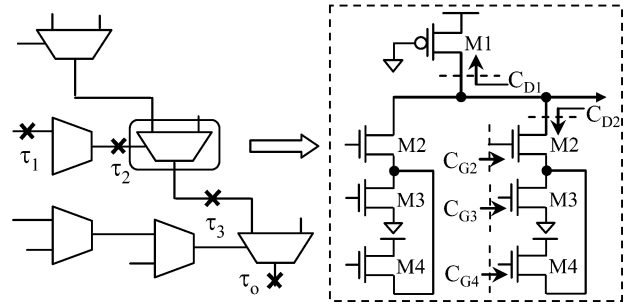


Fig. 8. Circuit model of the proposed MUX.

TABLE I
(A) PARASITIC PARAMETERS OF THE MUX CELL

Simulated size		Parasitic capacitance & resistance			
M1	1.7u/0.18u, M=2	C_{D1}	9.8 f	C_{G3}	4.28 f
M2	2.5u/0.18u, M=2	C_{D2}	5.7 f	C_{G4}	0.38 f
M3	1.45u/0.18u, M=2	C_{G2}	13 f	C_L	10.7 f
M4	0.35u/0.18u, M=1	R	1.15K		
$(N \cdot C_{D2} + C_{D1} + C_L)$ in (3)		$[\log_2 N(2C_{D2} + C_{D1}) + \dots + C_L]$ in (5)			
Single-stage MUX8	66.1 f	Proposed MUX8	91.96 f		
Single-stage MUX16	111.7 f	Proposed MUX16	117.82 f		
Single-stage MUX32	202.7 f	Proposed MUX32	143.68 f		

N is the number of multiplexing inputs. The total capacitance inside the parentheses is the total capacitance at the output node. ΔR is the variation of the channel resistance of the driving transistors.

For the proposed MUX, ΔT_1 is the accumulation of jitter in the stages that the signal passes through, as shown in Fig. 8

$$\Delta T_{1,Pro} = \Delta \tau_1 + \Delta \tau_2 + \dots + \Delta \tau_N \quad (4)$$

$$= \Delta R \cdot \left[\log_2 N(2C_{D2} + C_{D1}) + (\log_2 N - 2)(C_{G3} + C_{G4}) + C_{G2} + C_L \right] \ln 2. \quad (5)$$

$C_{G2} - C_{G4}$ are the gate capacitances, and ΔR is the variation of the channel resistance of the driving MOS. The total capacitance in the bracket can be regarded as the total capacitance on the data path. Note that we assume that all nodes are driven by transistors of the same size. Since the single-stage MUX has a parallel structure, the total capacitance is proportional to N . However, a tree structure has a $\log_2 N$ complexity. For large N , the proposed structure has a smaller jitter. Through simulating the jitter caused by process variation, Table I shows the simulated size and extracted capacitances used in both MUXs in the upper half. By (3) and (5), the low half shows the total capacitance for the MUX with different numbers of inputs (8, 16, and 32). As one can see, single-stage MUXs have less jitter when N is small. However, tree-type MUXs are better when N is large. For $N = 16$, they have the same jitter performance.

Fig. 9 shows Monte Carlo simulation using HSPICE. Thirty samples are taken and averaged for each case. As one can see, the proposed MUX equals the single-stage MUX when $N = 16$.

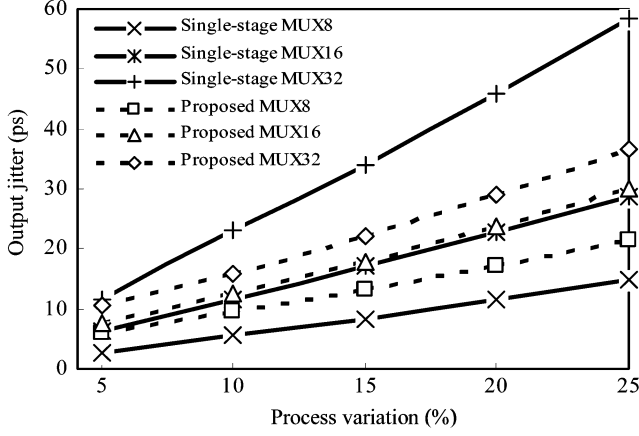


Fig. 9. Simulated jitter caused by process variation.

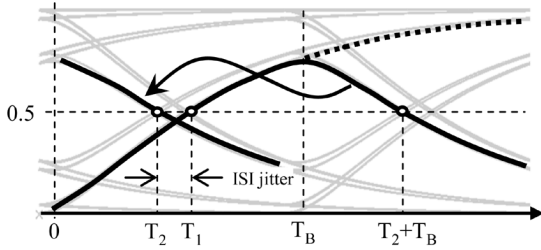


Fig. 10. Timing jitter caused by ISI effects.

It is much better when $N = 32$, as suggested in Table I. Of course, the single-stage MUX is better when $N = 8$.

B. ISI Jitter Analysis

Fig. 10 shows the simulated eye diagram. The jitter is caused by ISI effects. Here, T_1 and T_2 are the times the output waveforms pass through $1/2$ when rising and falling. The jitter is $(T_1 - T_2)$. To calculate it, the S -domain and time-domain transfer functions, namely, $f(s)$ and $f(t)$, respectively, must be obtained first. The impulse responses $F(t)$ of the MUX system are

$$F(t) = u(t) \int_0^T f(t)dt - u(t - T_B) \int_0^T f(t - T_B)dt \quad (6)$$

$$F(T_1) = \frac{1}{2}F(T_B + T_2) = \frac{1}{2}. \quad (7)$$

With regard to the transfer function, T_1 and T_2 can be solved by mathematical software such as MATLAB. Thus, the jitter is then obtained.

C. ISI Jitter Calculation for the Single-Stage MUX

As shown in Fig. 7, the S -domain and time-domain transfer functions of a single-stage MUX are

$$f(s) = \frac{1}{(1 + s\tau_I)(1 + s\tau_O)} \quad (8)$$

$$f(t) = \frac{e^{-t/\tau_I} - e^{-t/\tau_O}}{\tau_I - \tau_O}. \quad (9)$$

τ_I and τ_O are the time constants at the phase input and the data output, respectively

$$\tau_I = R(C_{D2} + C_{D1} + C_{G2}) \quad (10)$$

$$\tau_O = R(N \cdot C_{D2} + C_{D1} + C_L). \quad (11)$$

Substituting (9) into (6), the impulse response $F(t)$ of the single-stage MUX is

$$F(t) = u(t) \left(\frac{1 - \tau_I e^{-t/\tau_I} - \tau_O e^{-t/\tau_O}}{\tau_I - \tau_O} \right) - u(t - T_B) \left(\frac{1 - \tau_I e^{-(t-T_B)/\tau_I} - \tau_O e^{-(t-T_B)/\tau_O}}{\tau_I - \tau_O} \right). \quad (12)$$

Substituting (12) into (7), T_1 and T_2 can be obtained from

$$\frac{\tau_I e^{-T_1/\tau_I} - \tau_O e^{-T_1/\tau_O}}{\tau_I - \tau_O} = \frac{1}{2} \quad (13)$$

$$\frac{\tau_I e^{-T_2/\tau_I} (1 - e^{-T_B/\tau_I}) - \tau_O e^{-T_2/\tau_O} (1 - e^{-T_B/\tau_O})}{\tau_I - \tau_O} = \frac{1}{2}. \quad (14)$$

With (11) and (12), by using MATLAB, one can obtain T_1 and T_2 that satisfy the equations. Again, the ISI jitter can be obtained.

D. ISI Jitter Calculation for the Proposed MUX

For the proposed MUX shown in Fig. 8, each 2–1 MUX can be modeled by the cascade of multiple one-pole systems. Here, $\tau_1, \tau_2, \dots, \tau_n$ denote the time constants at the outputs of different stages, and τ_O is the time constant at the output of the last stage. Assume that $\tau_1 \approx \tau_2 \approx \dots \approx \tau_n = \tau_I$ because they have the same circuit topology

$$\tau_I = R(2C_{D2} + C_{D1} + C_{G1}) \quad (15)$$

$$\tau_O = R(2C_{D2} + C_{D1} + C_L). \quad (16)$$

Assume that there are n stages, $n = \log_2 N$. The S -domain and time-domain transfer functions derived from the convolution are

$$f(s) = \frac{1}{(1 + s\tau_I)^n (1 + s\tau_O)} \quad (17)$$

$$f(t) = \int_0^t \left(\frac{1}{\tau_I^n} \frac{\tau^{n-1}}{(n-1)!} e^{-\tau/\tau_I} \right) \left(\frac{1}{\tau_O} e^{-(t-\tau)/\tau_O} \right) d\tau. \quad (18)$$

The step input response, or the integration of the time-domain transfer function, is derived as follows. Note that the derivation process is complicated. The authors will provide the step-by-step process upon request

$$\int_0^t f(t)dt = 1 - P^n e^{-t/\tau_O} - e^{-t/\tau_I} \times \sum_{K=0}^{n-1} \frac{(t/\tau_I)^K}{K!} (1 - P^{n-K}) \quad (19)$$

$$P = \tau_O / (\tau_O - \tau_I). \quad (20)$$

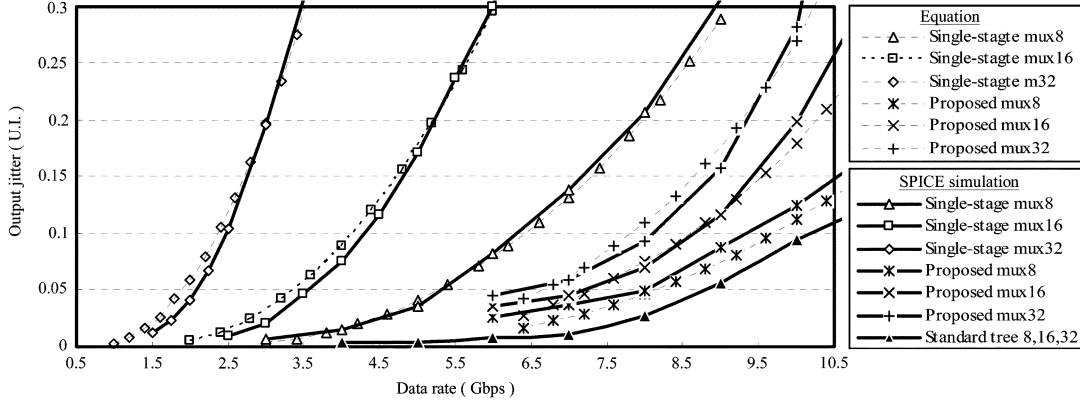


Fig. 11. Simulated and calculated jitters caused by ISI effects.

In (19), K is a positive integrator. By (6) and (7), we are able to obtain the following equations similar to (13) and (14)

$$P^n e^{-T_1/\tau_0} + e^{-T_1/\tau_1} \times \sum_{K=0}^{n-1} \frac{(T_1/\tau_1)^K}{K!} (1 - P^{n-K}) = \frac{1}{2} \quad (21)$$

$$\left[P^n e^{-T_2/\tau_0} + e^{-T_2/\tau_1} \times \sum_{K=0}^{n-1} \frac{(T_2/\tau_1)^K}{K!} (1 - P^{n-K}) \right] - \left[P^n e^{-(T_2+T_B)/\tau_0} + e^{-(T_2+T_B)/\tau_1} \times \sum_{K=0}^{n-1} \frac{((T_2+T_B)/\tau_1)^K}{K!} (1 - P^{n-K}) \right] = \frac{1}{2}. \quad (22)$$

Similarly, by using MATLAB, one is able to obtain T_1 and T_2 that satisfy the equations as (21) and (22). As a result, the jitter caused by ISI is obtained.

E. Simulation Results of Jitter Caused by ISI

According to the same setting in Table I, Fig. 11 shows the simulated and calculated jitters for MUXs of different topologies, number of inputs, and data rates. The X axis is the data rate, and the Y axis is the jitter in unit intervals (UIs). The dotted lines show the results obtained by (13)–(14) for the single-stage MUX and (21)–(22) for the proposed MUX. The standard tree-type MUX in Fig. 1 is also included.

First of all, the analyzed results match well with the simulated results. Second, the proposed MUX has less jitter than the single-stage MUX for the same data rate. Third, the proposed MUX can operate at higher data rates than single-stage ones. Also note that for the proposed one, the ISI jitter increases linearly proportionally with the number of stages or $\log_2 N$, whereas the ISI jitter is linearly proportional to N for a single-stage one.

As compared with the standard tree-type MUX, the proposed MUX has the better jitter performance due to the retiming at the output stage. However, its power consumption is another issue.

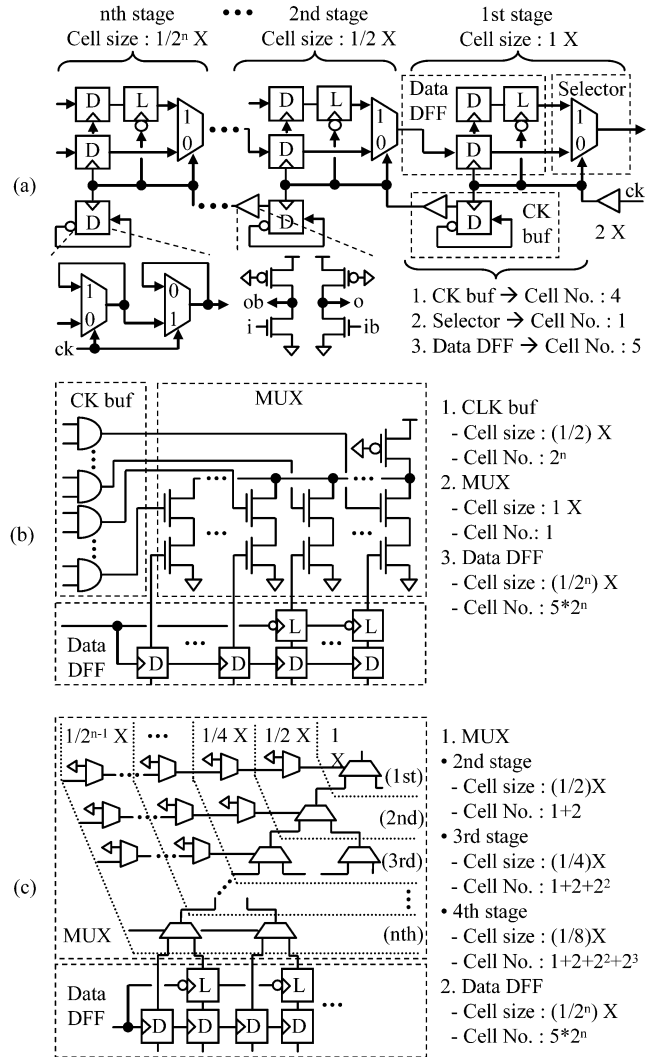


Fig. 12. Circuit structures of (a) the standard tree-type MUX, (b) the single-stage MUX, and (c) the proposed MUX.

F. Power Consumption

Fig. 12 shows the circuit structures of different MUX architectures. There, n is the number of stages. Cell No is the number of cells being used in a stage. Cell Size is the size scaling of a

stage as compared to the output stage. For example, for an 8–1 tree-type MUX, the cell sizes are scaled as (1, 1/2, and 1/4) according to the data rate. For logic gates, currents are normalized to a single selector as

$$I_A = I_S \quad I_D = I_B = 2 \cdot I_S. \quad (23)$$

I_A , I_D , I_B , and I_S are the currents of AND gates, DFFs, buffers, and selectors. For the standard tree-type MUX, the circuit sizes are halved, and the total number of blocks is doubled stage by stage. Hence, the total current in each stage remains the same

$$I_{\text{tree, nth stage}} = 2^n (10) \frac{1}{2^n} \cdot I_S = 10 \cdot I_S \quad (24)$$

$$I_{\text{tree}} = \underbrace{4 \cdot I_S}_{0 \text{ stage}} + \underbrace{(n-1) \cdot 10 \cdot I_S}_{1 \sim (n-1) \text{th stage}} + \underbrace{(6 \cdot I_S)}_{n \text{th stage}} = n \cdot 10 \cdot I_S. \quad (25)$$

For the single-stage MUX, the sizes of the clock buffer and data registers are 1/2 and 1/2ⁿ of the selector according to their loading effects and operation frequency, respectively. The number of data registers is 5 · 2ⁿ. Thus, the total current is

$$I_{\text{single}} = \underbrace{2^n \cdot \frac{I_S}{2}}_{\text{CK buf}} + \underbrace{5 \cdot 2^n \cdot \frac{I_S}{2^n}}_{\text{Data DFF}} + \underbrace{I_S}_{\text{MUX}} = (2^{n-1} + 6) \cdot I_S. \quad (26)$$

For the proposed MUX, the size scaling of all the selectors is similar with that for the standard tree-type MUX. The total current is

$$I_{\text{pro}} = \left[\underbrace{1}_{1 \text{st}} + \underbrace{\frac{1}{2^1} (2^0 + 2^1)}_{2 \text{nd}} + \underbrace{\frac{1}{2^2} (2^0 + 2^1 + 2^2)}_{3 \text{rd}} + \dots + \underbrace{\frac{1}{2^{n-1}} (2^0 + 2^1 + \dots + 2^{n-1})}_{n \text{th}} \right] \cdot I_S + \underbrace{2^n \cdot (5/2^n)}_{\text{Data DFF}} \cdot I_S. \quad (27)$$

Fig. 13 shows the SPICE simulation results of the current consumptions for the three MUX architectures. The numbers of inputs are 8, 16, and 32. The total current is dominated by the static current. Table II compares the currents obtained by analysis (25)–(27) and simulation. The results match well in all cases.

IV. IMPLEMENTATION AND MEASUREMENT

Fig. 14 shows the system architecture that has been implemented. An 8-bit linear feedback shift register is used as a random pattern generator. A self-biased phase-locked loop

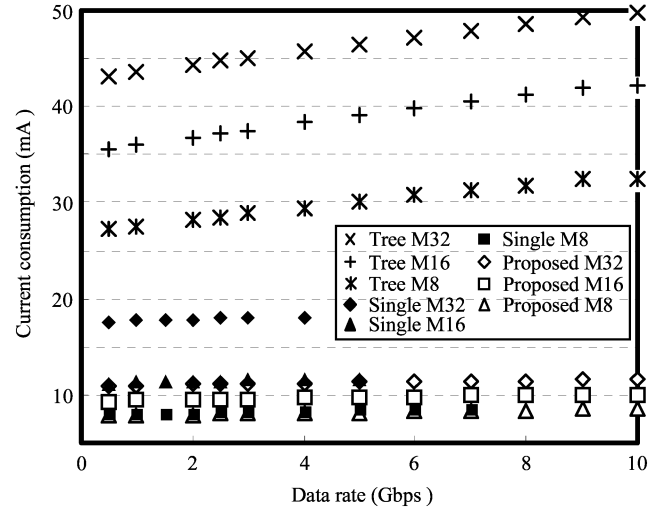


Fig. 13. Simulated current consumption.

TABLE II
MUX CURRENT CONSUMPTION

MUX Architecture	Simulation			Calculation		
	M32	M16	M8	M32	M16	M8
Standard tree @ 3G/10Gbps	45.1mA / 49.7mA	37.5mA / 42.2mA	28.8mA / 32.5mA	50	40	30
Proposed @ 3G/10Gbps	11mA / 11.6mA	9.3mA / 10mA	7.8mA / 8.5mA	13 $\frac{1}{16}$	11 $\frac{1}{8}$	9 $\frac{1}{4}$
Single stage @ 3Gbps	18mA	11.5mA	8.26mA	22	14	10

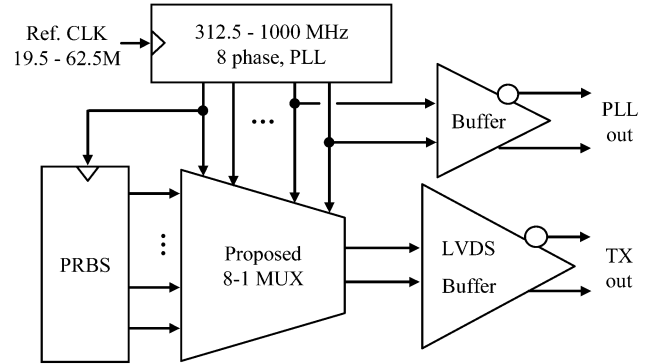


Fig. 14. Test chip architecture.

(PLL) [8] is used to generate eight-phase clock signals with a wide frequency range. The proposed MUX serializes 8-bit parallel single-end data into differential outputs with a data rate that is eight times the frequency of the PLL. For off-chip driving, two multistage current-mode buffers are inserted for the MUX and PLL, as shown in Figs. 15 and 16. The last stage is a low-voltage differential signaling (LVDS) driver [9]. The 50-Ω termination is achieved by a parallel connection of a 112-Ω on-chip ploy resistor and the 90-Ω turn-on resistance of the data switches of the LVDS driver.

The predriver outputs two pairs of differential signals to control the four data switches of the LVDS drivers. Since P- and N-type switches have different input capacitances, the

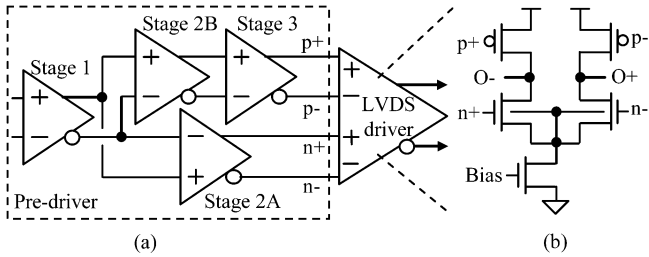


Fig. 15. (a) Building blocks of the LVDS buffer and (b) LVDS driver schematic.

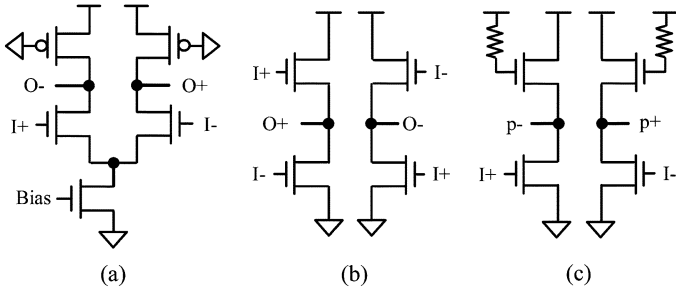


Fig. 16. Schematic of the predriver. (a) Stages 1 and 2A. (b) Stage 2B. (c) Stage 3.

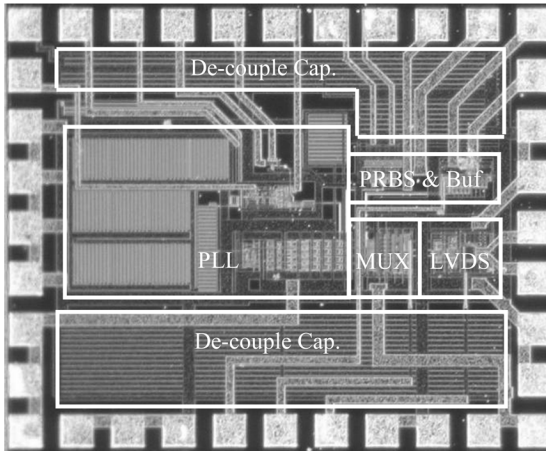


Fig. 17. Test chip photograph.

predrivers are organized differently, i.e., two stages for the N-type switches and three stages for the P-type ones, as shown in Fig. 15. Different predriver stages have different circuit diagrams to meet their function demands. Their circuit diagrams are shown in Fig. 16. Fig. 17 shows the chip photograph. It is fabricated in a TSMC 0.18- μm CMOS process. The PLL and the MUX occupy areas of 0.264 and 0.029 mm^2 , respectively. The measurement is performed on a PCB in Roger material. The Agilent 81130A generates the reference clock to the PLL, and the Agilent 11801C measures the eye diagrams. The measurement is focused on verifying the analysis and simulations on the output timing jitter of the proposed MUX at different data rates. Thus, the reference clock was swept from 19.53 to 62.5 MHz that allows the PLL to oscillate from 312.5 MHz to 1 GHz. As a result, the MUX will operate at a bit rate from 2.5 to 8 gigabits/s.

Fig. 18 shows the measured jitter at different data rates. The PLL and TX represent the jitters measured at the PLL output and

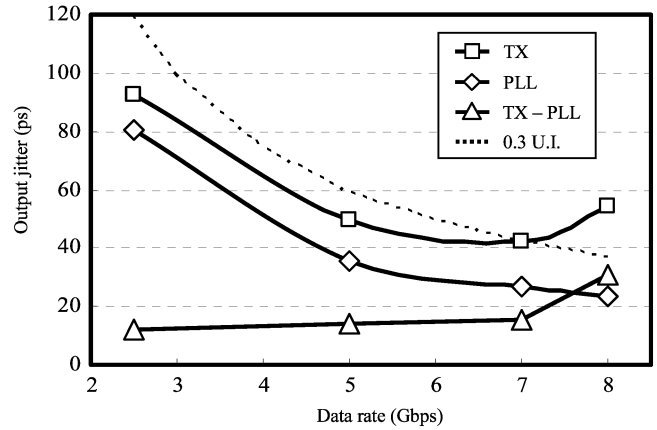


Fig. 18. Measured jitter at different data rates.

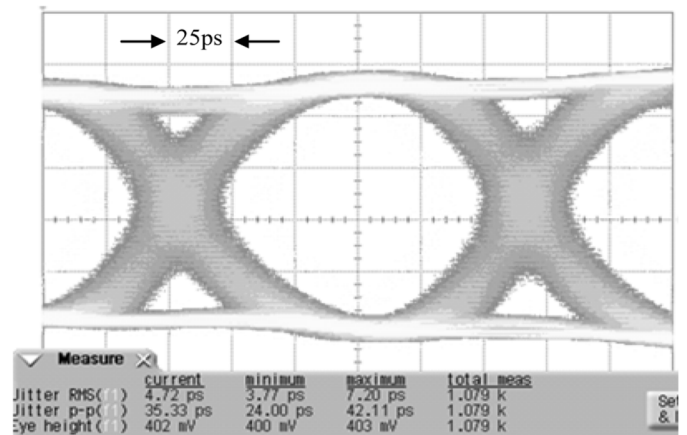


Fig. 19. Measured data-eye diagram at a bit rate of 7 gigabits/s (TX out).

the TX output, respectively. The dotted line represents a jitter limitation of 0.3 UI set by many serial I/O standards. As one can see, below 7 gigabits/s, the jitter is dominated by the PLL jitter. Normally, a ring-oscillator-type PLL has a higher jitter at low frequency. Above 7 gigabits/s, the jitter is dominated by the MUX. Such measured results match the simulated results shown in Fig. 11. Both indicate that above 7 gigabits/s, the jitter begins to rise exponentially due to ISI effects.

With the limitation of 0.3 UI, the maximal operation speed is 7 gigabits/s. Fig. 19 shows the output data-eye diagram at 7 gigabits/s. The data transition time is 70 ps, and the amplitude is 400 mV. Table III summarizes the performance of the test chip. The area and power consumption for the MUX, PLL, PRBS, and LVDS are listed individually. The jitters for the MUX and PLL are also individually listed. At 2.5 and 7 gigabits/s, the peak-to-peak jitters are 92.8 and 42.1 ps, or 0.24 and 0.29 UI, respectively.

V. CONCLUSION

In this paper, we have proposed a MUX in tree topology that uses a multiphase low-frequency clock which is normally applicable to single-stage MUXs only. The parasitic effects at each stage are minimized by multiplexing only two inputs. Therefore, the jitter caused by process variation and ISI is reduced, and the

TABLE III
PERFORMANCE SUMMARY

Function	Data transmitter with proposed tree-topology MUX	
Technology	0.18 μm 1P6M CMOS	
Voltage	1.8V	
Power @ 5Gbps	~18.68mW (PLL) ~39.54mW (LVDS)	~30.06mW (MUX) ~10.44mW (PRBS)
Jitter (pk-pk)	26.7ps @ 875MHz (PLL) 80.67ps @ 312.5MHz (PLL)	42.11ps @ 7Gbps (TX) 92.8ps @ 2.5Gbps (TX)
Jitter (rms)	4.5ps @ 875MHz (PLL) 13.01ps @ 312.5MHz (PLL)	7.2ps @ 7Gbps (TX) 15.26ps @ 2.5Gbps (TX)
Area	660 μm \times 400 μm (PLL), 165 μm \times 180 μm (MUX) 190 μm \times 180 μm (LVDS), 350 μm \times 100 μm (Buf, PRBS) 1280 μm \times 1040 μm (Chip)	

data rate is increased. This has been reassured by the mathematical analysis and the circuit-level simulation as well.

The proposed MUX, with PLL and LVDS drivers, has been designed and implemented in a TSMC 0.18- μm 1P6M CMOS process. It occupies an area of 165 μm \times 180 μm and consumes 30 mW of power at a data rate of 5 gigabits/s. It is able to operate up to 7 gigabits/s for a peak-to-peak jitter of 42.1 ps or 0.29 UI. Measured results, as well as simulated ones, suggest that the jitter is dominated by ISI effects when the data rate exceeds 7 gigabits/s. Otherwise, it is dominated by the PLL.

ACKNOWLEDGMENT

The authors would like to thank CIC for supporting the chip fabrication.

REFERENCES

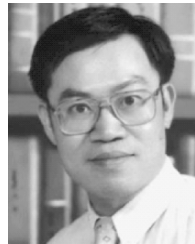
- [1] M. Ida, N. Kato, and T. Takada, "A 4 Gb/s GaAs 16:1 multiplexer/1:16 demultiplexer LSI chip," *IEEE J. Solid-State Circuits*, vol. 24, no. 4, pp. 928–932, Aug. 1989.
- [2] K. Lee, S. Kim, G. Ahn, and D.-K. Jeong, "A CMOS serial link for fully duplexed data communication," *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 353–364, Apr. 1995.
- [3] A. Maxim, B. Scott, E. Schneider, M. Hagge, S. Chacko, and D. Stuurca, "A low-jitter 125–1250-MHz process-independent and ripple-poleless 0.18- μm CMOS PLL based on a sample-reset loop filter," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1673–1683, Nov. 2001.
- [4] S.-J. Bae, H.-J. Chi, Y.-S. Sohn, and H.-J. Park, "A VCDL-based 60–760-MHz dual-loop DLL with infinite phase-shift capability and adaptive-bandwidth scheme," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1119–1129, May 2005.

- [5] J. L. Zerbe *et al.*, "Equalization and clock recovery for a 2.5–10-Gb/s 2-PAM/4-PAM backplane transceiver cell," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2121–2130, Dec. 2003.
- [6] K.-Y. K. Chang, J. Wei, C. Huang, S. Li, K. Donnelly, M. Horowitz, L. Yingxuan, and S. Sidiropoulos, "A 0.4–4-Gb/s CMOS quad transceiver cell using on-chip regulated dual-loop PLLs," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 747–754, May 2003.
- [7] M.-J. E. Lee, W. J. Dally, and P. Chiang, "Low-power area-efficient high-speed I/O circuit techniques," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, Nov. 2000.
- [8] J. G. Manteatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1723–1732, Nov. 1996.
- [9] M. Chen, J. Silva-Martinez, M. Nix, and M. E. Robinson, "Low-voltage low-power LVDS drivers," *IEEE J. Solid-State Circuits*, vol. 40, no. 2, pp. 472–479, Feb. 2005.



Hungwen Lu received the B.S. degree in electronic engineering from National Central University, Jhongli, Taiwan, in 2001, where he is currently working toward the Ph.D. degree in the Department of Electrical Engineering.

His research interests include high-speed interconnect design and mixed-signal circuit design.



Chauchin Su (M'90) received the B.S. and M.S. degrees in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1979 and 1981, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 1990.

Since graduation, he has been with the Department of Electrical and Control Engineering, National Chiao Tung University. His research interests include mixed-analog and digital-system testing and design for testability. He is also involved in projects

on baseband and circuit design for wireless communication.



Chien-Nan Jimmy Liu (M'03) received the B.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan.

He is currently an Associate Professor with the Department of Electrical Engineering, National Central University. His research interests include behavioral modeling for analog/mixed-signal designs, high-level power and noise modeling, and functional verification for HDL designs.

Dr. Liu is a member of Phi Tau Phi.