A 1.7 Gbps DLL-Based Clock Data Recovery for a Serial Display Interface in 0.35-μm CMOS

Yong-Hwan Moon, Sang-Ho Kim, Tae-Ho Kim, Hyung-Min Park, and Jin-Ku Kang

This paper presents a delay-locked-loop–based clock and data recovery (CDR) circuit design with a *n*B(*n*+2)B data formatting scheme for a high-speed serial display interface. The *n*B(*n*+2)B data is formatted by inserting a '01' clock information pattern in every piece of *N*-bit data. The proposed CDR recovers clock and data in 1:10 demultiplexed form without an external reference clock. To validate the feasibility of the scheme, a 1.7-Gbps CDR based on the proposed scheme is designed, simulated, and fabricated. Input data patterns were formatted as 10B12B for a high-performance display interface. The proposed CDR consumes approximately 8 mA under a 3.3-V power supply using a 0.35-μm CMOS process and the measured peak-to-peak jitter of the recovered clock is 44 ps.

Keywords: Clock data recovery (CDR), delay-locked loop (DLL), *n*B(*n*+2)B data formatting scheme, high-speed serial interface, display interface.

I. Introduction

In high-speed serial interface applications, phase-locked loop (PLL)-based clock and data recovery (CDR) is mostly used [1]. However, the PLL-based CDR consumes considerable power and generates accumulated jitter due to the voltage-controlled oscillator. Furthermore, the conventional PLL-based CDR requires an 8B10B encoding scheme for providing DC balance and ample transition edges for clock recovery. Therefore, at the receiver side, 8B10B decoding, comma detection, and word alignment blocks are required. On the other hand, the delay-locked loop (DLL)-based CDR usually consumes less power than the PLL-based CDR. In addition, the voltage-controlled delay line (VCDL) in the DLL-based CDR does not accumulate jitter. However, the conventional DLL-based CDR design requires an external reference clock line [2]-[4].

The proposed DLL-based CDR uses a single data line encoded by the *n*B(*n*+2)B data formatting scheme with embedded clock information for a display interface [5]. The proposed CDR operates on the training data patterns and real input data patterns with the 10B12B data formatting scheme. The proposed *n*B(*n*+2)B data formatting scheme is realized by inserting a '01' clock information pattern for every piece of *N*-bit input data. The proposed 10B12B data formatting scheme has a 17% coding overhead. Compared to the 8B10B, the encoding scheme has a 20% coding overhead. Hence, the proposed 10B12B data formatting scheme transmits a higher data rate than the conventional 8B10B encoding scheme. In addition, the proposed *n*B(*n*+2)B data formatting scheme requires a simple word alignment operation. The proposed scheme thus has a simpler architecture and less power consumption than the conventional 8B10B encoding scheme.

This paper first presents the architecture and operation

Manuscript received Mar. 8, 2011; revised June 17, 2011; accepted July 1, 2011.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0026743).

Yong-Hwan Moon (phone: +82 32 860 8721, moonyong@inha.edu), Tae-Ho Kim (corresponding author, taeho.kim.inha@gmail.com), and Jin-Ku Kang (jkang@inha.ac.kr) are with the School of Electronics Engineering, Inha University, Incheon, Rep. of Korea.

Sang-Ho Kim (sharpksh@gmail.com) is with Siliconworks Co., Ilsan, Gyeonggido, Rep. of Korea.

Hyung-Min Park (hyungmin@lginnotek.com) is with LG Innotek Co., Ansan, Gyeonggido, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0143

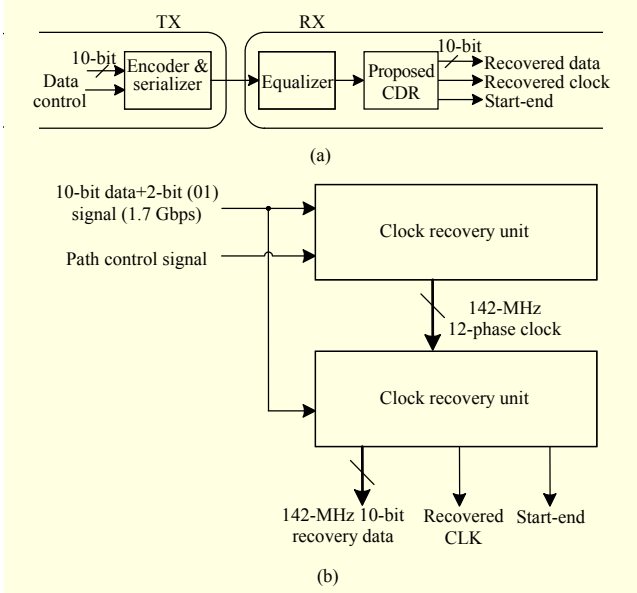


Fig. 1. Block diagram of (a) proposed serial interface and (b) proposed DLL-based CDR.

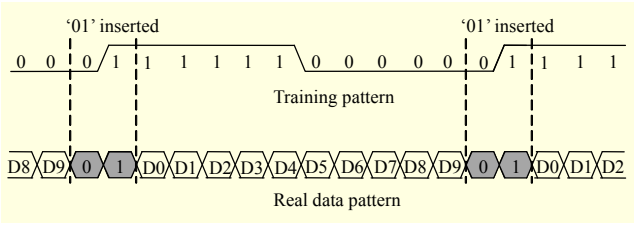


Fig. 2. Formatted input pattern.

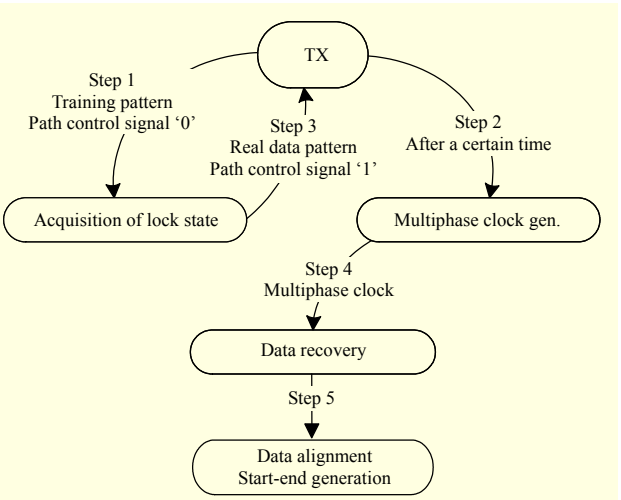


Fig. 3. Flow chart of proposed CDR.

principle. Details of the proposed clock recovery unit are then explained. The circuit design and measurement results follow.

II. Architecture and Operation

Figure 1(a) shows the proposed serial interface block and Fig. 1(b) presents a block diagram of the proposed DLL-based CDR. The transmitter (TX) block has an encoder and a serializer. The main block of the receiver (RX) includes the proposed CDR. The CDR has a clock recovery unit and a data recovery block. In the proposed 1.7-Gbps DLL-based CDR design, the training data and real data patterns are formatted by inserting '01' in every piece of 10-bit input data. In other words, the value '*n*' of the proposed *n*B(*n*+2)B data formatting scheme is 10. The reason why we choose the 10B12B data formatting is that the circuit is targeted for high-definition display interface applications with a 10-bit gray level [6], [7]. The proposed CDR recovers the clock from the inserted clock information '01' on the *n*B(*n*+2)B formatted serial data. Since the input data is formatted by inserting the '01' pattern in every *N*-bit, the number of multiphase clocks to be generated is *N*+2. For 1.7-Gbps CDR design with 10B12B data formatting, the clock recovery unit needs to generate 142-MHz 12 multiphase clocks (phase difference = 588 ps = 1-bit period of 1.7 Gbps). The data recovery block retimes the data using the generated 12 multiphase clocks in the clock recovery unit.

The 10B12B formatted data patterns for the proposed CDR are shown in Fig. 2. The training pattern serves as a reference clock for the DLL-based CDR. The duty ratio of the training pattern is not a concern as long as the pattern provides a signal transition. The encoded data is transmitted at 1.7 Gbps for handling the actual 1.42-Gbps data rate because the coding overhead of the proposed 10B12B data formatting scheme is 17%. The proposed 10B12B data formatting scheme can be expanded to *n*B(*n*+2)B. If *n* is increased, the data formatting overhead is decreased. However, the number of delay cells in the VCDL increases, and the power consumption and the jitter can be affected. Depending on the targeted high-speed interfaces, the demultiplexing structure, the recovered clock rate, the number of multiphase clocks, and the input data rate can be varied to obtain optimal design conditions.

Figure 3 shows the overall operation flow chart. The input data transmission is accomplished with two operation steps and each step sends different data patterns. At the beginning of data transmission, the TX sends the formatted training pattern for a certain period. During the training period, the proposed CDR in the RX circuit is virtually operating as a DLL. The proposed CDR thus generates a recovered clock and acquires the lock state in the training pattern period. After the TX sends the training data pattern, the TX switches to send the real data pattern. During the real data pattern period, the CDR continues to recover the clock by tracking the inserted '01' pattern, and retimes the data. The path control signal is a control signal

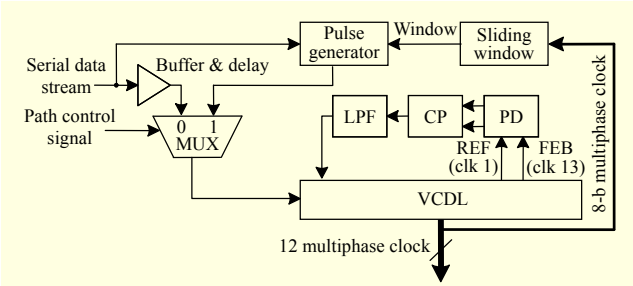


Fig. 4. Block diagram of clock recovery unit

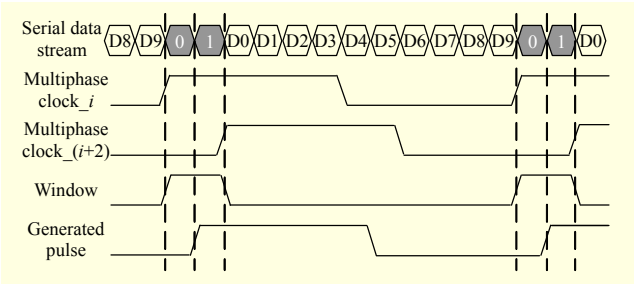


Fig. 5. Ideal timing diagram of window and pulse generator under lock state.

generated in the TX block. The path control signal distinguishes the training data pattern period from the real data transmission period. The TX sends the training data pattern and real data transmission, while the path control signal is '0' and '1,' respectively. Since the proposed DLL-based clock recovery targets high-speed digital display interfaces, the training sequence bits can be transmitted during vertical and horizontal blanking periods. These blanking periods are much longer than the required minimum training sequence period. In other serial link applications, the optimized training sequence period can be as short as possible.

The complete transceiver system based on the proposed DLL-based CDR should have a loss of lock (LOL) generated from the RX and control the training/data mode in the TX. If the TX receives the LOL signal from the RX, the TX should start to send the training bits to the RX. Hence, the lock detection function block in the RX should be added for that control, but the function is not included in this prototype.

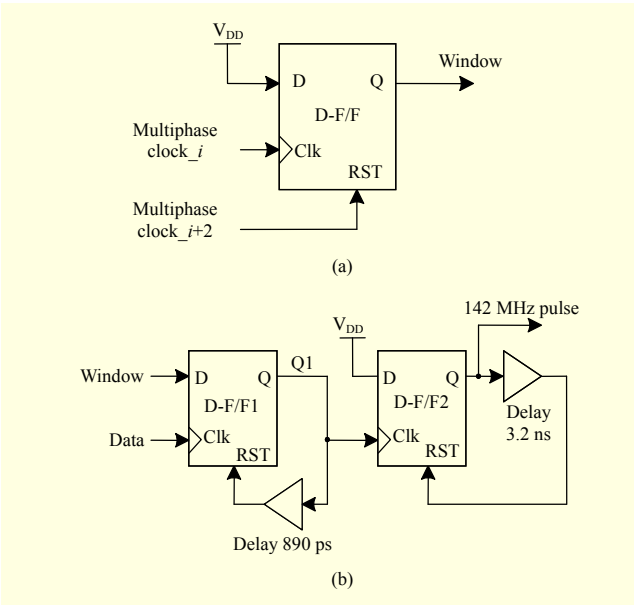


Fig. 6. Block diagram of (a) window generator and (b) pulse generator.

III. Clock and Data Recovery Unit

1. Clock Recovery Unit

Figure 4 shows the clock recovery unit of the proposed CDR. While the TX is sending the training data pattern, the path control signal connects the serial data stream (training data pattern) to the VCDL. The clock recovery unit then operates as an ordinary DLL. The training pattern acts as a 142-MHz reference clock signal. The clock recovery unit acquires the lock state through the DLL operation. Although the DLL goes to the lock state, the path maintains the pure DLL operation while the path control signal remains at '0.' After the training pattern period, the TX transmits the real data pattern and the path control signal goes to '1.' The MUX then selects a signal from the pulse generator. The pulse generator block extracts the clock and continues tracking the clock phase by the inserted '01' pattern. The window signal is produced for masking random data patterns by extracting only the signal transition ('01' clock information) information. While the encoded random input data patterns are provided, the clock recovery unit keeps extracting the transition information and generates the clock signal. The clock recovery unit produces 12 multiphase clocks by the VCDL. The data recovery block recovers the data using the 12 multiphase clocks.

Figure 5 depicts the ideal timing diagram of the window and the pulse signal under the lock state. The window signal is generated using the two multiphase clocks (11th and 1st clock signals) by tapping two signals from the VCDL. The window generator and the pulse generator are shown in Fig. 6. The window generator can be realized by a single D flip-flop (D-F/F) with a reset. The window signal marches with the inserted '01' pattern on the data stream. The pulse generator generates a 142-MHz clock by a combination of the window signal and the inserted '01' clock information. The pulse generator consists of two D-F/Fs and two delay cells. The first D-F/F of the pulse generator creates roughly 0.9-ns pulse width by both the window signal and the positive edge of the inserted '01' pattern every 7 ns (period of 142 MHz). The pulse

becomes enlarged to a 3.2-ns pulse width by the second D-F/F. However, if the delay is increased to more than 3.5 ns, the next positive edge will be not generated by the feedback reset. Considering process, supply voltage, and temperature (PVT) variation for safe operation, the delay is set as 3.2 ns (+300 ps timing margin for PVT variation). The optimal pulse is hence generated with a 45% duty cycle. The duty ratio is not critical in this design, because the data recovery is accomplished with only the rising edge of the clock. The critical point here is precise window position control relative to the inserted '01' position in the encoded input data. To ensure that the window is positioned in the correct place, a 'sliding window scheme' is implemented, as shown in Fig. 7.

The window sliding scheme generates four different windows for capturing the inserted '01' pattern under PVT variation. The four windows can be shaped by different combinations of multiphase clocks from DLL. The prototype has four window choices, which are the expected window, +/– shifted windows, and an enlarged window from the optimally expected window. The final window is selected by measuring the relative position between the inserted '01' pattern and the windows. The control pins (s1, s0) are placed externally in this design stage. An adaptive window selecting scheme can be implemented with an extra logic. One possible solution is that the inserted '01' transition timing is oversampled by the finer clocks (sampling resolution of about 120 ps) and the window selection can be made by the relative positioning of the '01' transition based on sampled bits. The finer clocks should be generated with finer delays.

In the proposed 10B12B encoding scheme, fixed '01' patterns are inserted into the data stream. Thus, in the frequency domain analysis, there could be some peaks in the spectrum that may cause an electromagnetic interference (EMI) problem in system integration. The spread spectrum clock generator can therefore be used to reduce the EMI problem in the TX. If the transmitted data is spread spectrum data, the CDR should have a proper bandwidth for handling the spread spectrum data.

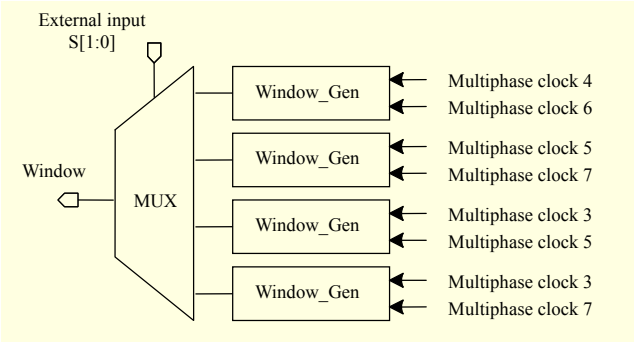


Fig. 7. Block diagram of sliding window scheme.

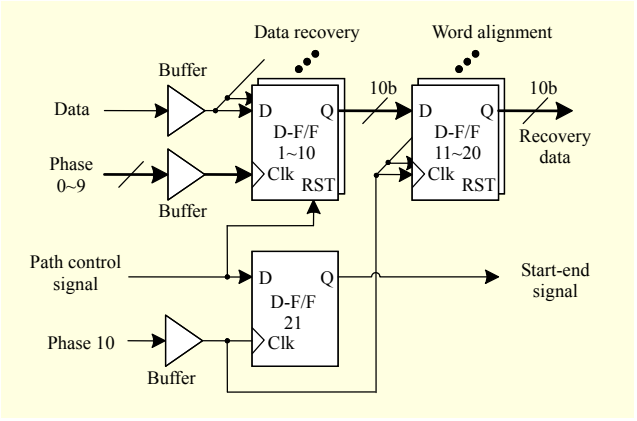


Fig. 8. Block diagram of data recovery unit.

2. Data Recovery Unit

Figure 8 shows a block diagram of the data recovery block. The data recovery block consists of D-F/Fs and buffers. The proposed data recovery starts to recover the data when the path control signal goes to '1.' The proposed data recovery block recovers the 10-bit real data in parallel at 142 Mbps using 10 different phase clocks (@142 MHz) of the generated 12 multiphase clocks from the clock recovery unit. D-F/Fs 1 to 10 recover each data and D-F/Fs 11 to 20 align the recovered data at phase 10. D-F/F 21 generates the start-end signal, which indicates the start and end of recovered real data. The start-end signal is '0' in the training pattern period. If the start-end signal is '1,' then the proposed CDR recovers the data during the real data pattern period. The inserted '01' clock information pattern is not recovered. Thus, the word alignment scheme is easily implemented without an additional logic compared to conventional 8B10B word alignment.

IV. Circuit Design

1. DLL Design

A DLL circuit used in the clock recovery unit consists of a phase detector (PD), a loop filter, and the VCDL.

Figure 9 shows the PD, which is a conventional phase frequency detector with an extra D-F/F and a XOR for excluding the first reference clock edge to be compared with the first feedback clock edge. The last D-F/F (the bottom one) prevents the false lock operation and is a modified version of [8]. The total delay of the VCDL is still designed between $0.5T$ and $1.5T$, where T is the clock period for faster locking.

The unit cell of the VCDL is shown in Fig. 10(a) and Fig. 10(b) shows a block diagram of the VCDL in the DLL. The unit cell of the VCDL is based on a current-starved inverter structure for its simplicity and small area occupation. For insensitivity to the supply noise, the unit cell has a cascoded

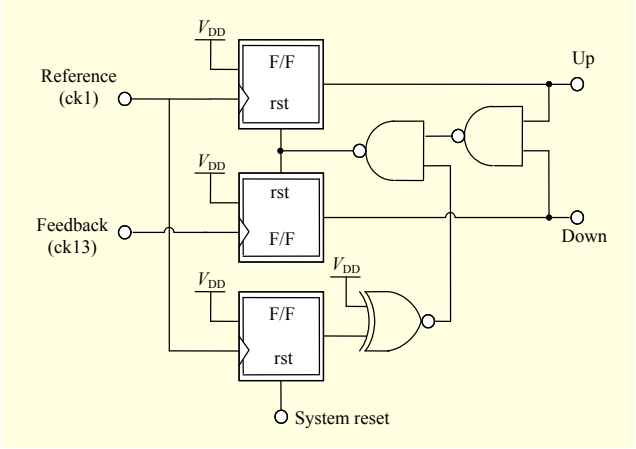


Fig. 9. Circuit diagram of phase detector.

current source structure. It has four bias voltages and one voltage control (VC) node for delay adjustment. Each delay stage consists of two inverter cells. Under three different operating conditions, the simulated delay versus control voltage for the VCDL (composed by 26 delay cells and 13 delay stages) is demonstrated in Fig. 11. The total delay is set around between 6 ns and 8.5 ns at a control voltage range of 0.5 V to 2.5 V on the typical operating condition.

Corner simulations were carried out for assuring the targeted delay coverage under PVT variations. As shown in Fig. 11, under three process corners, the targeted total delay of 7 ns could be covered with a control voltage between 1.1 V and 1.99 V. The delay value of the VCDL is 7.8 ns for the slow condition and 6.4 ns for the fast condition, respectively, at a control voltage of 1.4 V. Thus, the total delay variations of the VCDL are +800 ps and –600 ps under two extreme operating conditions. Assuming the variations are distributed to each delay stage, the maximum deviations of each delay stage are +61.54 ps and –46.16 ps. The maximum interstage mismatch can then be about 108 ps. Considering the ideal spacing between multiphase clocks is 588 ps (one bit data period of 1.7 Gbps data), the maximum predictable clock phase variation of the interstage is about 18% of the clock timing margin. Thus, there is an ample timing margin for data recovery with input data jitter included. The measured values are also well matched to simulation as shown in Fig. 11.

In a liquid crystal display (LCD) interface, high power supply noise is produced by the amplifier in the high-voltage LCD drivers. Therefore, the proposed CDR should recover the clock robustly under the power supply noise. The proposed CDR used a bias circuit highly insensitive to the power supply noise [9].

2. Equalizer Circuit

At higher data rates, the intersymbol interference (ISI)

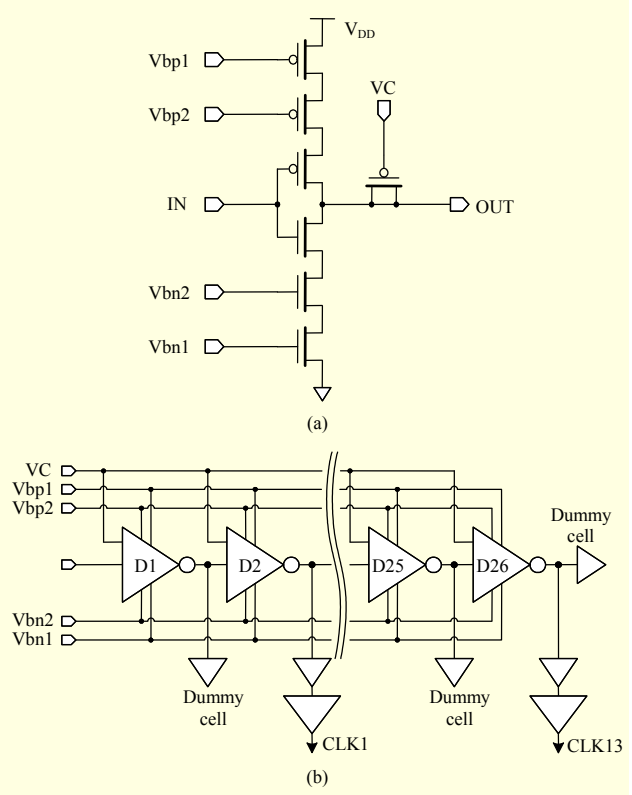


Fig. 10. Circuit diagram of (a) unit cell and (b)VCDL.

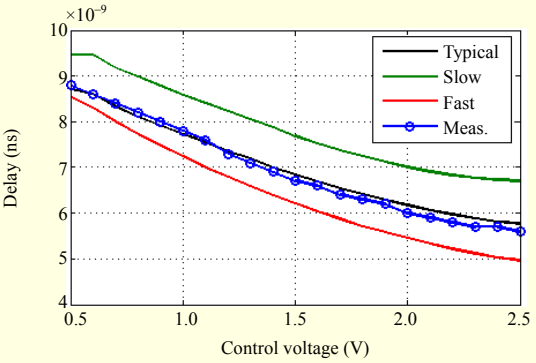


Fig. 11. Measured and simulated total delay of VCDL.

corruption caused by the skin effect and the dielectric loss of the electrical channel is exacerbated. To overcome this problem, channel modeling is done and the channel loss should be compensated by the equalizer. In our test environment, the channel is a 30-inch-long trace with FR-4 and the insertion loss was measured as –10 dB at 1.2 GHz.

An analog equalizer with capacitance degeneration is added to the RX to compensate the high frequency loss caused by the channel. The amount of compensation level is controlled up to 10 dB with a 3-bit external control signal (EQ_CTRL). The block diagram and circuits of the equalizer are shown in Fig. 12.

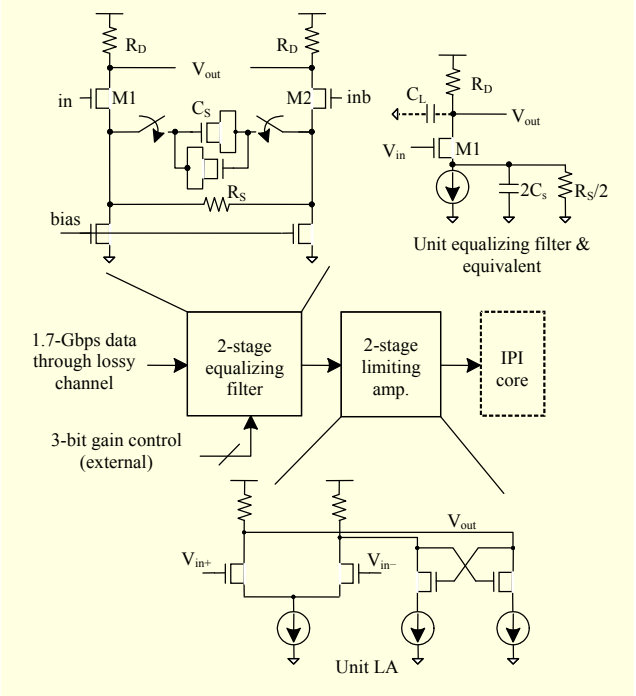


Fig. 12. Block diagram and circuits of equalizer.

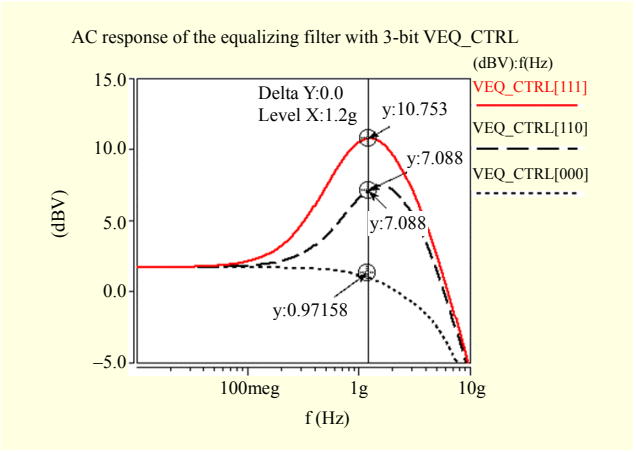


Fig. 13. AC response of equalizer with EQ_CTRL value.

The equalizer's compensation level is increased as the EQ_CTRL[2:0] value increases. The compensation level is as high as 10 dB with EQ_CTRL[2:0]='111.' Figure 13 shows the simulated compensation level with various EQ_CTRL values.

V. Measurement Results

The proposed 1.7-Gbps DLL-based CDR with 10B12B data formatting was fabricated in a 3.3-V, 0.35-μm CMOS process. Figure 14 shows the core layout and the chip microphotograph of the proposed CDR. The area of the CDR core is 520 μm × 253 μm. The CDR core consumed approximately 8 mA, and

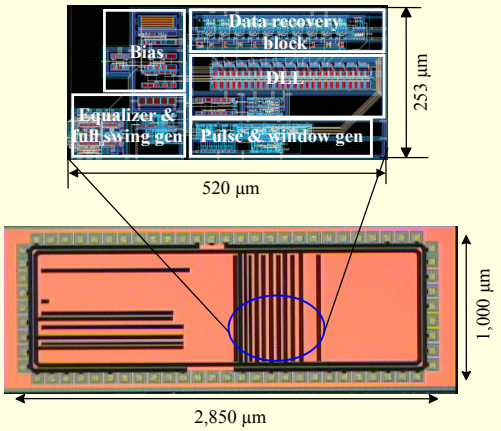


Fig. 14. Core layout and chip microphotograph.

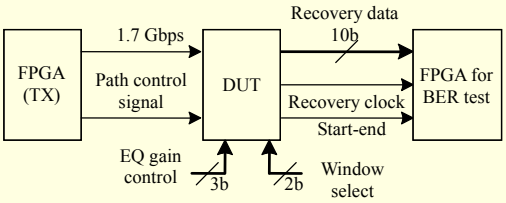


Fig. 15. Measurement setup.

the whole chip including the output buffer consumes 34 mA at a 1.7-Gbps input data rate under 3.3-V supply voltage. The device under test (DUT) is implemented with a chip-on-board on the FR4 printed circuit board.

Figure 15 shows the measurement environment for bit error rate (BER) test. The proposed CDR was tested with the field-programmable gate array (FPGA) board for generation of encoded patterns. The FPGA board, TX, transmits the training pattern and the real data pattern. Measurement shows that the operating data rate is between 1.2 Gbps and 1.73 Gbps. Table 1 shows several test data patterns of the 10-bit real data and clock information. The FPGA board serializes the 10-bit data with '01' clock information and transmits the serial data to the CDR. The proposed CDR requires two types of data pattern, training and real data. The proposed CDR recovers the clock and data during the real data pattern. The BER test was executed for the real data pattern generated by the pseudo random bit sequence (PRBS) in the FPGA. With 2^{31}–1 random patterns, no error was detected for a day operation. As shown in Fig. 17, BER test was measured through FPGA's on the TX/RX sides. Since the signal patterns in the transmitter were generated by FPGA programming, the jitter modulation source for measuring jitter tolerance could not be incorporated in the test setup. Therefore, jitter tolerance performance could not be measured. Since the locking time of the DLL operation is measured as about 420 ns, the training period must be longer than 420 ns. Thus, about 720

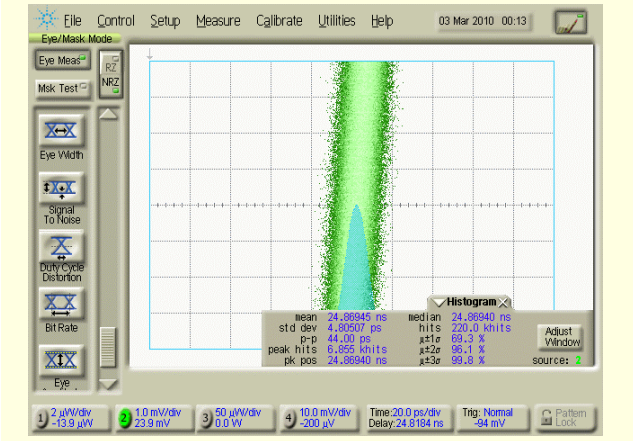


Fig. 16. Measured jitter of recovered clock.

bits are required for the minimum training period for a 1.7 Gbps input rate.

The 38-ps peak-to-peak jitter at 1.2-Gbps data rate and 44-ps peak-to-peak jitter at 1.7 Gbps were measured, respectively. Figure 16 shows the root-mean-square (RMS) jitter and the peak-to-peak jitter of the 142-MHz recovered clock with the 1.7-Gbps real data pattern. The RMS jitter and the peak-to-peak jitter of the recovered clock are measured as 4.8 ps and 44 ps, respectively. From the test results on the sliding window scheme, the expected optimal window position shows better results than +/– shifted windows, and the extended window pulse shows the best data recovery performance. This explains why the generated widow pulse width is shortened slightly due to the RC effect on the line.

In Table 1, several input test patterns are given to check for a possible ISI effect. To investigate the effect, the prototype circuit includes an equalizer at the input. The input path can be connected to the input buffer directly or through the equalizer by a digital switch. The ISI effect can be evaluated by measuring the data performance with the equalizer and without the equalizer. Under severe ISI input data conditions, the CDR with the equalizer and without the equalizer recovers the clock up to the 20-inch trace line. However, in the 30-inch trace line, only the CDR core with equalizer recovers the clock correctly. Thus, the test indicates that the equalizer should be 'on' at full strength (set '111' for VEQ_CTRL bits) for error free operation in the longer trace line.

Figure 17 shows the recovered data in data channel 1 and data channel 7 among 10 data channels in parallel at 142 Mbps for input patterns in Table 1 with the 30-inch FR4 trace line and the equalizer. The eye pattern of the output is shown in Fig. 18. The CDR operates the power supply voltage in a range from 2.9 V to 3.5 V. A performance summary of the proposed CDR is given in Table 2, and a comparison with [1] and [10] is given in Table 3. The summary illustrates the advantages that the proposed CDR provides in terms of power consumption and jitter performance.

Table 1. Several test patterns for checking worst case ISI.

No.	Clock inform.		D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
1	0	1	1	1	1	1	1	1	1	1	1	1
2	0	1	0	1	1	0	1	1	0	0	1	1
3	0	1	0	1	1	1	0	1	0	0	0	1
4	0	1	0	0	0	0	0	1	0	0	0	0
5	0	1	0	0	0	1	1	1	0	1	0	1
6	0	1	0	1	1	1	0	0	0	0	1	0
7	0	1	1	1	1	0	0	1	0	1	1	1
8	0	1	0	0	0	1	0	0	0	1	0	1
9	0	1	1	1	0	0	0	1	1	1	0	1
10	0	1	0	0	0	0	1	1	0	0	1	1

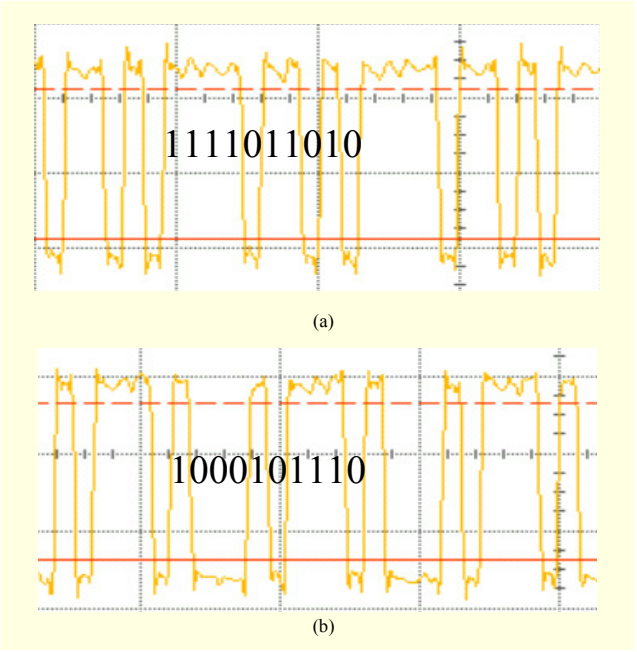


Fig. 17. Recovered data in (a) data channel 1 and (b) data channel 7.

VI. Conclusion

A DLL-based CDR has a design based on the 10B12B formatting scheme for application in high-performance display interfaces. The proposed data formatting scheme is realized by inserting a '01' pattern at every 10-bit input. The proposed CDR recovers the clock and data in a 1:10 demultiplexed manner without an external reference clock. The CDR consumes approximately 8 mA at 1.7-Gbps input data under a

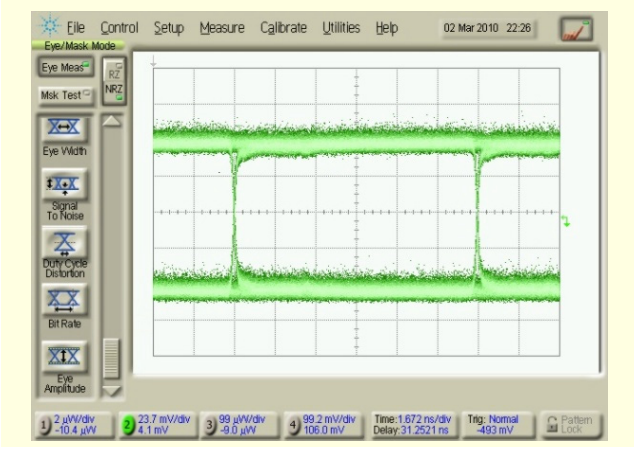


Fig. 18. Eye diagram of recovered data (@142 Mbps).

Table 2. Measured performance summary.

Specification	Results
Technology	0.35-μm CMOS process
Supply voltage	3.3 V
Chip area (only core)	520 μm × 253 μm
Operating data rate	1.2 Gbps to 1.73 Gbps
Jitter of recovered clock	44.00-ps peak-to-peak @1.7 Gbps 4.80-ps RMS @1.7 Gbps
Power supply voltage range	2.9 V to 3.5 V
Power consumption	8 mA @3.3 V without output buffer 34 mA @3.3 V with output buffer
Locking time	420 ns

Table 3. Performance comparison summary.

Specification	Proposed	[1]	[10]
Technology	0.35-μm CMOS process	0.28-μm CMOS process	0.25-μm CMOS process
Supply voltage	3.3 V	3 V	2.5 V
Chip area (only core)	520 μm × 253μm	1,000 μm × 400 μm	2 mm^2
Power consumption	1 mA (DLL)+ 3 mA (data rec.)+ 0.5 mA (pulse gen.)+ 0.2 mA (bias)+ 3.3 mA (equalizer) @ 3.3 V	67 mA @3 V	137 mA @2.5 V (total)
Jitter of recovered clock	4.80-ps RMS @1.7 Gbps	11-ps RMS @2 Gbps	120.1-ps peak-to-peak @1.82 Gbps
Coding scheme	10B12B	8B10B	24B28B
Basic structure	DLL-based CDR w/o reference clock	PLL-based CDR	PLL-based CDR

3.3-V power supply using a 0.35-μm CMOS process. The RMS jitter and the peak-to-peak jitter of the recovered clock are measured as 4.8 ps and 44 ps, respectively.

Acknowledgment

The authors also thank the IDEC program and for its hardware and software assistance for the design and simulation.

References

[1] K. Yamguchi et al., "A 2.0 Gb/s Clock-Embedded Interface for Full-HD 10-Bit 120-Hz LCD Drivers with 1/5-Rate Noise-Tolerant Phase and Frequency Recovery," *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 192-194.

[2] C.N. Chuang and S.I. Liu, "A 3-8 GHz Delay-Locked Loop With Cycle Jitter Calibration," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 11, Nov. 2008, pp. 1094-1098.

[3] X. Maillard, F. Devisch, and M. Kuijk, "A 900-Mb/s CMOS Data Recovery DLL Using Half-Frequency Clock," *IEEE J. Solid-State Circuits*, vol. 37, no. 6, June 2002, pp. 711-715.

[4] A.L. Coban, M.H. Koroglu, and K.A. Ahmed, "A 2.5-3.125-Gb/s Quad Transceiver With Second-Order Analog DLL-Based CDRs," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, Sept. 2005, pp. 1940-1947.

[5] T.-H. Kim, S.-H. Kim, and J.-K. Kang, "A DLL-Based Clock Data Recovery with a Modified Input Format," *IEICE Electron. Express (ELEX)*, vol. 7, no. 8, Apr. 2010, pp. 539-545.

[6] C.S. Jang et al., "An Intra Interface of Flat Panel Displays for High-End TV Applications," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, Aug. 2008, pp. 1447-1452.

[7] R.I. McCartney and M.J. Bell, "A Third Generation Timing Controller And Column-Driver Architecture Using Point-to-Point differential Signaling," *J. Soc. Inf. Display*, vol. 13, no. 2, Feb. 2005, pp. 91-97.

[8] C.H. Kim et al., "A 64-Mbit, 640-MByte/s Bidirectional Data Strobed, Double-Data-Rate SDRAM with a 40-mW DLL for a 256-MByte Memory System," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, Nov. 1998, pp. 1703-1710.

[9] D.A. Johns, K. Martin, *Analog Integrated Circuit Design*, New York: John Wiley & Sons, Inc., 1997, p. 259.

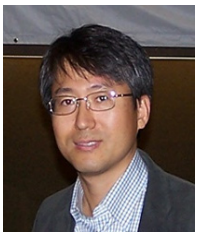
[10] I. Jung et al., "A 140-Mb/s to 1.82-Gb/s Continuous-Rate Embedded Clock Receiver for Flat-Panel Displays," *IEEE Trans. Circuits Syst. II, Exp. Briefs,* vol. 56, no. 10. Oct. 2009, pp. 773-777.

Yong-Hwan Moon received the BS and MS in electrical engineering in 2002 and 2004, respectively, from Inha University, Incheon, Rep. of Korea. He is currently working toward the PhD in the Department of Electrical and Engineering, Inha University. His research interests include high-speed interface and clock and data recovery circuits.

Sang-Ho Kim received the BS and MS in electronics engineering from Inha University, Incheon, Rep. of Korea, in 2007 and 2010, respectively. Since 2010, he has been with Siliconworks Co., Ilsan, Rep. of Korea. His research interests include high-speed clock and data recovery and VLSI circuit design.

Tae-Ho Kim received the BS and MS in electronics engineering from Inha University, Incheon, Rep. of Korea, in 2007 and 2009, respectively. He is currently working toward the PhD at the same university. His research interests include mixed-mode high-speed serial link interface and signal integrity.

Hyung-Min Park received the BS and MS in electronics engineering from Inha University, Incheon, Rep. Korea, in 2009 and 2011, respectively. Since 2011, he has been with LG Innotek Co., Ansan, Rep. of Korea. His research interests include high-speed clock generation and VLSI circuit design.



Jin-Ku Kang received his PhD in electrical engineering from North Carolina State University in 1996. From 1983 to 1988, he worked at Samsung Electronics, Inc., Korea. From 1996 to 1997, he was with Intel as a senior design engineer. Since 1997, he has been a professor in the School of Electronics Engineering, Inha University, Rep. of Korea. His research interests are high-speed CMOS VLSI design, mixed-mode IC design, and high-speed serial interface design.