A Referenceless All-Digital Fast Frequency Acquisition Full-Rate CDR Circuit for USB 2.0 in 65nm CMOS Technology

Ching-Che Chung and Wei-Cheng Dai

Department of Computer Science and Information Engineering

National Chung Cheng University

No. 168 University Rd, Ming-Hsiung, Chia-Yi, Taiwan, R.O.C.

Email: wildwolf@s3lab.org

Abstract −An all-digital fast frequency acquisition full-rate clock and data recovery (CDR) circuit for USB 2.0 applications without a reference clock is presented in this paper. The proposed digitally controlled oscillator (DCO) with an embedded time-to-digital converter (TDC) can recover the frequency of the synchronous data pattern in a very short time. In addition, the whole frequency acquisition can be finished within 31 cycles. A dual mode phase and frequency detector (PFD) is proposed to perform phase and frequency tracking with random data pattern to maintain the frequency and phase of the recovery clock. The proposed CDR circuit can operate at 480MHz for the USB 2.0 high-speed mode. The proposed CDR circuit can tolerance input data jitter up to 150ps with the bit error rate less than 10^{-12}. The proposed CDR circuit is implemented in a standard process 65nm CMOS process, the core area is $150\mu m \times 150\mu m$, and the power consumption is 1.75mW (@480MHz).

Index Terms— clock and data recovery, clocks, synchronization, digital_phase_locked_loops, jitter, oscillator.

I. INTRODUCTION

The clock and data recovery (CDR) circuit is the key component in high-speed serial data link. A CDR circuit has to generate a synchronized clock to the incoming serial data and recovers the data pattern. To achieve higher data rate and lower bit error rate (BER), the charge-pump based phase-locked loops (PLLs) with oversampling architecture is proposed in [1]. Fig. 1 shows the oversampling CDR circuit. The PLL generates multi-phase clocks (Ckout[n:0]) from an external reference clock (Ref_Clk). Then, the CDR loop recovers clock and data of the input data stream (Data) with an over-sampling phase detector (PD). The oversampling PD controls a low gain charge-pump (CP2) to continue tracking the phase between the input data steam (Data) and the output clock using the information of the oversampled data. However, this architecture needs an off-chip crystal oscillator as a precisely timing reference. The need for external component increases the cost of the design and the complexity of system integration. In addition, the power consumption is also increased. As a result, the CDR circuit without a reference clock [2-5] becomes more attractive in today's system-on-a-chip (SoC) era.

However, the referenceless CDR circuit with a charge-pump based control loop [2, 3, 5] suffered from the long time unilateral frequency tracking problem, so that the frequency tracking can not be finished within a short synchronization pattern. The CDR circuit should have fast frequency acquisition ability, especially for the USB 2.0 high-speed mode, there are only 31 cycles synchronization patterns in front of the random data. If the frequency error of the

recovery clock can not be reduced to a certain small range before the random data input, the CDR circuit will easily lose lock with input data jitter.
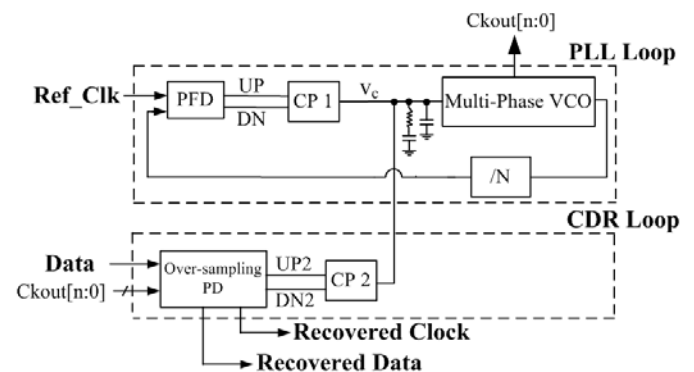

FIGURE 1. Oversampling CDR circuit.

In this paper, the proposed all-digital CDR circuit uses a time-to-digital converter (TDC) to measure the symbol rate of the input data stream. Therefore, the frequency acquisition can be quickly complete. The whole frequency acquisition can be finished within the synchronization patterns to satisfy the requirement of the target applications. The proposed CDR circuit is designed for USB 2.0 high-speed mode, thus, the frequency acquisition should be finished in 31 cycles. Moreover, the proposed dual mode phase and frequency detector (PFD) supports two modes for tracking with the synchronization patterns and the random data patterns. The proposed PFD operates like the common PFD with synchronization pattern. In addition, it can be applied to track the phase of the random data patterns to maintain the frequency and the phase of the recovery clock. When the proposed CDR circuit is compared with the oversampling CDR circuit, the proposed CDR circuit doesn't require a multi-phase clock generator with an external crystal or oscillator as a reference clock. Thus, the design complexity and the cost of the design are greatly reduced.

The rest of the paper is organized as follows: Section II describes the packet format and the overall system operation for the USB 2.0 high-speed mode. The implementation of the proposed CDR circuit is discussed in Section III. Section IV shows the experimental simulation results. Finally, Section V concludes with a summary.

II. OVERALL SYSTEM DESCRIPTION

The packet format for the USB 2.0 high-speed mode is shown in Fig. 2. In the beginning of data transmission, the synchronization pattern is sent first for the CDR circuit to perform frequency and phase acquisition. The synchronization pattern is composed of continuous thirty-one "0" and one "1". In addition, in the USB 2.0

specifications, after six continuous "1", it requires to stuff one "0" to avoid too many continuous "1" that may cause no transition in a long duration. If stuffed bits are not added, it will be very difficult to perform clock synchronization. The packet data with stuffed bits are encoded with non-return to zero, inverted (NRZI) encoder. The NRZI signal has a transition if the bit being transmitted is "0", and does not have a transition if the bit being transmitted is "1". Therefore, the CDR circuit can perform frequency and phase acquisition with the synchronization pattern.
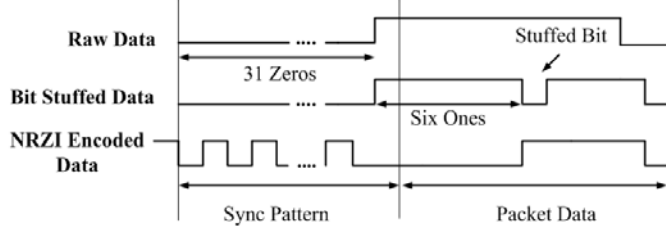

FIGURE 2. Packet format of the USB 2.0 high-speed mode.

The proposed referenceless all-digital fast frequency acquisition full-rate CDR circuit is shown in Fig. 3. It is composed of a dual mode phase and frequency detector (PFD), a TDC-embedded digitally controlled oscillator (DCO), a lock-in procedure control state machine (State Machine) and a CDR controller (Controller). The data transition (Data_T) is extracted by delayed the input data and exclusive-OR with the original input data. The lock-in procedures of the proposed CDR circuit has three phases, TDC-locking phase, coarse-tuning phase and fine-tuning phase. The lock-in procedure control state machine which triggers by the Data_T controls the lock-in procedures. In TDC-locking phase, the symbol rate of the input data, which is the time difference between two positive edge transitions of the Data_T in the synchronization pattern, is measured by the TDC-embedded DCO. Then, the quantized period information (TDC_code) is sent to the CDR controller to set up the initial value of the DCO control code (DCO_Code). After the TDC-locking phase, the DCO's output frequency is close to the target frequency, and the lock-in procedure control state machine turns off the TDC function of the TDC-embedded DCO by setting TDC_lock signal to "1".
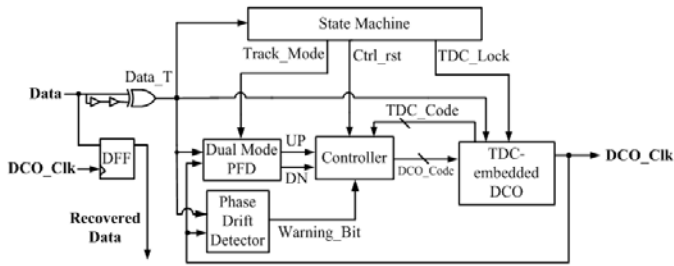

FIGURE 3. The proposed referenceless all-digital CDR circuit.

After the TDC-locking phase, the lock-in procedure control state machine enters coarse-tuning phase. In this phase, the lock-in procedure control state machine sets the operation mode of the dual mode PFD to the PFD mode by setting Track_Mode signal to "1". The dual mode PFD operates like a common PFD in this mode. When the output clock (DCO_Clk) leads the data transition edge (Data_T), the DN signal is generated which means that the DCO should be slowed down. Otherwise, the UP signal is generated, and the DCO should be speeded up. The CDR controller adjusts the DCO control code (DCO_code) according to UP/DN signals to coarse-tune the DCO's output frequency. After two phase polarity changes, the

frequency error is reduced and the state machine enters the fine-tuning phase.

In the fine-tuning phase, the lock-in procedure control state machine sets the operation mode of the dual mode PFD to PD mode by setting Track_Mode signal to "0". In PD mode, the dual mode PFD can track the random data bits which may have no transition for maximum 6 bit time. In the fine-tuning phase, the proposed CDR circuit keeps tracking the phase error between the data transition (Data_T) and the output clock (DCO_Clk) to fine-tune the DCO's output frequency. To avoid phase drifting while no data transition in a long duration, when phase error between the data transition edge (Data_T) and the output clock (DCO_Clk) is larger than half period of the bit time, the phase drift detector sets the Warning_Bit signal to "1" to increase the fine-tuning step until the phase error is smaller than the half period of the bit time. Then, the phase drift detector sets the Warning_Bit signal back to "0", and the fine-tuning step is reduced.
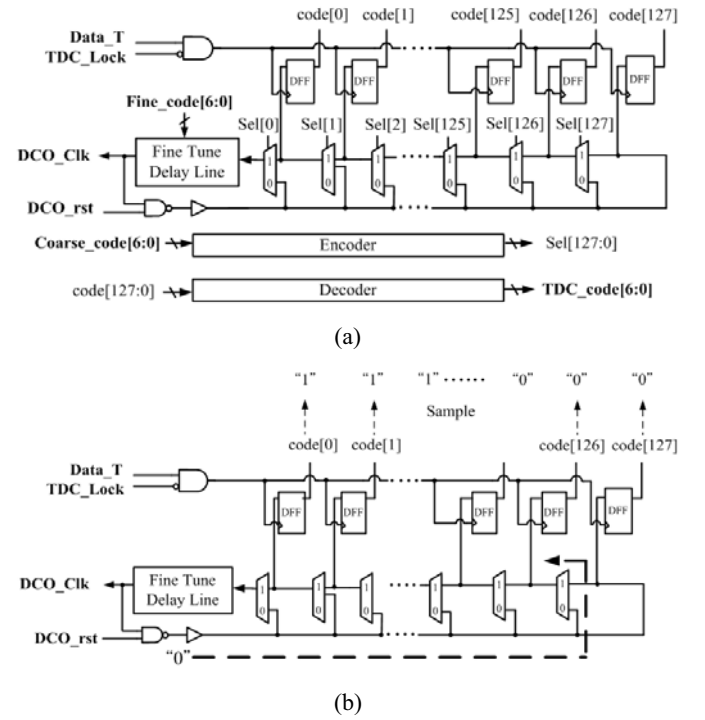
III. CIRCUIT IMPLEMENTATION

(a)

(b)

FIGURE 4. (a) The proposed TDC-embedded DCO. (b) The operation of the TDC.

The detail of the proposed TDC-embedded DCO circuit is shown in Fig. 4 (a). The DCO is composed of a coarse-tuning delay line and a fine-tuning delay line. The coarse-tuning delay line is constructed by a chain of 2-to-1 multiplexers (MUXs). The fine-tuning delay line is composed of the digital-controlled varactors (DCVs) [6] to achieve finer resolution. The DCO control code (DCO_Code[13:0]) is composed of a coarse-tuning control code (Coarse_code[6:0]) and a fine-tuning control code (Fine_code[6:0]). The encoder of the DCO encodes the binary control codes (Coarse_code[6:0]) into the thermometer codes (Sel[127:0]) for coarse-tuning selection. The resolution of the coarse delay line is 34ps. In addition, the resolution of the fine-tuning delay line is 8ps and the total delay controllable range of the fine-tuning delay line is 56ps that can cover the delay step of the coarse-tuning delay line.

To achieve fast frequency acquisition, the TDC is applied to quantize the timing difference between two consecutive positive edge transitions of the Data_T. Fig. 4 (b) shows the operation of the proposed TDC. When the CDR circuit is reset, the input signal of DCO_rst is "0". In addition, the coarse-tuning control signal (Sel[127:0]) is set to its maximum value to choose the longest path of the coarse-tuning delay line. Then, when the first positive edge transition of the Data_T comes, the DCO_rst is set to "1". The output of the bottom NAND gate produces a "0" which begins to pass through the coarse-tuning delay line. When the second positive edge transition of the Data_T comes, the D-Flip/Flops (DFFs) sample the node values in the coarse-tuning delay line and save as code[127:0]. Then, the decoder of the DCO decodes the thermometer codes (code [127:0]) into binary TDC codes (TDC_code[6:0]). The TDC quantizes the period information of the Data_T into times of the MUXs delay time. As a result, the CDR controller can use the TDC code to set the initial value of the DCO control code, and the frequency acquisition of the proposed CDR can be achieved in a very short time.

After the measurement of the symbol rate is complete by the TDC, the lock-in procedure control state machine sets the TDC_Lock signal to "1" to turn off the TDC operation to save the power consumption. The frequency error after TDC-locking phase can be further compensated by the coarse-tuning delay line and the fine-tuning delay line in the following coarse-tuning phase and fine-tuning phase of the lock-in procedures.
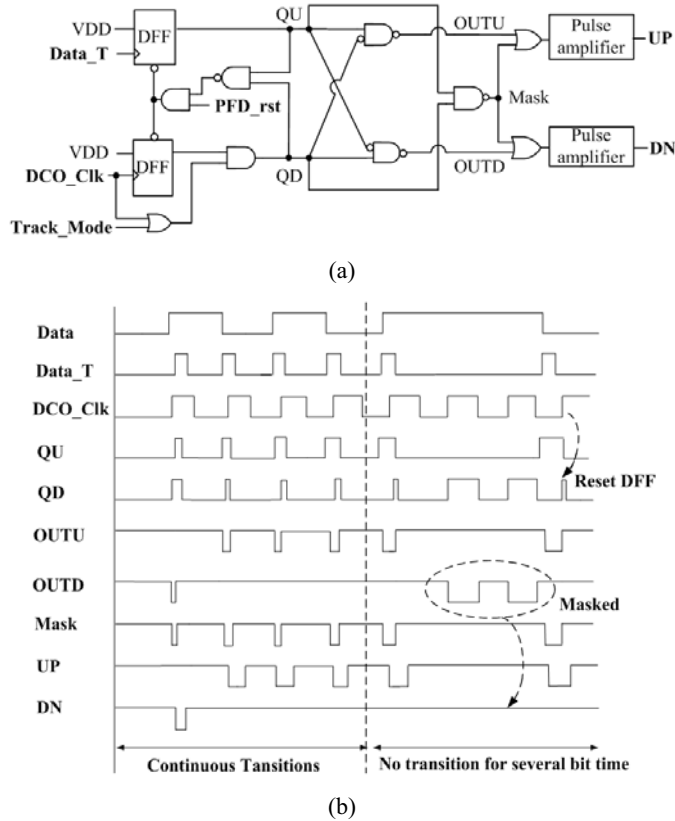

(a)

(b)

FIGURE 5. (a) The proposed dual mode PFD. (b) Timing diagram of the dual mode PFD.

The detail circuit of the proposed dual mode PFD is shown in Fig. 5 (a). After the synchronization patterns, in the random data patterns, there may have no transition for several bit time as shown in Fig. 5 (b). Therefore, the conventional PFD can not be applied in this condition, and it will produce wrong UP/DN signals. Therefore, the

dual mode PFD is proposed in this paper to overcome this problem. The operation mode of the proposed PFD is controlled by the signal Track_Mode. When the Track_Mode is set to "1", the dual mode PFD operates just like the conventional three-state PFD. This mode is used with the synchronization patterns to track the phase and the frequency of the Data_T with continuous transitions.

When Track_Mode is set to "0", the dual mode PFD is switched to the PD mode. This mode is used with the random data patterns to maintain the frequency and the phase between the Data_T and the DCO_Clk. When there is no data transition for several bit time, the upper DFF triggered by Data_T is not triggered, thus, the QU signal is "0" in this region. Therefore, the Mask signal is "1", and the PFD's output signals UP/DN are disabled by the Mask signal. Thus, the PFD is disabled in this region to avoid producing wrong UP/DN signals to the CDR controller. In no data transition region, the output QD signal of the bottom DFF triggered by DCO_Clk is set to "0" to ignore the edge transition of the DCO_Clk. Subsequently, when the positive edge transition of the Data_T comes, two DFFs are reset, and the PFD is enabled again. Finally the pulses are passing through the pulse amplifier that the pulse width of the UP/DN signals would be wide enough that the controller can recognize them.
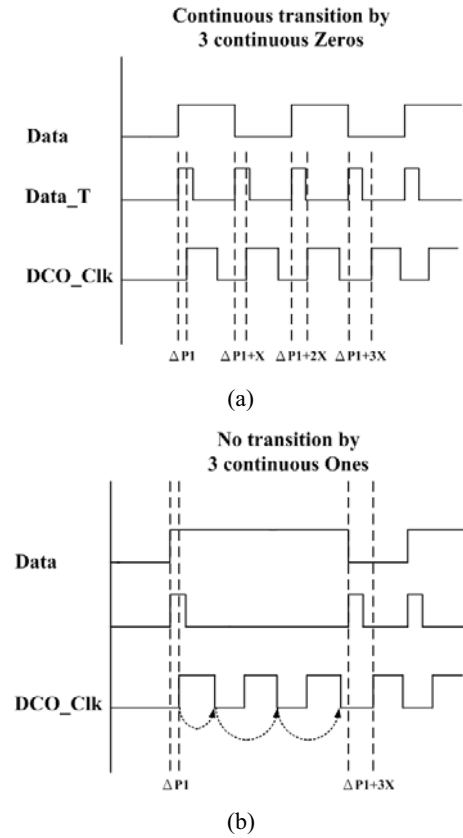

(a)

(b)

FIGURE 6. Phase acquisition with (a) continuous data transition (b) no data transition.

In no data transition region, the PFD is disabled. Thus, the frequency error between the Data_T and the DCO_Clk will accumulate the phase error. To achieve robust phase tracking in no data transition region, the adaptive DCO_code control scheme is presented. Fig. 6 (a) shows the NRZI data stream encoded by 3 continuous "0", the CDR circuit receives continuous data transitions in this case. If the period difference between Data_T and DCO_Clk is X, and the initial phase error is $\Delta P1$. After 3 cycles, the phase error is increased to $\Delta P1+3X$ because of the frequency error. Fig. 6 (b) shows the NRZI data stream encoded by 3 continuous "1", there

is no data transition during three DCO_Clk cycles. Thus, after 3 cycles, the phase error is also increased to $\Delta P1+3X$. The proposed dual mode PFD is disabled in this region, and therefore, there is no DCO control code adjustment, too. So that, the edge transitions of DCO_Clk in no data transition region are counted. When the next Data_T edge transition comes, the DCO control code is increased or decreased by the counter value. Thus, the accumulated phase error in no data transition region can be quickly compensated by the adaptive DCO control scheme.

IV. EXPERIMENTAL RESULTS

Fig. 7 shows the simulation results of the proposed CDR circuit. The initial value of the coarse-tuning control code (Coarse_code[6:0]) is set to its maximum value to choose the longest path of the coarse-tuning delay line for TDC operation. After the CDR circuit is reset, the first input pattern is the synchronization pattern with regular data transition. In the first two Data_T cycles, the timing difference between two consecutive positive edge transitions of the Data_T is measured by the TDC, and then the TDC code is used in the CDR controller to set the initial value of the DCO control code.

After the TDC-locking phase, the output frequency is close to the target symbol rate. Then, in the coarse-tuning phase, the CDR controller adjusts the DCO control code to further reduce the frequency error and phase error. Subsequently, in fine-tuning phase, the proposed CDR circuit continues tracking the phase and frequency of the Data_T with the proposed dual mode PFD. To simulate the jitter effects of the data jitter, the input data stream with jitter effects is inputted. The proposed CDR passes the simulation with input peak-to-peak jitter up to 150ps with the bit error rate (BER) less than 10^{-12}.
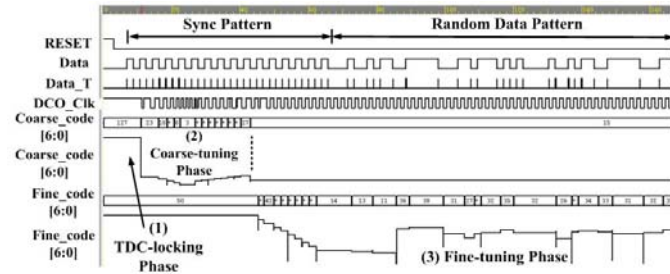

FIGURE 7. Simulation results of the proposed CDR circuit.

The proposed CDR circuit is implemented in a standard performance (SP) 65nm CMOS technology. Fig. 8 shows the layout of the proposed design. The core area is 150μm × 150μm including the on-chip testing components. The operation frequency of the proposed CDR is at 480MHz which satisfies the USB 2.0 high-speed mode specifications. The power consumption with 1.0V supply voltage is 1.75mW. The summary of the test chip is shown in Table I.

V. CONCLUSION

In this paper, an all-digital fast frequency acquisition full-rate CDR circuit without reference clock is implemented in 65nm CMOS technology. The proposed CDR circuit does not need an external reference clock and the oversampling scheme as in prior studies. Thus, the area cost and the power consumption are both reduced. The proposed TDC-embedded DCO is used to achieve fast frequency acquisition in one step. Therefore, the whole frequency acquisition time can be done within 31 cycles, and the proposed CDR circuit satisfies the requirements of the USB 2.0 high-speed mode.
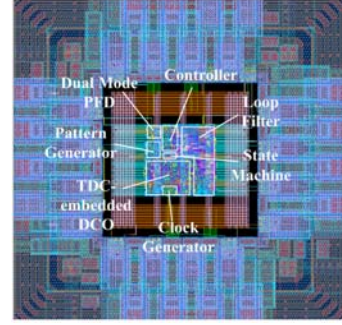

FIGURE 8. Layout of the proposed CDR circuit.

TABLE I. Chip summary.

Process	65nm CMOS
Operating speed	480MHz
Supply Voltage	1.0V
Area	150 μm ×150 μm
Power Consumption	1.75 mW
Locking time	< 31 cycles
Input Jitter Tolerance	150 ps with BER < 10^{-12}
Reference clock	No

ACKNOWLEDGEMENT

REFERENCES

[1] Sang-Hyun Lee, et al., "A 5-Gb/s 0.25-μm CMOS jitter tolerant variable-interval oversampling clock/data recovery circuit," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1822-1830, Dec. 2002.

[2] Rong-Jyi Yang, et al. "A 155.52 Mbps – 3.125 Gbps continuous-rate clock and data recovery circuit," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1380-1390, Jun. 2006.

[3] Shao-Hung Lin and Shen-Iuan Liu, "Full-rate bang-bang phase/frequency detectors for unilateral continuous-rate CDRs, " *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 55, no.12, pp. 1214-1218, Dec. 2008.

[4] Sitt Tontisirin and Reinhard Tielert, "A Gb/s one-forth-rate CMOS CDR circuit without external reference clock, " in *Proc. IEEE International Symposium on Circuit and Systems (ISCAS)*, May. 2006, pp. 3265-3268.

[5] Seon-Kyoo Lee, et al., "A 650Mb/s-to-8Gb/s referenceless CDR circuit with automatic acquisition of data rate, " in *ISSCC Dig. Tech. Papers*, Feb. 2009, pp. 184-185.

[6] Pao-Lung Chen, Ching-Che Chung and Chen-Yi Lee, "A portable digitally controlled oscillator using novel varactors," *IEEE Transaction on Circuits and System II: Analog and Digital Signal Processing*, vol. 52, no. 5, pp. 233-237, May 2005.