

# A Power and Area Efficient CMOS Clock/Data Recovery Circuit for High-Speed Serial Interfaces

Dao-Long Chen

**Abstract**—A power and area efficient CMOS clock/data recovery circuit designed for a wide range of applications in high-speed serial data communications is described. It uses an analog phase-locked loop (PLL) to generate the high-speed clocks with an absolute rms jitter of less than 60 ps and a digital PLL<sup>1</sup> which is designed to minimize chip area and power consumption to recover the clock and data signals from the incoming data stream. Fabricated in a 0.8  $\mu\text{m}$  single-polysilicon, double-metal CMOS process, the digital PLL only consumes 45 mW at 125 Mb/s from a single 5 V supply, while the analog PLL consumes 92 mW. The chip area is 1.7 mm<sup>2</sup> for the digital PLL and 0.44 mm<sup>2</sup> for the analog PLL. It can handle an input data rate up to 280 Mb/s.

## I. INTRODUCTION

IN recent years, many serial interface standards have been proposed to improve the I/O performance of a computer system or network [1]–[5]. By embedding the clock signal into the transmitted data stream, a serial interface can operate at very high data rates without the timing skew problem between the clock and data signals. However, at the receiving end, a clock/data recovery circuit is needed to recover the embedded clock signal from the incoming data stream and to retime the data.

Traditionally, analog phase-locked loops (PLL's) have been used to implement the clock/data recovery circuit for high-speed operations [6], [7]. Although, in general, the analog PLL's can operate at higher frequencies, they tend to be more difficult to design than their digital counterparts. For example, the designers have to consider the frequency drift during long strings of ones and zeros in the incoming data stream and the phase/frequency reacquisition process after a possible loss of bit synchronization due to a short interruption in the incoming data stream. Furthermore, as pointed out in [8], [9], and [10], analog PLL's are also more sensitive to supply noise and to variations in processing and operating conditions.

As a result, whenever the available process technology permits, designers have tried to replace analog PLL's with digital PLL's [9]–[12]. However, there are also a few drawbacks associated with the digital PLL's besides the limitation on the operation speed. They are generally worse than the analog PLL's in terms of chip area and power consumption. As a stand-alone chip, these are usually not serious problems. But, as the PLL is integrated with other serial interface functions, it becomes desirable to minimize both the chip area and power consumption of the clock/data recovery circuit. It is particularly true when we need to integrate more than one

clock/data recovery circuit on a single chip, as in many circuit- or packet-switching oriented applications.

This paper describes a high-speed clock/data recovery circuit which is specifically designed to optimize chip area and power consumption. Similar to [10] and [11], a hybrid analog/digital PLL architecture is used in this design to take advantage of the best of what an analog PLL and a digital PLL can do. However, numerous improvements have been made to the architecture so that it can operate at higher data rates while consuming less power and occupying less area. Special attention has also been paid to the design of the clock/data recovery circuit so that it can be used as a macrocell in different applications.

In Section II, the architecture, as well as the special features of this optimized clock/data recovery circuit are presented. We will compare our approach with the architectures previously published and show how power and area can be saved. The design issues which are critical to the overall circuit performance are also discussed. In Section III, the experimental results are presented and discussed. The performance of this design is compared with what has previously been published. Finally, Section IV gives the conclusion.

## II. PLL ARCHITECTURE AND CIRCUIT DESIGN

Fig. 1 shows the block diagram of the clock/data recovery circuit implemented in this design. The initial phase acquisition is accomplished by oversampling the incoming data with ten clock phases generated by the analog PLL as shown in Fig. 2. Based on the sampling results, the data sampler can determine where the first data transition occurred and select the clock phase which is closest to the data transition as the recovered clock. Once the initial phase acquisition is done, the data sampler is disabled to reduce power consumption. In the mean time, the phase detector and the loop filter are enabled to keep the recovered clock phase-locked to the incoming data. Since the initial phase acquisition is based on the location of the first transition on the data signal and it is possible that the first few transitions could be corrupted, a pulse swallower circuit is provided which, when enabled, can remove the first two pulses from the incoming data signal. A more detailed description of each functional block and their design considerations are as follows.

The analog PLL is simply a clock synthesizer which generates multiple high-speed clocks from an off-chip reference clock [13]. Although a delay-locked loop could have smaller phase jitter [11], it is not a cost-effective solution at higher speeds since a more expensive crystal oscillator is required.

Some of the earlier clock/data recovery circuits have implemented digital clock synthesizers to generate the high-speed

Manuscript received July 31, 1995; revised December 15, 1995.

The author is with Symbios Logic Inc., Fort Collins, CO 80525 USA.

Publisher Item Identifier S 0018-9200(96)05697-1.

<sup>1</sup>A digital PLL is defined in this paper as a hardware-based PLL consisted of digital functional blocks.

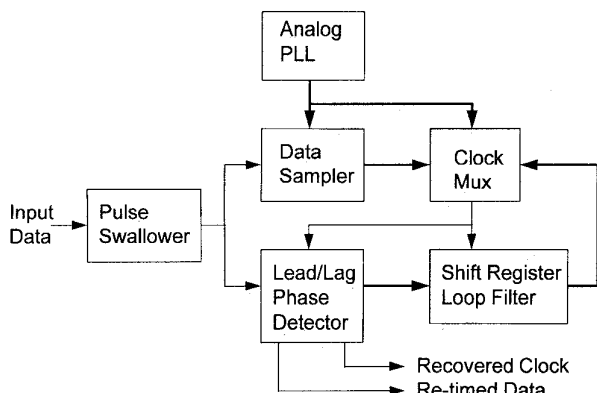


Fig. 1. Functional block diagram of the clock/data recovery circuit.

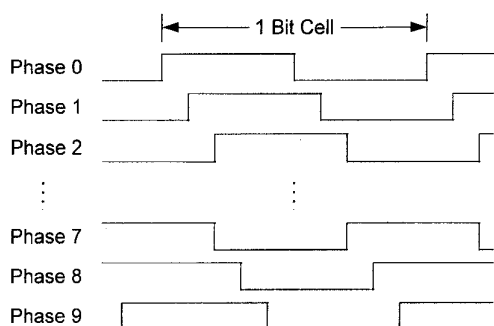


Fig. 2. The analog PLL generates ten clock phases which are equally spaced in time.

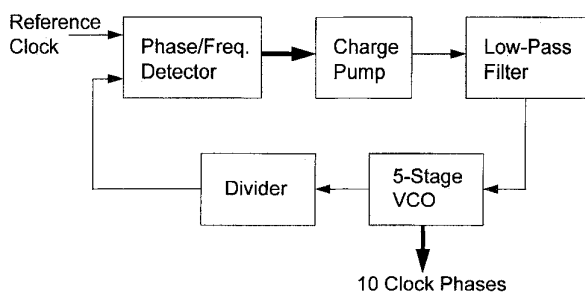


Fig. 3. Functional block diagram of the analog PLL clock synthesizer.

clocks [9], [12]. However, they are generally not very power and area efficient. The typical reason for using a digital clock synthesizer is to make the design more robust against supply noise and variations in processing and operating conditions. Although it may have been true in the past, the advances in analog PLL clock synthesizer design in recent years have made it possible to integrate the analog clock synthesizer with millions of other transistors on the same chip in high-volume applications [14]. Many semiconductor companies now offer analog PLL clock synthesizers as a macrocell in their cell libraries. Therefore, in this design we chose to use an analog PLL to save power and area.

Besides generating the high-speed clocks for the transmit path, the analog PLL also generates multiple clock phases, which are equally spaced in time, as input to the digital PLL. In [9], [10], and [11], 32 clock phases are generated to

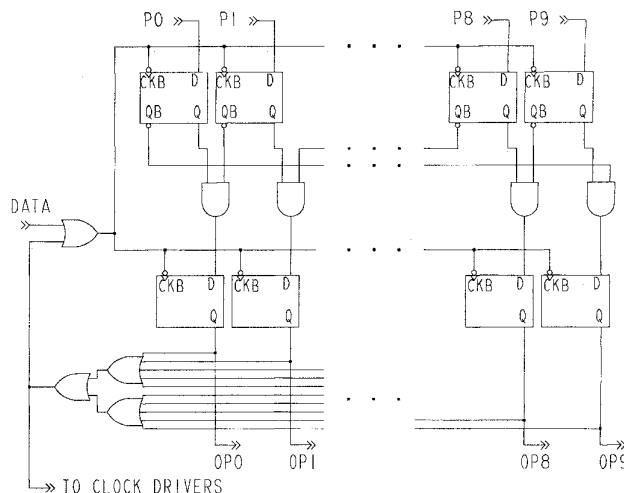


Fig. 4. Simplified circuit schematic of the data sampler.

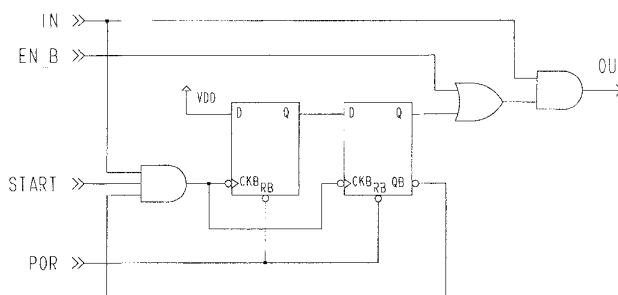


Fig. 5. Circuit schematic of the pulse swallower circuit.

oversample the incoming data.<sup>2</sup> The amount of oversampling used will directly affect the tolerance of phase jitter on the incoming data. Although oversampling as low as 3× has been used [15], the input jitter tolerance will be relatively poor. On the other hand, if too many clock phases are used, power consumption and chip area will increase.

To compromise, a voltage-controlled oscillator (VCO) with five differential oscillator stages has been chosen. Fig. 3 shows the block diagram of the analog PLL. With the five-stage differential VCO, ten clock phases are available to sample the data. For all the applications mentioned earlier [1]–[5], the input jitter tolerance obtained from a 10× oversampling rate is adequate, provided that other sources of error, such as the random jitter and duty cycle distortion on the VCO clocks, are properly controlled. Since good results in terms of random jitter and duty cycle distortion have previously been demonstrated [16], an input jitter tolerance of better than 60% of the bit time should be easily achievable with a 10× oversampling rate. The same analog PLL, with the same VCO circuit reported in [7] and [16], is also implemented in this design. The only differences are that a five-stage VCO and a divide-by-five divider were used in this design.

In [9], [10], and [12], the multiple clock phases from the clock synthesizer are used by the digital PLL to sample the incoming data and determine where the transitions occurred.

<sup>2</sup>In [11], actually, the data signal is used to strobe the clock signals, as will be explained later in this section.

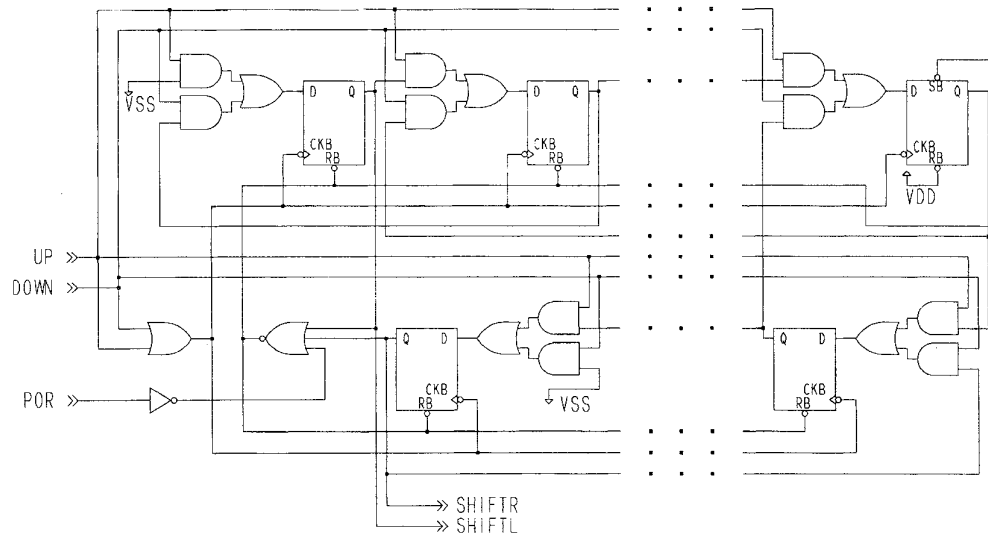


Fig. 6. Circuit schematic of the bidirectional shift register.

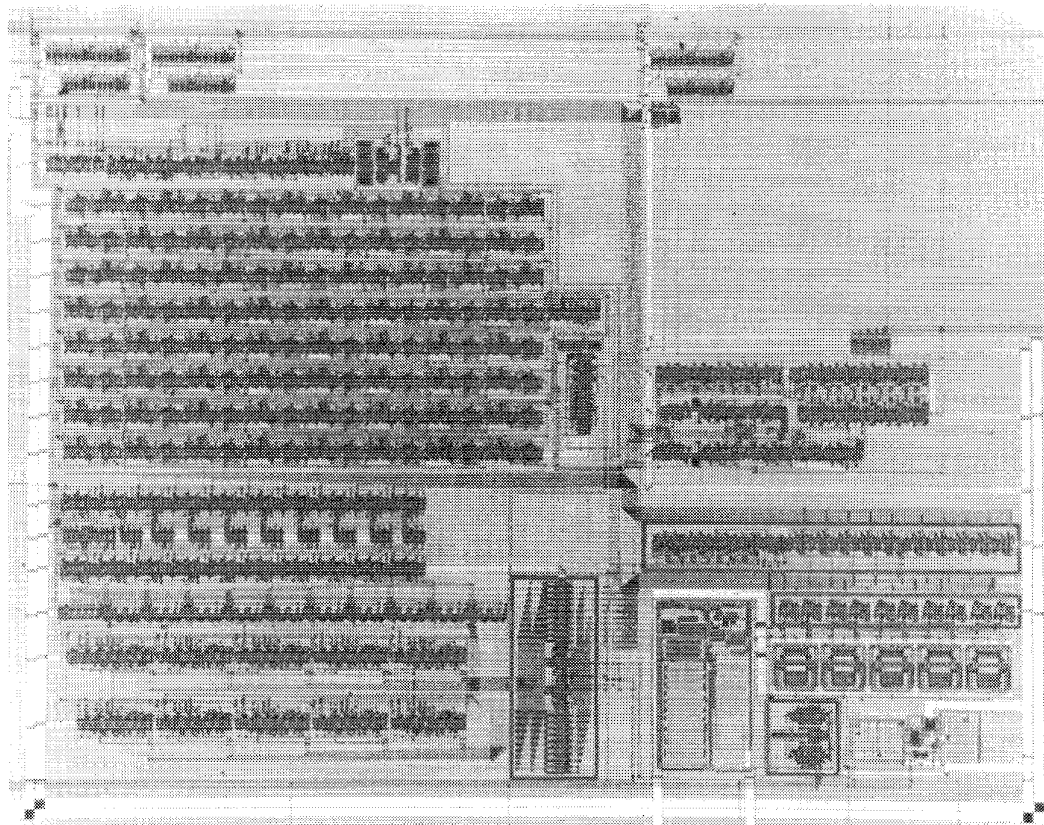


Fig. 7. Photomicrograph of the circuit.

However, due to the phase jitter on the input data signal, it is possible that there will be more than one transition detected within a single sampling period. Extra logic is needed to handle situations like that [10]. Since it is important for us to save power and area, instead of using the clocks to sample the data, this implementation uses the data to sample the clocks, as proposed in [11]. However, a major difference in our approach is that the data sampler is only used during the initial phase

acquisition period. Once the phase acquisition is accomplished, the data sampler and the associated clock drivers are disabled.

Fig. 4 shows the circuit schematic of the data sampler. The DATA signal is used by the DFF's to strobe the ten clock phases. The first high-to-low transition on the DATA signal will cause one of the AND gates to switch from logic 0 to logic 1. Subsequently, one of the output signals, OP0 to OP9, will switch from logic 0 to logic 1 after the second high-to-low

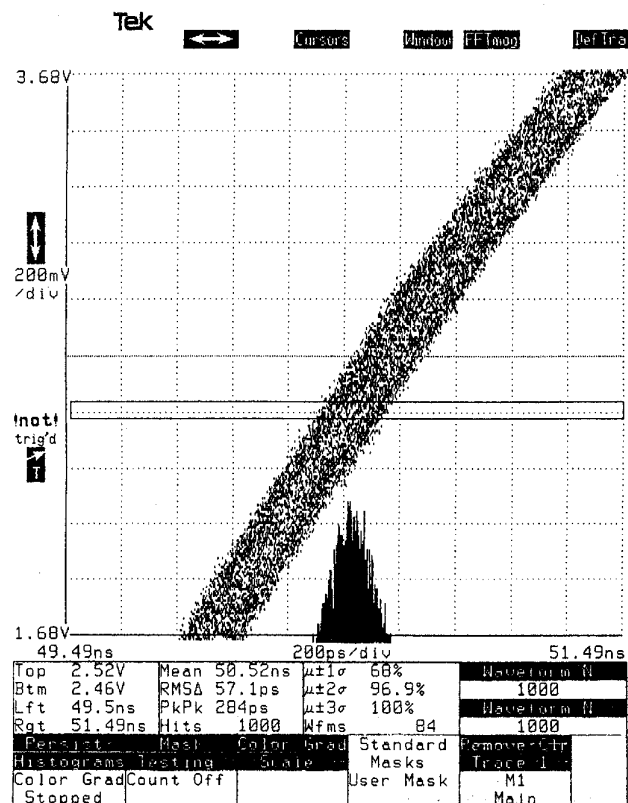


Fig. 8. Phase jitter histogram of the analog PLL.

transition on the DATA signal. This instructs the digital PLL which one of the ten clock phases should be selected initially as the recovered clock.

After the initial phase acquisition is done, the data sampler is disabled by itself through the OR gate feedback path. As a result, power consumption is greatly reduced. Furthermore, since the data sampler is not used during normal data tracking, a transition pulse generator, which is part of the midbit transition detector described in [11], is no longer needed. This further reduces power consumption and chip area. Note that two rows of DFF's are used in the data sampler. Besides allowing the sampler to operate at higher speeds through pipelining, they also reduce the probability of any potential metastability problems due to asynchronous sampling.

With the help of the data sampler, the clock/data recovery circuit can lock onto the incoming data in as low as three bit times after the first low-to-high transition. The actual phase acquisition time depends on the data encoding scheme used. If it is desirable to avoid locking onto the first few transitions, the pulse swallower circuit, as shown in Fig. 5, can be enabled to take out the first two pulses on the input data signal.

When the initial phase acquisition is done, one of the ten clock phases will be selected as the recovered clock and be used to retim the incoming data. In the mean time, the digital PLL is switched to the normal data tracking mode. Contrary to what had previously been implemented in [9]–[12], a simple lead/lag phase detector, comprised of just a few flip-flops, is used in the normal data tracking mode [8]. Instead of a binary

number, the output of the phase detector is now an UP or DOWN pulse. Consequently, the binary encoder, which had been used in [9]–[11], is no longer needed. This further reduces the power consumption and chip area.

Another major advantage of using a lead/lag phase detector is that a much simpler digital loop filter can be used. In [9]–[11], the digital loop filters consist of several arithmetic units. As a result, they are relatively complicated and consume much power. Besides, they tend to limit the maximum speed of the entire clock/data recovery circuit at higher data rates. By using the lead/lag phase detector, either a simple up/down counter or a bidirectional shift register can be used as the loop filter. Although the up/down counter is smaller, we decided to choose the shift register for its speed. A programmable shift register has been implemented in this design. The user can program it as a 8-b, 16-b, or 32-b shift register to adjust the loop bandwidth.

Fig. 6 shows the simplified circuit schematic of the bidirectional shift register. After power-up, all the outputs of the flip-flops in the shift register are reset to zero except the output of the center flip-flop. By shifting the output of the center flip-flop, which is a logic 1, to the left or to the right based on the UP and DOWN signals, the random phase noise on the data signal can be filtered out. Since it is unlikely that the phase jitter will be evenly distributed around the selected clock phase, after a while the logic 1 output will be shifted to the left-most or right-most flip-flop. When that happens, the clock phase selected as the recovered clock will also be shifted accordingly. As a result, most of the time the recovered clock will be jumping back and forth between two adjacent clock phases. After the selected clock phase is switched, the shift register will reset itself and return to the same initial condition after power-up. It should be noted that, due to the finite frequency difference between the transmitter and receiver, the recovered clock will gradually shift from one clock phase to another over time even if there is no phase jitter. Nonetheless, since in most of the applications the maximum frequency difference is specified at  $\pm 100$  ppm or less [1]–[5], this happens very slowly in comparison to the bit time.

### III. EXPERIMENTAL RESULTS

The clock/data recovery circuit has been fabricated in a  $0.8 \mu\text{m}$  single-polysilicon, double-metal digital CMOS process. The first pass silicon was fully functional. The chip photomicrograph is shown in Fig. 7. The analog PLL has a chip area of  $0.44 \text{ mm}^2$ , while the digital PLL is  $1.7 \text{ mm}^2$ . The entire circuit can operate up to 280 Mb/s with a  $2^{23} - 1$  pseudorandom bit pattern input. Fig. 8 shows the jitter histogram of the symbol clock, which is the bit clock divided by five, from the analog PLL. The absolute rms jitter is typically smaller than 60 ps. In order to save power, the VCO inside the analog PLL operates at the same frequency as the input data rate. Consequently, special attention was given to the circuit and layout design of the oscillator cells and the clock buffers to ensure a 50% duty cycle on the clock outputs. Otherwise, the input jitter tolerance will suffer due to the static phase alignment error. By probing of the VCO

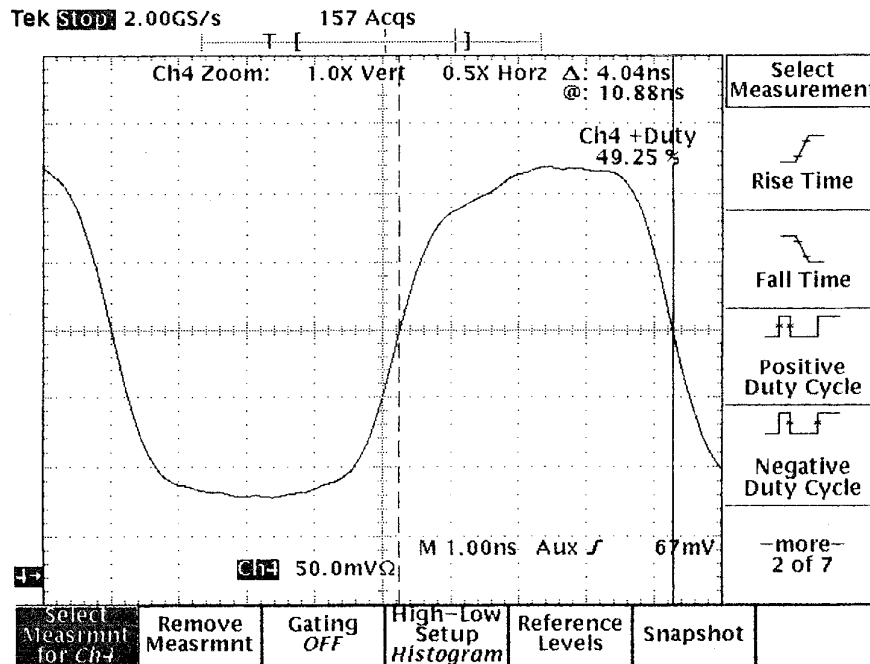


Fig. 9. Typical bit clock waveform at the output of the VCO.

clock outputs, the duty-cycle distortion on the VCO clocks was measured to be less than 2%. Fig. 9 shows the waveform of one of the VCO clock outputs at 125 MHz.

The input jitter tolerance of the digital PLL has been determined by varying the amount of jitter intentionally injected. Fig. 10 shows the heavily jittered input data signal, the recovered symbol clock (recovered bit clock divided by five), and the re-timed data at 125 Mb/s. In all cases, the measured jitter tolerance is larger than 60% of the bit time. Furthermore, since the recovered clock is frequency-locked to a reference clock, the digital PLL can tolerate a very long string of ones or zeros. Even with a worst-case frequency difference of 200 ppm, it takes 500 bits of consecutive ones or zeros to generate a phase shift of one tenth of a bit time. No difference in terms of the input jitter tolerance or the bit error rate has been detected with either the  $2^7 - 1$  or  $2^{23} - 1$  pseudorandom bit pattern.

Fig. 11 shows typical waveforms of the initial phase acquisition process. Trace 1 is the  $2^7 - 1$  pseudorandom input data sequence at 125 Mb/s. Trace 2 is the 25 MHz recovered symbol clock. Trace 3 is the retimed output data sequence and, finally, trace 4 is the START signal as shown in Fig. 5. When the START signal is activated, the first two data pulses are swallowed by the pulse swallower circuit (Fig. 5) and the next two pulses are used by the data sampler (Fig. 4) to select one of the ten clock phases as the recovered clock. After that, the digital PLL is phase-locked to the incoming data stream and begins to decode the input data. The total phase acquisition time is only about 104 ns in this case. Since the analog PLL is always frequency-locked to the reference clock, no time is spent on frequency acquisition.

The power consumption of the analog PLL is 92 mW at 125 MHz with a 5 V supply. More than 50 mW of the power

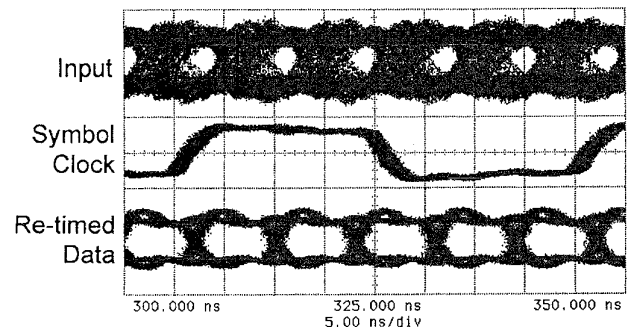


Fig. 10. Eye diagrams of the input and output data signals. The center trace is the recovered symbol clock.

is dissipated in the differential-to-single-ended converters in the VCO, which can be easily reduced by half through a straightforward design modification. The digital PLL consumes 45 mW at 125 Mb/s which is substantially lower than what had previously been reported. Table I shows the data rate, power consumption, and chip area for this design and other previously published designs. If we can define design efficiency as the product of chip area and power consumption divided by the data rate, clearly this design has achieved an efficiency rating that has never been attained before by a digital PLL-based clock/data recovery circuit. It should be noted that, due to fixed bias currents used in the analog PLL, the efficiency rating of this design will actually improve with increased data rates (i.e., the power consumption will not double when the data rate is doubled). Therefore, the numbers listed in Table I can be considered as a conservative comparison.

Since different CMOS processes were used in [9], [10], and [11], Table I also includes design efficiency numbers that

TABLE I  
PERFORMANCE COMPARISON

	This Work	Results in [9]	Results in [10]	Results in [11]	Results in [12]
CMOS Process ( $\mu\text{m}$ )	0.8	1.0	2.0	1.75	0.8
Data Rate (Mb/s)	125	10	30	10	125
Power Consumption (mW)	137	675*	600	125	175
Chip Area ( $\text{mm}^2$ )	2.14	5.96	19.7	4	6.38
Design Efficiency Before Adjustment ( $\text{mWmm}^2/\text{Mb/s}$ )	2.35	402	394	50	8.93
Design Efficiency After Adjustment ( $\text{mWmm}^2/\text{Mb/s}$ )	2.35	206	25.2	4.78	8.93

\* Calculated from the Intel 82C501AD data sheet

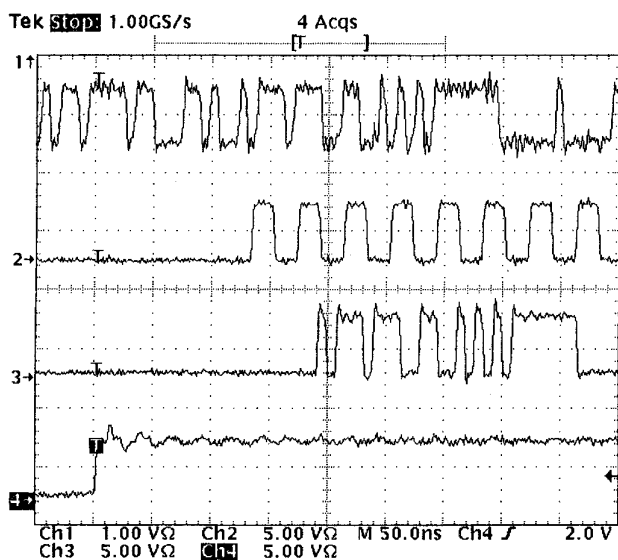


Fig. 11. Initial phase acquisition process.

have been recalculated by normalizing the chip area and power consumption with a  $0.8 \mu\text{m}$  process. For example, it is assumed that a design which was implemented in a  $1 \mu\text{m}$  process will occupy 0.64 of the original chip area if implemented in a  $0.8 \mu\text{m}$  process. Furthermore, since the gate capacitance is likely to decrease by a factor of 0.8, it is also assumed that the power consumption will be reduced by 20%. Consequently, the design efficiency will be adjusted by a factor of 0.512. As shown in Table I, even after the adjustments, this design still shows significant efficiency improvement over previous implementations. It should also be noted that this circuit was designed to cover a wide range of applications. If the maximum speed is not a concern and the shift register is replaced with an up/down counter, both the chip area and power consumption can be substantially reduced.

The results recently published in [15] are not included in Table I due to the lack of information on the analog PLL clock

synthesizer used in the design. However, if a comparison is made solely based on the digital PLL's, this implementation will have a design efficiency of  $0.612 \text{ mW} \cdot \text{mm}^2/\text{Mb/s}$ . In contrast, the digital PLL in [15] has a design efficiency of  $2.17 \text{ mW} \cdot \text{mm}^2/\text{Mb/s}$  before adjusting the chip area and power consumption for different CMOS processes used. It will increase to  $5.14 \text{ mW} \cdot \text{mm}^2/\text{Mb/s}$  after the adjustment since a  $0.6 \mu\text{m}$  process is used in [15].

#### IV. CONCLUSION

A power and area efficient clock/data recovery circuit has been designed for high-speed operations. By combining an analog PLL clock synthesizer with a digital PLL clock recovery circuit, it can tolerate long strings of ones and zeros in the incoming data stream since its clocks are generated by the analog PLL which is frequency-locked to a reference clock. The initial phase acquisition can be achieved in as low as three bit times by sampling the incoming data with multiple clock phases generated by the analog PLL. To save power, the data sampler will disable itself after the initial phase acquisition is accomplished. A single-phase lead/lag detector is used in the digital PLL to maintain bit synchronization during the normal data tracking mode. If, for any reason, bit synchronization is lost during data transmission, the digital PLL will automatically resynchronize itself with the incoming data without external intervention since the frequency of the recovered clock will never drift away. To further reduce power consumption, instead of complicated arithmetic units, a simple shift register is used in the digital PLL as the loop filter. Implemented in a  $0.8\text{-}\mu\text{m}$  single-polysilicon, double-metal standard CMOS process, the digital PLL only consumes  $45 \text{ mW}$  at  $125 \text{ Mb/s}$  from a single  $5 \text{ V}$  supply, while the analog PLL consumes  $92 \text{ mW}$ . The chip area is  $1.7 \text{ mm}^2$  for the digital PLL and  $0.44 \text{ mm}^2$  for the analog PLL. If it is desirable to integrate more than one serial channel on a single chip, several digital PLL's can share one analog PLL clock synthesizer.

## ACKNOWLEDGMENT

The author would like to thank R. Arakawa, J. Kenney, and W. Koldeway for their help on circuit layout, A. Padilla and C. Schwab for their assistance on laboratory setup, R. Bitting, Dr. C. Kurker, and D. Rehm for their valuable suggestions, and E. Marchand for his support throughout this project.

## REFERENCES

- [1] *Fiber Channel—Physical and Signaling Interface (FC-PH)*, American National Standards Institute, Inc., 1994.
- [2] *Serial Storage Architecture—SSA-PH (Transport Layer)*, American National Standards Institute, Inc., 1994.
- [3] *ATM User-Network Interface Specification (ATM-UNI)*, The ATM Forum, Inc., 1994.
- [4] *FDDI Twisted Pair Physical Layer Medium Dependent (TP-PMD)*, American National Standards Institute, Inc., 1994.
- [5] *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications: MAC Parameters, Physical Layer, Medium Attachment Units and Repeater for 100 Mb/s Operation*, ANSI/IEEE Std. 802.3, 1994.
- [6] L. DeVito *et al.*, "A 52 MHz and 155 MHz clock-recovery phase-locked loop," in *ISSCC Dig. Tech. Papers*, 1991, pp. 142–143.
- [7] D.-L. Chen and R. Waldron, "A single-chip 266 Mb/s CMOS transmitter/receiver for serial data communications," in *ISSCC Dig. Tech. Papers*, 1993, pp. 100–101.
- [8] W. C. Lindsey and C. M. Chie, "A survey of digital phase-locked loops," *Proc. IEEE*, vol. 69, pp. 410–431, Apr. 1981.
- [9] M. Bazes and R. Ashuri, "A novel CMOS digital clock and data decoder," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1934–1940, Dec. 1992.
- [10] B. Kim *et al.*, "A 30-MHz hybrid analog/digital clock recovery circuit in 2- $\mu$ m CMOS," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1385–1394, Dec. 1990.
- [11] J. Sonntag and R. Leonowich, "A monolithic CMOS 10 MHz DPLL for burst-mode data retiming," in *ISSCC Dig. Tech. Papers*, 1990, pp. 194–195.
- [12] B. Guo *et al.*, "A 125 Mbs CMOS all-digital data transceiver using synchronous uniform sampling," in *ISSCC Dig. Tech. Papers*, 1994, pp. 112–113.
- [13] D.-L. Chen, "Designing on-chip clock generators," *IEEE Circuits Dev.*, vol. 8, pp. 32–36, July 1992.
- [14] J. Schutz, "A 3.3 V 0.6  $\mu$ m BiCMOS superscalar microprocessor," in *ISSCC Dig. Tech. Papers*, 1994, pp. 202–203.
- [15] S. Kim *et al.*, "An 800 Mbps multi-channel CMOS serial link with 3 $\times$  oversampling," in *Proc. IEEE CICC*, 1995, pp. 451–454.
- [16] D.-L. Chen *et al.*, "A 500 MHz CMOS phase-locked loop clock generator," in *Proc. IEEE EDMS*, 1992, pp. 81–84.