A CMOS High Speed Data Recovery Circuit Using the Matched Delay Sampling Technique [1]

Jin-Ku Kang, Wentai Liu, Ralph K. Cavin, III

Department of Electrical and Computer Engineering
North Carolina State University

Abstract

This paper presents a scheme and circuitry for demultiplexing and synchronizing high-speed serial data using the matched delay sampling technique. By simultaneously propagating data and clock signals through two different delay taps, the sampler achieves a very fine sampling resolution which is determined by the *difference* between the data and clock delays. Thus, the sampler is capable of oversampling high speed data signals without the need of a high-speed clock and it could be used in a data recovery circuit. A data recovery circuit using the matched delay sampling technique has been designed and fabricated in 1.2μm CMOS technology. The chip has been tested at 417Mb/s (2.4ns NRZ) input data and demultiplexes serial input data into four 104Mb/s output streams with 800mW power consumption at 4V power supply. While recovering data, the sampling clock running at 1/4 of the data frequency is phase-tracking with the input data based on information extracted from a digital phase control circuit.

[1]This research is supported by NSF MIP-9212346

1 Introduction

In high-performance computing and communications systems (*e.g.* SONET or ATM), a high-bandwidth network interface is an important component. Such interfaces require high speed receivers which function at at data rates 500Mbs and above. Previously reported high speed receivers have been implemented in GaAs, bipolar or special nMOS technologies [1] [2] [3] [4]. With advancements in CMOS technology, it is becoming practical to implement most receiver functions in CMOS in order to provide low power, low cost, and high integration.

Data recovery systems using a data-sampling method with phase detection and decision logic to select the most appropriate sample as the recovered data have been developed in CMOS technology [5] [6] [7]. These oversampling techniques, in which only the clock (or data) is propagated through delay taps for generating multi-phase clocks(or data), limit the sampling resolution to the minimal intrinsic gate delay. As a result these techniques were applied to relatively low speed system applications. To accommodate a higher data rate by the single delay line oversampling method, others chose to lower the number of sampling points per one bit input datum [8] [9]. However, as the number of samples per bit is reduced, the data recovery circuit becomes more vulnerable to jitter in the input data or clock. Therefore tighter control of the jitter is required for higher speed data recovery using the single delay line sampling method with a low oversampling ratio.

For high speed data recovery, we adopt a data sampling scheme called *the matched delay sampling technique* [10] in which the data and sampling clock are simultaneously propagated through two different delay lines so that the sampling resolution is the *difference* between the two delay unit times rather than their absolute delays. Thus, sampling resolution can be much finer than with conventional sampling techniques. Finer sampling resolution provides the capability of oversampling even high input data rate. By allowing the data recovery function to operate on a large number of samples per bit, jitter tolerance can be improved. Also, this technique requires a clock at only $1/N$ of the data frequency, since the sampler captures multiple (N) data bits at a time. The circuit effectively functions as a 1:N demultiplexor. Our structure therefore avoids using a high speed clock.

In this recovery technique, each bit of input data is oversampled 16 times with a very fine sampling resolution. A *'correct bit'* in the digitized oversampled points is then selected as recovered data by locating the NRZ input data boundary. For one clock cycle, N data bits are recovered in parallel. By extracting the phase relationship between the input data and clock from the oversampled points, a digital phase control circuit provides dynamic alignment of the clock rising edge to the mid point of the incoming NRZ data. The recovery technique could be used on SONET-based telecommunications, ATM digital communications, transceiver circuits and other high speed data communications.

This paper describes our data recovery technique and its CMOS implementation us-

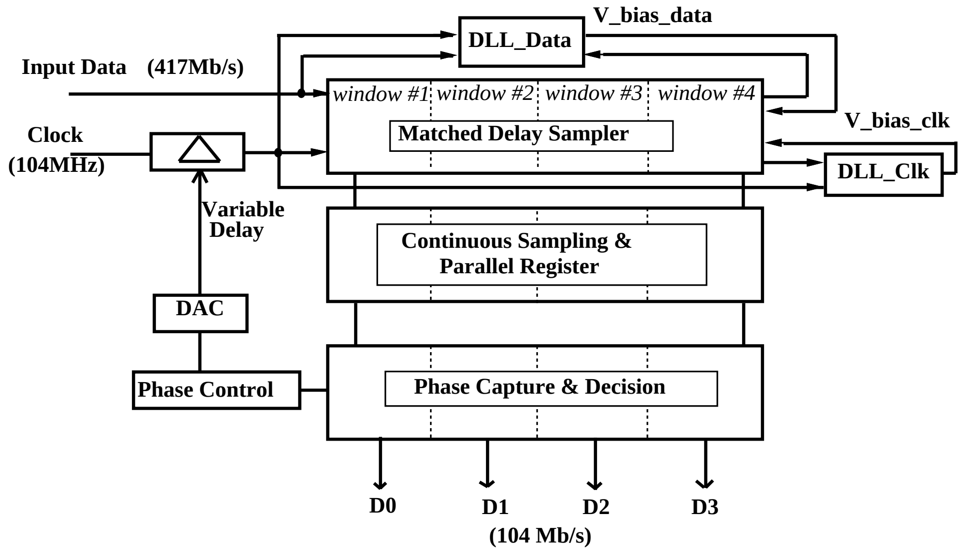


Figure 1: Block diagram for data recovery circuit

ing a high resolution matched delay sampler [11]. Section 2 discusses the basic theory of the data recovery circuit which includes the matched delay sampler, requirements for continuous sampling and decision and phase control blocks. Section 3 then describes a circuit implementation of the data recovery circuit with on-chip test structures. This is followed in Section 4 by a stability analysis of the data recovery circuit. Experimental results on the fabricated chips are described in Section 5, followed by the conclusions.

2 Principle of Operation

A block diagram of the 1:4 demultiplexing data recovery circuit is shown in Fig. 1. The sampler is divided into four windows with each window consisting of 16 sampling stages. The waveforms seen at each processing block during steady-state phase locking are shown in Fig. 2. As shown in Fig. 2, in the steady state the sampling clock is aligned to the middle point of the first bit of the four input data bits to be recovered each clock cycle. After oversampling the input data, the sampled points are digitally processed to recover the data and to lock the phase between the data and clock.

The sampler is designed to oversample input data at some desired resolution. A block diagram of the sampler is shown in Fig. 3. The data and clock are propagated through separate delay chains, consisting of delays D_d and D_c respectively. The two delay locked loops shown in Fig. 1 maintain the nominal delay time on each delay line across varying operating conditions. Each latch samples the data sampling point at its clock's

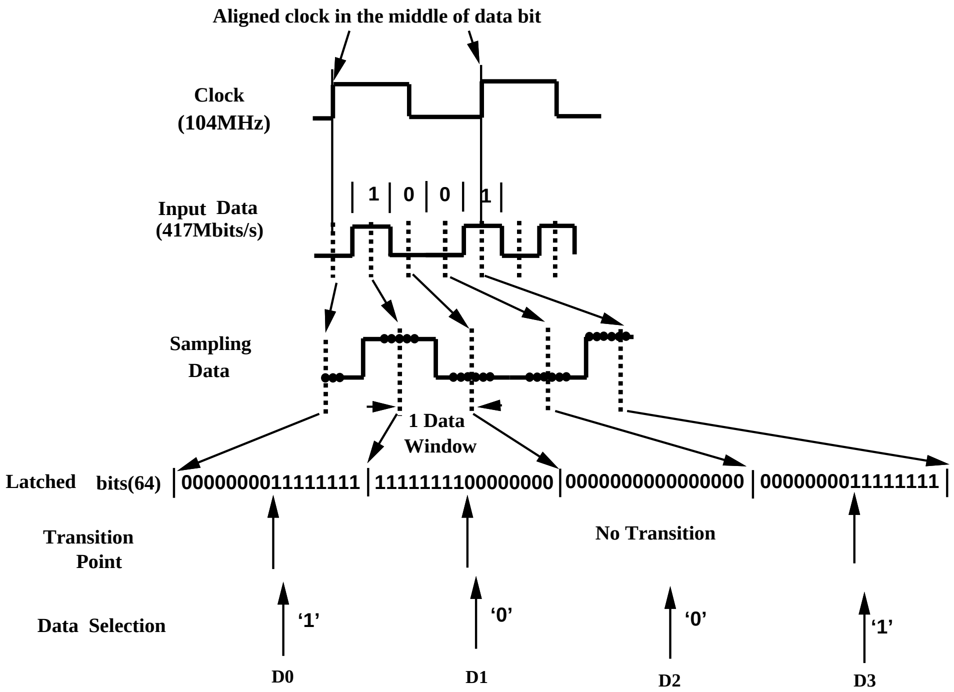


Figure 2: Waveforms in the block diagram

rising edge. Adjacent latches(i and $i-1$) thus sample the digital data at an effective interval of $\Delta t = D_c - D_d$. If all Δt are equal and $\Delta t > 0$, the digital data are sampled uniformly at this resolution. Because the resolution is controlled by the difference of the matched delays, very fine sampling resolution can be achieved. It allows operation at a higher speed data rate than conventional methods. The data rate limits in this technique are the input data bandwidth of the sampler and controllability of data delay unit.

In this method, the sampling clock period can be much slower than the data rate due to two reasons: first, the actual sampling rate is determined by the relative skew between two delay elements. Second, the clock period is determined by the number of windows in the system's implementation, *e.g.*, 1:4 or 1:8 demultiplexed data recovery. The clock rate thus can be four or eight times slower than data rate. However by propagating both the data and clock through two different delay units and having a finite number of sampling stages, there are some sampling points which will not be sampled after the clock propagates through the sampling stages if the appropriate timing requirement of delay units and the clock period is not satisfied.

The time to pass the sampling clock through the clock delay chain is $N \times D_c$, where N is the number of sampling stages. In the 1:4 demultiplexed data recovery circuit, the total delay time in clock delay chain should be $4 \times T$, where T is the clock period. As stated before, if D_c and D_d are the unit delay times of the clock and data respectively, the resolution is

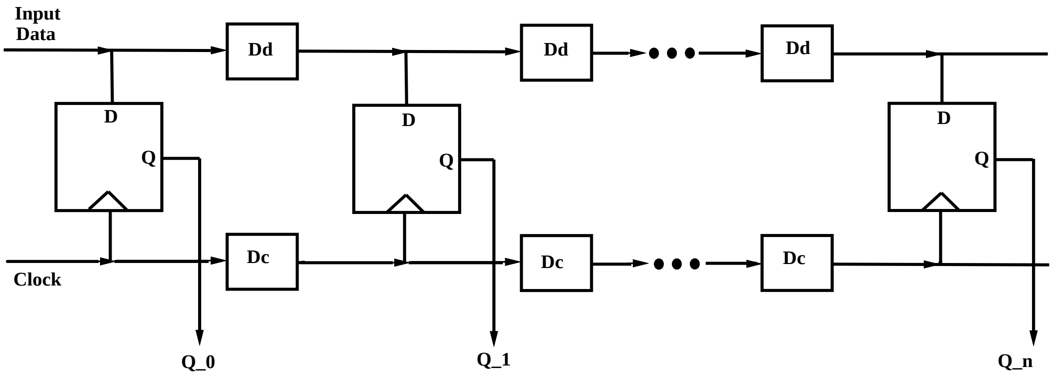


Figure 3: Block diagram of the matched delay sampler

$\Delta t = D_c - D_d$. It is obvious that if a sampling clock edge reaches the first latch at time $t=0$, its final sample at the Nth sampling stage is the data at time $t = (N-1)\Delta t$, and the next clock must arrive at the first latch at $t = N\Delta t$ in order to continuously sample incoming data. Thus the sampling clock period, T, must have values for D_c and D_d that make $T = N\Delta t$. For cyclic operation, $T = MD_c$ where M is the number of oversampling points on one bit input data. In the 625Mbs (SONET-OC12) data recovery application with 16 times oversampling of each bit, T = 6.4ns, N = 64, M=16, D_c = 400ps and D_d = 300ps are required. Thus Δt (resolution) is 100ps in order for one bit NRZ input data (pulse width=1.6ns) to be oversampled at 16 sampled points. In other words, a window for one bit input data is composed of 16 samples. Therefore, in each clock cycle four windows, a total of 64 sampled points, are available for parallel processing. This enables a 1:4 demultiplexing type of data recovery. For different data rate, the clock period, D_c, and D_d could be modified. For 417Mbs data recovery, the parameters are changed to T = 9.6ns, D_c = 600ps, and D_d = 450ps with a resolution of 150ps.

Since multiple periods of the data and clock (four clocks) are present in the sampler, there is the danger that latched data can be overwritten before they are processed by the decision logic. This could be overcome by shifting latched data to another latching stage and thus it is possible to sample and store the input data continuously. The continuous data sampling scheme [12] using a FIFO structure for the matched delay sampler is designed so that no data samples are lost. More details about the continuous sampling and parallel latching are described in next section. At the end of this continuous data sampling block, the oversampled digital data is latched in parallel at the input clock rate for digital processing.

Data recovery is done by locating the transition point (input data boundary) in each data window. A noise filtering function is used to eliminate any single isolated noise sample and metastability effect in the sampler. Selecting one of the 16 oversampled points based on the transition point recovers data. If there is no transition in a window, the bit derived by the previously latched transition point is taken as the recovered data. The encoded transition points also give phase information used to synchronize the clock with the input data. This digital phase value is converted to an analog signal and controls the variable delay on the clock input stage to the sampler. Since the circuit is a closed loop structure, stability issues must be addressed. This system will be shown mathematically to be stable as described in Section 4.

3 Circuit Implementation

3.1 The Matched Delay Sampler

The matched delay sampler has been designed in 1.2μm CMOS technology with a sampling resolution externally adjustable between 25ps and 250ps [13]. As the data rate increases, the jitter and noise occurring in the sampler could cause problems. Differential circuits in the data delay line are used to improve the power supply noise rejection and thus reduce jitter. There are three main components of the sampler: a clock delay unit, a data delay unit, and a D flip/flop. The data recovery circuit contains 64 sampling stages. A circuit diagram of the single sampling stage is shown in Fig.4. The clock delay unit is implemented using two biased CMOS inverters whose delays are controllable by a single bias voltage. A CMOS differential delay circuit whose delay is controlled by two bias voltages was used as the data delay unit. Circuit schematics of the delay elements are shown in Fig.5. The range of the clock delay is designed to be adjustable from 300ps - 1.2ns and the data delay is from 200ps - 600ps. An additional consideration in the design of the data delay is the ability to pass high speed data with symmetric rise and fall times. The input data rate capability is determined, first, by the bandwidth of the data delay chain in the sampler. Then, the resolution is determined by the clock and data delay values. Also, the other constraints (clock period and number of sampling stages described in Section 2) must be met for correct data recovery operation at a specific data rate.

A bistable edge trigged D flip-flop was used to sample the differential data and is shown in Fig. 6. Metastability is of concern since data may be sampled during transition. However, the clock period (9.6ns) is long enough to resolve from any metastable state in the D flip/flop. More details on this flip-flop can be found in reference [13].

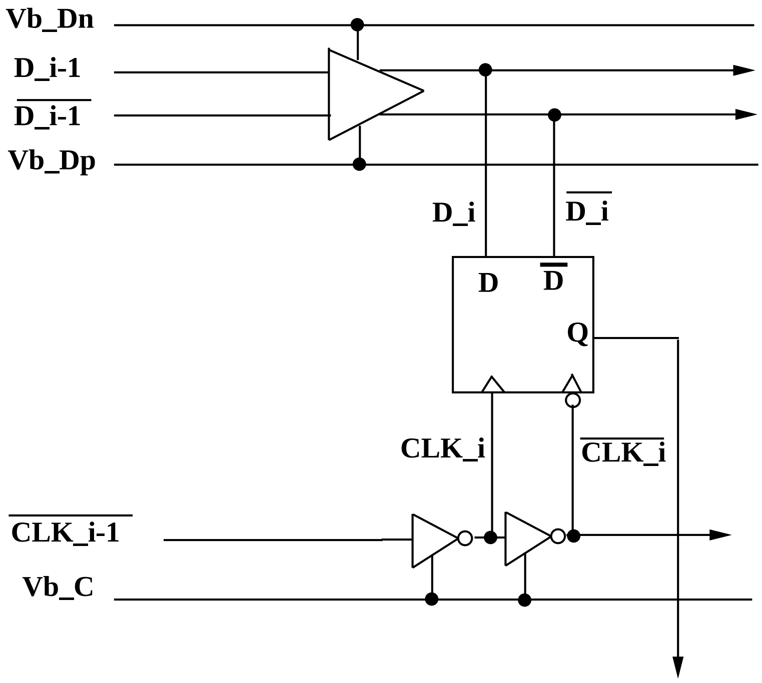


Figure 4: Single stage of the sampler

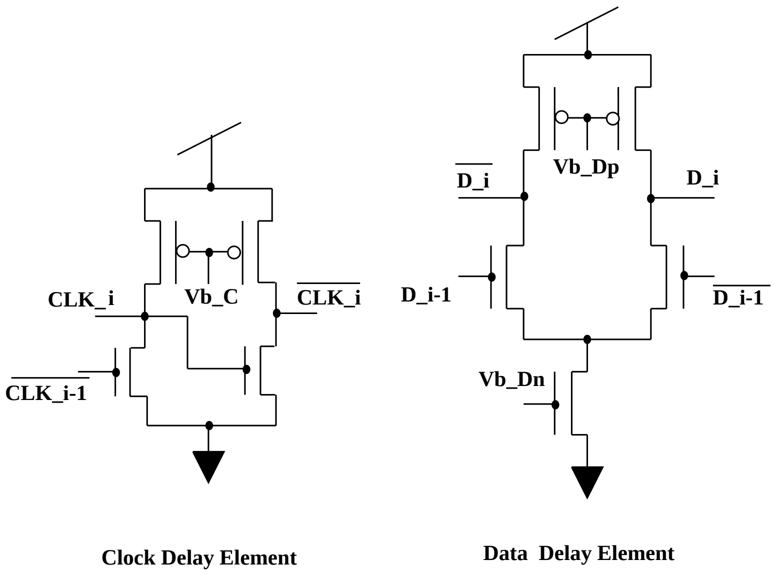


Figure 5: Delay circuits in the sampler

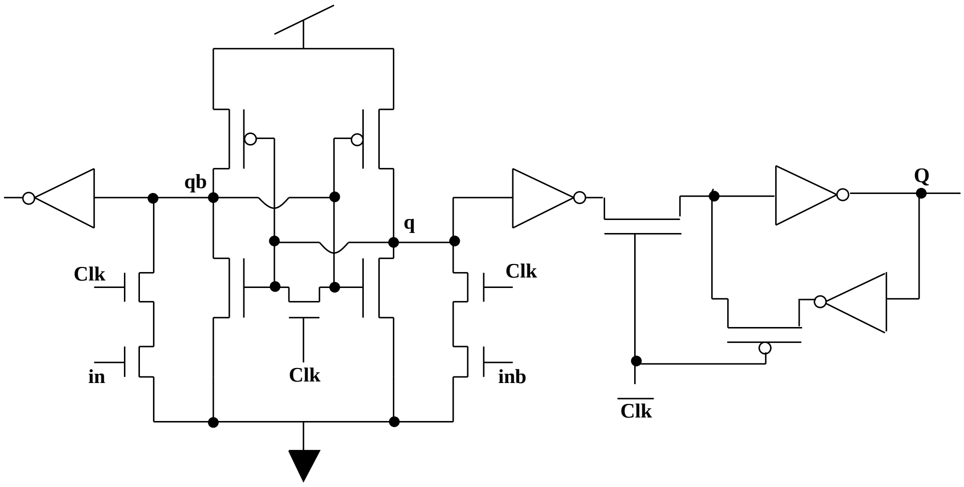


Figure 6: D flip-flop circuit

3.2 Continuous Sampling and Parallel Register

The data recovery circuit has 64 sampling stages. Therefore, to pass the clock signal through the clock delay chain, it takes $64 \times D_c$, where D_c is the delay of a single clock delay element. By setting D_c=600ps and D_d=450ps, incoming data of 417Mbs data rate is sampled at a resolution of 150ps. In the 1:4 demultiplexing data recovery application, the total lapse time in clock delay chain is $4 \times T_{clk}$, which means four clocks are present in the clock delay chain of the sampler. Due to the presence of multiple clocks in the sampler, the sampled points could be overwritten before proceeding to the data processing block. To avoid this, a FIFO structure is used to store sampled data before is is processed. This FIFO is illustrated in Fig. 7.

The oversampled serial bits needs to be converted into parallel data before they can be processed further. In order to latch the oversampled bits in parallel, they must first be aligned in time (de-skewed). Implementation of different delay units at every sampling stage for de-skewing, however, would require a very large area. Instead of implementing area-consuming delay units on de-skewing lines, intermediate latch stages were used in a FIFO structure, as shown in Fig. 7. By clocking within a safe window period during which all oversampled bits are available, the de-skewed data can be latched simultaneously without violation of setup and hold time requirement.

As shown in Fig. 8, the first eight sampled points are latched by the clock $C2$. Once the sampled bit is latched, its duration time is 9.6ns, which is the sampling clock period. The time period that eight sampled bits are present at the same time is thus 4.8ns because the sampled bits are latched serially with the time skew of D_c(600ps). Considering propagation delay time of D flip-flop and process variations, it is optimal to place rising edge of $C2$ in

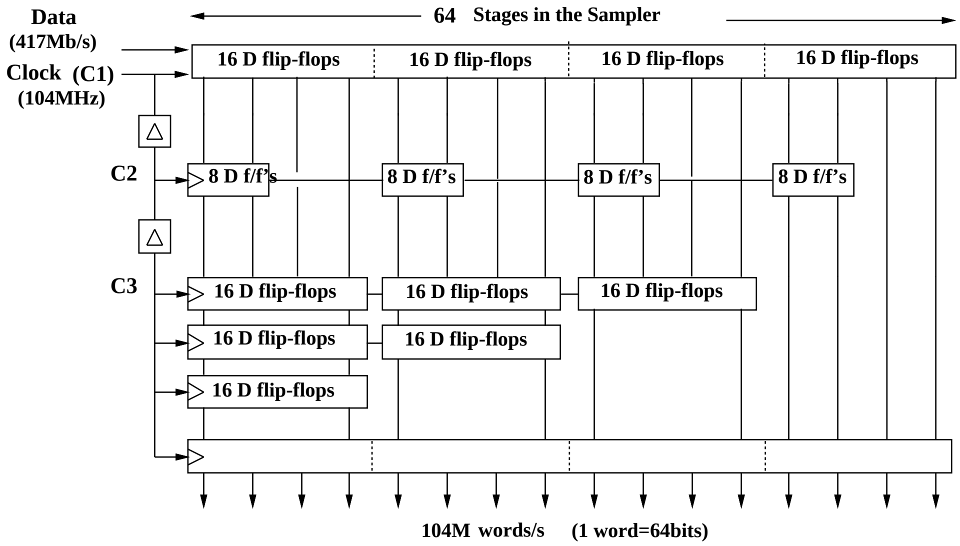


Figure 7: Block diagram of continuous sampling and parallel latch

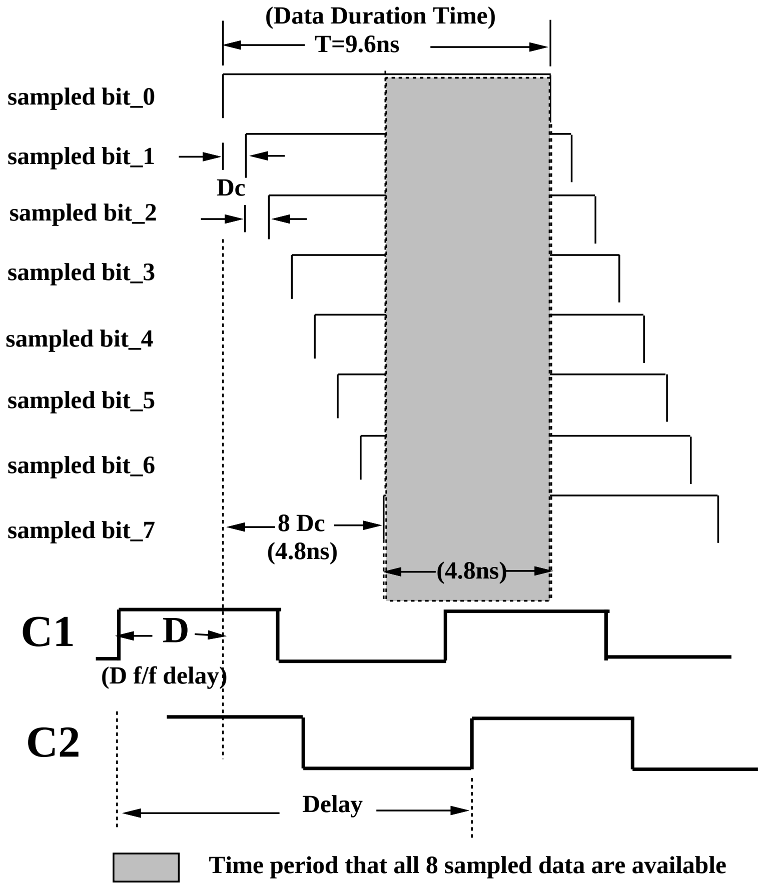


Figure 8: Clocking scheme for continuous sampling and parallel latch

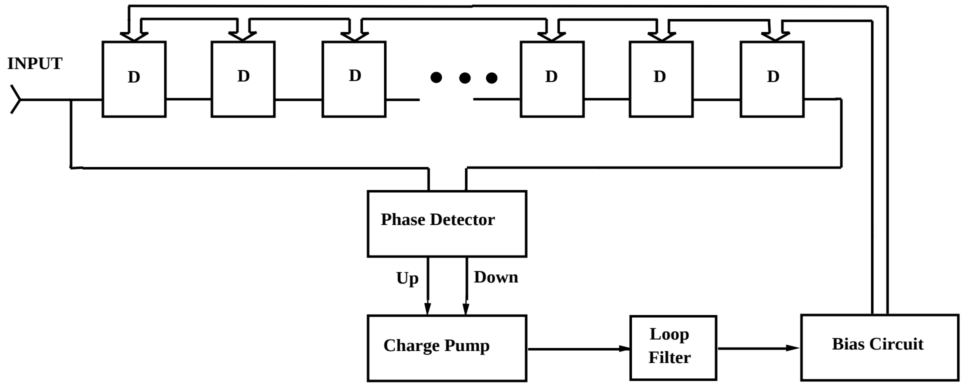


Figure 9: Block diagram of DLL in clock delay chain

the middle of that time period. So, the relationship between $C1$ and $C2$ is given as

$$C2 = C1 + Delay$$

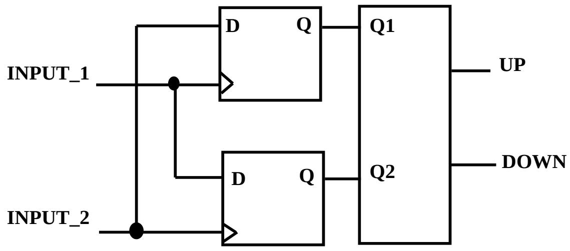
where $Delay$=4.8ns(8 sampled data skew)+4.8ns/2 + Δ(delay of D flip-flop).

Likewise, the $C3$ clock edge is to occur in the middle of the window where the next 8 sampled bits are present at the same time. So, the $C3$ should be placed after the same delay time from $C2$, *i.e.*, $C3 = C2 + Delay$. The $Delay$ can be realized with a fixed delay circuit and a clock driver. The first parallelized 64 samples from the four input data stream are available for digital processing about 58ns (four clock cycles plus two $Delay$'s) after input data are provided to the sampler. Since then, every 64 sampled points are ready every clock period for the next stage processing.

3.3 Delay Locked Loop

For high speed data recovery using the matched delay sampler, it is essential to maintain the nominal delay times in the sampler throughout variations in chip conditions. Fig. 1 shows the sampler structure with delay locked loops (DLL). Each DLL is composed of a phase detector, charge pump, loop filter. The filtered control voltage is the bias voltage which directly modulates the delay produced by the differential delay elements. A block diagram of the DLL used by the clock delay chain is shown in Fig. 9.

Since the lapse time of the clock delay chain and the data delay chain is $4 \times T$ and $3 \times T$ respectively, the phase of the signal entering the delay stages and the signal coming out of the last stage should be the same as long as the delay unit stays at the nominal delay time. The total delay time of the delay chains other than $4 \times T$ (clock delay chain) or $3 \times T$ (data



Q1	Q2	UP	DOWN	Control Voltage
0	1	1	0	↑ (+)
1	0	0	1	↓ (-)
1	1	0	0	—
0	0	0	0	—

Figure 10: Block diagram of Phase detector

delay chain) can be avoided by externally setting initial bias voltages near nominal delays. If there is a phase difference between in-coming and out-going signal of the delay chains, the delay for the delay units is adjusted to resynchronize them. The phase detector compares the rising edges of its inputs and generates "up" and "down" signals, which control the charge pump. This is illustrated in Fig. 10. A modified true single-phase clock D flip-flop [14] was used for the phase detector in order to achieve high-frequency operation and very small phase difference detectability. The charge pump's output is sent through the loop filter and then to the delay unit's bias inputs. Simulations show it takes about 20 ns for the clock delay chain to lock to the 600ps of initial phase error with an external loop filter capacitor of 20pF. The data delay element locking is more difficult than the clock delay element locking since input data is random and there might be no reference rising edge for phase comparison. Therefore, a rising edge detection function in the phase detector is required. The phase detector compares the phases only when there are rising edges at both the first and last stage of the data delay chain. However, in reality, this situation may not be arisen if line coding is used. The voltage output from the data DLL is converted into two bias voltages for the delay control of the data delay chain.

3.4 Digital Phase Control and Data Recovery

After oversampled bits are latched parallelly, digital phase capture and data recovery are done by digital processing. The digital processing block consists of a single bit noise filter for oversampled bits, a transition detector, 16 to 4 encoder, data selector and phase control logic, as shown in Fig. 11. The filter smoothes out an isolated single bit caused by a noise or metastability in the sampler. The transition detector is composed of 64 XOR gates and there are 16 XORs for each data window. Whenever '01' or '10' bit pairs occur in a data window, the transition detector encodes the transition position into four bit information. Based on the transition point, the data selector chooses the correct recovered data. The encoded transition position is used in the phase control logic for phase locking to input data.

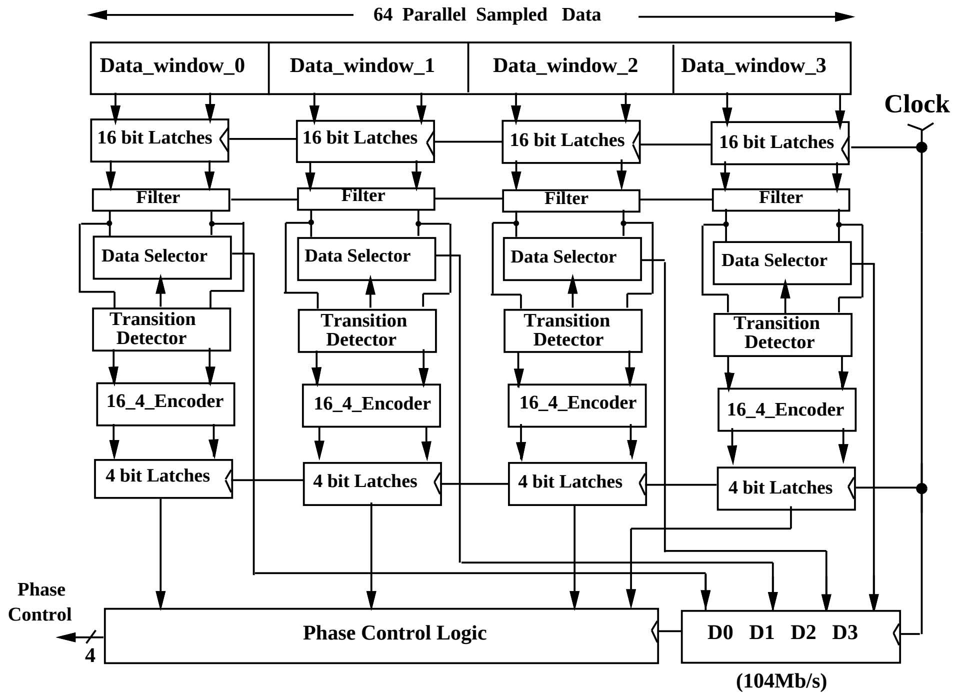


Figure 11: Digital Phase Lock Loop and Data recovery block diagram

As described, the 16 bits from transition detector in each data window are encoded into four bit phase information. In a jitter-free operation there is at most one transition point in each data window because of NRZ input data. At steady state the transition point will be placed in the mid-point of the 16 samples in one bit data window. The data recovery is done by picking one of oversampled bits based on the transition information. If there is no transition point in a data window, any sample in the window can serve as a recovered data. In the case of no transition, for evaluating the phase information, the previous latched encoded position will be used in the phase control logic. Therefore the new phase information will be updated only when a transition point is detected in the data window.

Jittered data might cause an error for data recovery when the transition occurs in the last two points of the 16 oversampled points in a data window. Thus for the robust data recovery, the data selecting point is offset by 2 position from the transition. As stated before, ordinarily, there is at most one transition in a data window. However, it is possible that two transitions could be detected for jittered input data. In order to handle two transitions in a data window, a 16 bit window is divided into two eight bit sections (left half and right half). By masking out the second transition position, the recovered data and four bit phase value are obtained based on the first transition position. The algorithm makes the system more tolerable to jitter and noise. Fig. 12 shows the transition detector, data selector and encoder blocks in a data window in detail. Fig. 13 illustrates how the data is recovered by

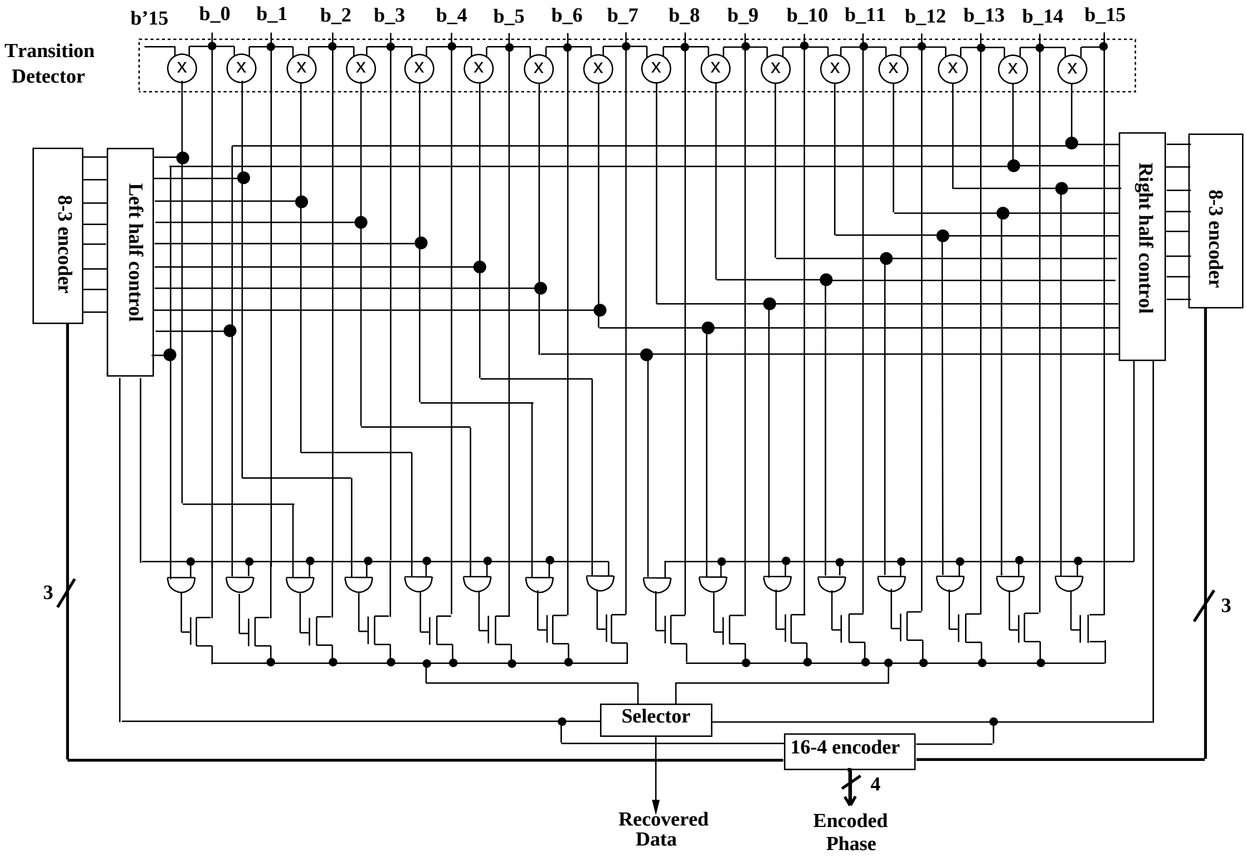


Figure 12: Circuit blocks for transition detector, data selector and encoder in a data window

the adopted algorithm in jittered input data with a transition near window boundary (a), jittered input data with two transitions (b) in a data window and normal operation (c).

The phase control logic controls the phase of the input clock. This will ensure the data boundary is placed in the middle of one bit data window. Since four bit input data are phase-encoded in each cycle, four transitions are generated. The final value is obtained by averaging the four encoded phase values. The digital phase correction information is converted to an analog signal that controls a variable delay circuit. A charge pump type DAC was implemented for generating an analog control voltage for phase correction. The amount of phase correction is done under the stable loop condition of the data recovery circuit. The derivation of phase correction for the stable operation is described in Section 4.

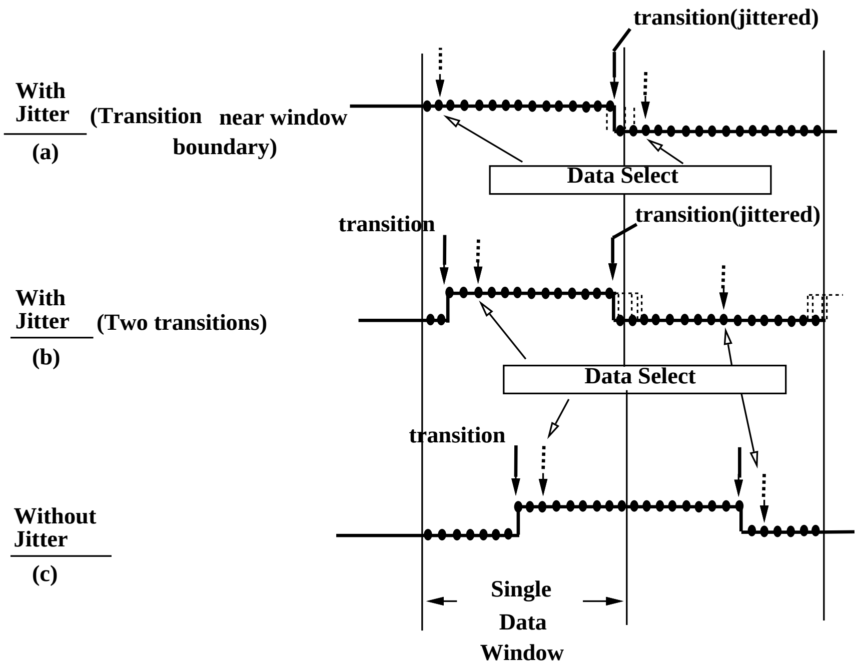


Figure 13: Algorithm for Data Recovery in jittered input data

3.5 On-Chip Test Structure

To avoid the difficulties associated with very high-speed I/O, a high-speed data generator was implemented on-chip, which multiplex of four slow speed (104Mbs) signals to generate a single high speed (417Mbs) signal. The circuit diagram for this data generator is shown in Fig. 14 [15]. Four phase clocks are provided from an external clock signal generator with a 2.4ns phase difference between clocks.

We also placed on-chip probe pads to provide a high-speed input directly to the sampler, bypassing the data generator in the event it malfunctions.

The bias voltage controls for the delay chains were designed to be controllable externally as well. Two types of VCO - a differential ring oscillator type and a XOR tree type - were designed for seeking a possibility to be used as a recovered clock signal. Both operating frequencies are controlled by a bias voltage on each differential circuit.

Considering wiring capacitance in the test board and power consumption in the output driver, the output driver was designed to drive a 200pF load at bit rates up to 80Mb/s. To meet the output driver capability, 1:4 demultiplexed outputs are 1:2 demultiplexed again before the outputs arrive the output drivers. Consequently, the data recovery circuit behaves as a 1:8 demultiplexing and synchronizing circuit when the chip is tested.

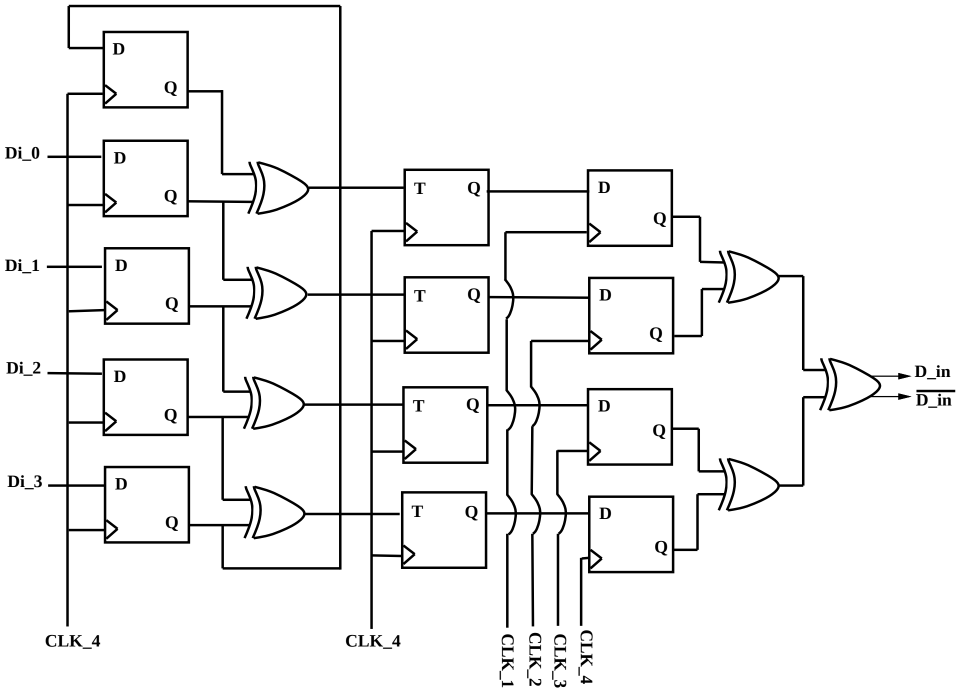


Figure 14: Circuit diagram for a high-speed data generator

4 Stability Analysis of Data Recovery Circuit

Due to the discrete nature of a data recovery circuit using a matched delay sampling technique, its operation must be described by a difference equation. In this section, the $z-$transform technique is employed to analyze the general behavior of the data recovery circuit. In this analysis, the phase error is assumed to be small and the general nonlinear difference can be approximated by a linear one which can be solved by the $z-$transform technique. Three additional factors should be considered for the stability modeling:

1. The delay is controlled by a variable delay circuit

2. Multiple clock periods for data sampling are present simultaneously (functioning as multiple pipelined stages)

3. The pipelined structure of the digital phase locked loop.

Suppose that a clock signal, $C(t)$, and a data signal, $D(t)$, are provided to the sampler. Suppose further that those signals are periodic with period T, and in their transmission synchronization is lost. In Fig.15, we show the clock signal leading the data signal at time kT by $\Delta(kT)$. Now we have access to a controllable (variable) delay τ that can be adjusted every T seconds, and that the clock signal is routed through as shown in Fig.16. Suppose the synchronization error, $\epsilon(kT)$, is available MT time units later, then, at time $t = kT$, the delay value is set at $\tau[(k-1)T]$ and the signals $C(t-\tau)$ and $D(t)$ are out of synchronization

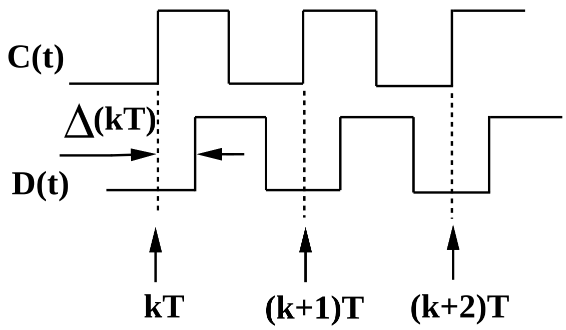


Figure 15: Clock signal leads data signal at time kT

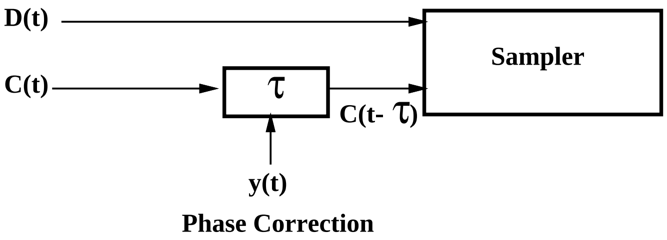


Figure 16: Model for variable delay control of clock signal

by

$$\Delta(kT) - \tau[(k-1)T] = \epsilon(k+M)T \tag{1}$$

In the data recovery circuit five clock cycles (four cycles to propagate through the sampler stages plus one cycle for parallel latching, thus $M = 5$) later than the input data and clock is provided to the sampler, the synchronization error, ϵ, will be available. Four synchronization errors obtained from four bit input data can be derived as:

$$\epsilon_0(kT) = \Delta(k-M)T - \tau(k-M-1)T \tag{2}$$

$$\epsilon_1(kT) = \Delta(k-M+1)T - \tau(k-M)T \tag{3}$$

$$\epsilon_2(kT) = \Delta(k-M+2)T - \tau(k-M+1)T \tag{4}$$

$$\epsilon_3(kT) = \Delta(k-M+3)T - \tau(k-M+2)T \tag{5}$$

where ϵ_i is the synchronization error for each section captured from a serial input data in the sampler. The variable delay is corrected by using the most recent four measured values of clock and input data misalignment. In the phase control circuit the average value of four ϵ's is taken. Thus the phase correction $y(kT)$ at time $t = kT$ is

$$y(kT) = \frac{1}{4}[\epsilon_0(k-1)T + \epsilon_1(k-1)T + \epsilon_2(k-1)T + \epsilon_3(k-1)T] \tag{6}$$

The reason that $(k-1)T$ is in the error terms is because one clock cycle is needed for phase correction calculation. From Eq.(2) - Eq.(5) and Eq.(6),

$$\begin{aligned} y(kT) &= \frac{1}{4}[\Delta(k-M-1)T + \Delta(k-M)T + \Delta(k-M+1)T + \Delta(k-M+2)T] \\ &\quad -\frac{1}{4}[\tau(k-M-2)T + \tau(k-M-1)T + \tau(k-M)T + \tau(k-M+1)T] \end{aligned} \tag{7}$$

Implementing the feedback scheme to control $\tau(kT)$ as

$$\tau(kT) = \tau[(k-1)T] + y(kT) \tag{8}$$

So that Eq.(8) becomes

$$\begin{aligned} \tau(kT) &= \tau[(k-1)T] + \frac{1}{4}[\Delta(k-M-1)T + \Delta(k-M)T + \Delta(k-M+1)T + \Delta(k-M+2)T] \\ &\quad -\frac{1}{4}[\tau(k-M-2)T + \tau(k-M-1)T + \tau(k-M)T + \tau(k-M+1)T] \end{aligned} \tag{9}$$

The z-transform of Eq.(9) is

$$\tau(z) = z^{-1}\tau(z) + \frac{1}{4}z^{-M-1}\Delta(z)(1+z+z^2+z^3) - \frac{1}{4}z^{-(M+2)}(1+z+z^2+z^3)\tau(z) \tag{10}$$

$$\tau(z) = \frac{\frac{1}{4}z^{-M-1}(1+z+z^2+z^3)\Delta(z)}{1-z^{-1}+\frac{1}{4}z^{-(M+2)}(1+z+z^2+z^3)} = \frac{\frac{1}{4}z(1+z+z^2+z^3)\Delta(z)}{z^{M+2}-z^{M+1}+\frac{1}{4}(1+z+z^2+z^3)} \tag{11}$$

Plugging M=5 into Eq.(11) gives

$$\tau(z) = \frac{z(1+z+z^2+z^3)\Delta(z)}{z^7-z^6+\frac{1}{4}(1+z+z^2+z^3)} \tag{12}$$

Notice from Eq.(12) that if $\Delta(kT)$ is constant, $\tau = \Delta$ is a steady state solution. However, it shows that two of the seven roots of characteristic polynomial lie outside of the unit circle, implying unstable behavior. Also, if $\Delta(kT)$ is not a constant, but rather a time varying or random process, then tracking performance could degrade.

To prevent an unstable feedback loop, we implemented a digital gain control block. The feedback error correction value becomes K (a constant value) times the phase error between the clock and the data. The K value should be less than $\frac{1}{3}$ to make the loop stable and can be realized either by a fixed value(e.g. $\frac{1}{4}$ or $\frac{1}{8}$) or a programmable sequence[7]. In the data recovery circuit a fixed K is controlled by a charge pump type DAC circuit. This gain control step changes Eq. (6) to:

$$y(kT) = K\frac{1}{4}[\epsilon_0(k-1)T + \epsilon_1(k-1)T + \epsilon_2(k-1)T + \epsilon_3(k-1)T] \tag{13}$$

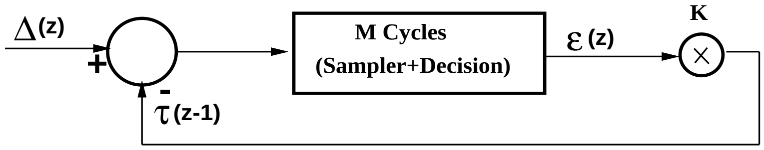


Figure 17: Loop model of the DPLL

Also Eq.(8) becomes

$$\tau(kT) = \tau[(k-1)T] + K\frac{1}{4}[\epsilon(k-1)T + \epsilon(k-1)T + \epsilon(k-1)T + \epsilon(k-1)T] \qquad (14)$$

and Eq.(11) becomes

$$\tau(z) = \frac{\frac{K}{4}z(1+z+z^2+z^3)\Delta(z)}{z^{M+2} - z^{M+1} + \frac{K}{4}(1+z+z^2+z^3)} \qquad (15)$$

If $0 < K < \frac{1}{3}$, all the poles are located inside the unit circle. Therefore, the system is stable.

Another important issue in a DPLL system is *static phase error*. The feedback system should be guaranteed to ensure that the loop has zero static phase offset. The loop model can be drawn as Fig.17.

From Eq.(1)

$$\Delta(z) - z^{-1}\tau(z) = z^M\epsilon(z) \qquad (16)$$

Thus

$$\frac{\epsilon(z)}{\Delta(z)} = z^{-M}(1 - \frac{\tau(z)}{\Delta(z)}z^{-1}) \qquad (17)$$

From Eq.(15)

$$\frac{\epsilon(z)}{\Delta(z)} = z^{-M}[1 - \frac{\frac{K}{4}z(1+z+z^2+z^3)}{z^{M+2} - z^{M+1} + \frac{K}{4}(1+z+z^2+z^3)}] = \frac{z(z-1)}{z^{M+2} - z^{M+1} + \frac{K}{4}(1+z+z^2+z^3)} \qquad (18)$$

Using the final theorem, the final value of the error will be expressed as Eq.(19) for the step phase change.

$$\lim_{z \to 1} \frac{\epsilon(z)}{\Delta(z)} z = 0 \qquad (19)$$

Taking the limit as z approaches 1, the step phase error approaches zero.

5 Experimental Results

A prototype high speed data recovery circuit has been designed and fabricated. The chip contains about 14,662 transistors and has an area of approximately $10.8mm^2$ in MOSIS

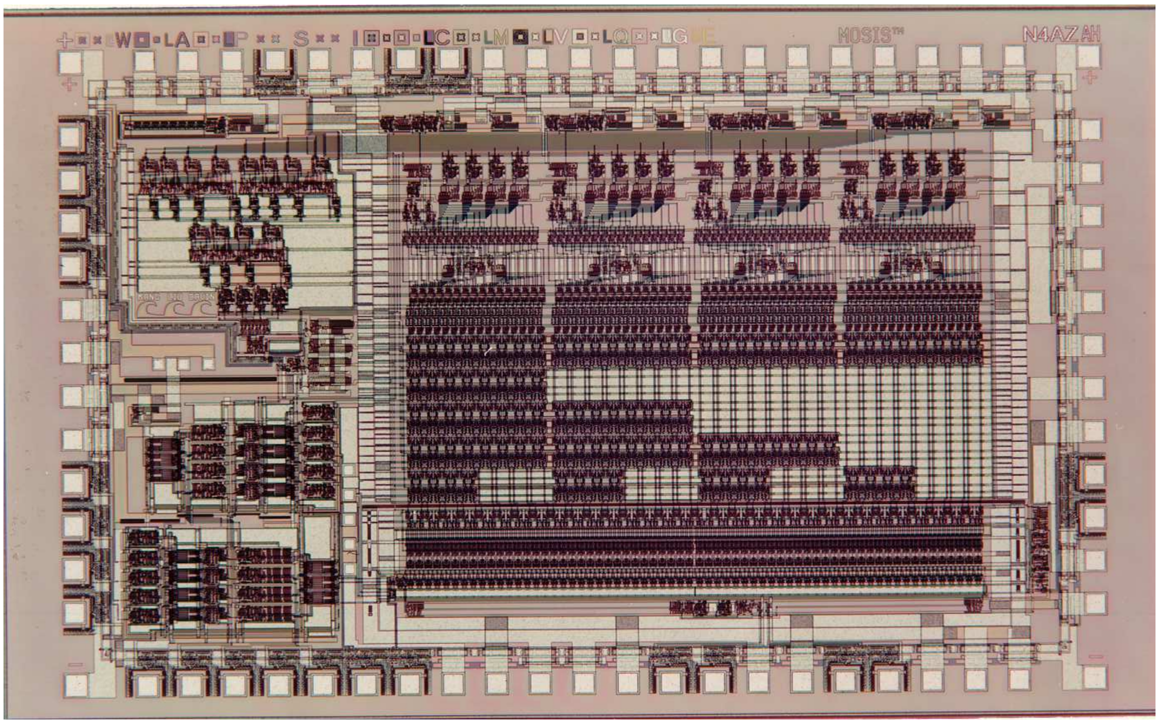


Figure 18: Photomicrograph of data recovery chip

1.2μm CMOS technology. A photomicrograph of the fabricated chip appears in Fig. 18. The design and performance statistics are summarized in Table 1.

The circuit was simulated by CAzM [16], a SPICE-like circuit simulator before fabrication. Physical design was done with Magic and simulations were run on the extracted layout. Simulations show that 625Mb/s input data was successfully recovered into 1:4 demultiplexed 156Mb/s output streams at 5V single power supply. Actual testing, though, was done under 4V power supply due to the voltage limitation of test equipment such as HP8133A. Two HP8133A pattern generators were used to provide the clock and input data at a maximal voltage swing of 3.3V. Consequently this made the power supply down to 4 V. The reduced power supply resulted in the data recovery circuit operating at 417Mb/s input data rate instead of 625Mb/s. The reason is that the current drivability in the delay cells in the sampler is reduced by the reduced power supply, which increases the delay of the delay units and decreases the maximum input data rate. The high speed serial input data was provided through the on-chip's probe pads and 8 channel output was taken with a HP1662 logic analyzer and a Tektronix 11801 digitizing oscilloscope.

The clock delay *vs.* clock bias voltage was measured and plotted in Fig.19. Because the clock is being locked at a certain frequency by the DLL, the clock delay is calculated from the clock period divided 16. The data delay *vs.* bias voltages was measured and plotted

Table 1: Design and Performance Statistics

Technology	1.2μm CMOS
Transistors	14662
Die size	4350μm$\times$ 6690μm
Total I/O pins	64 pin PGA
Input Data Rate	417Mb/s
Sampling Clock Rate	104MHz
Output Data(8 Channel)	52Mb/s
Sampling Resolution	150 ps
Power Dissipation	800mW at 4V
jitter(peak to peak)	120ps

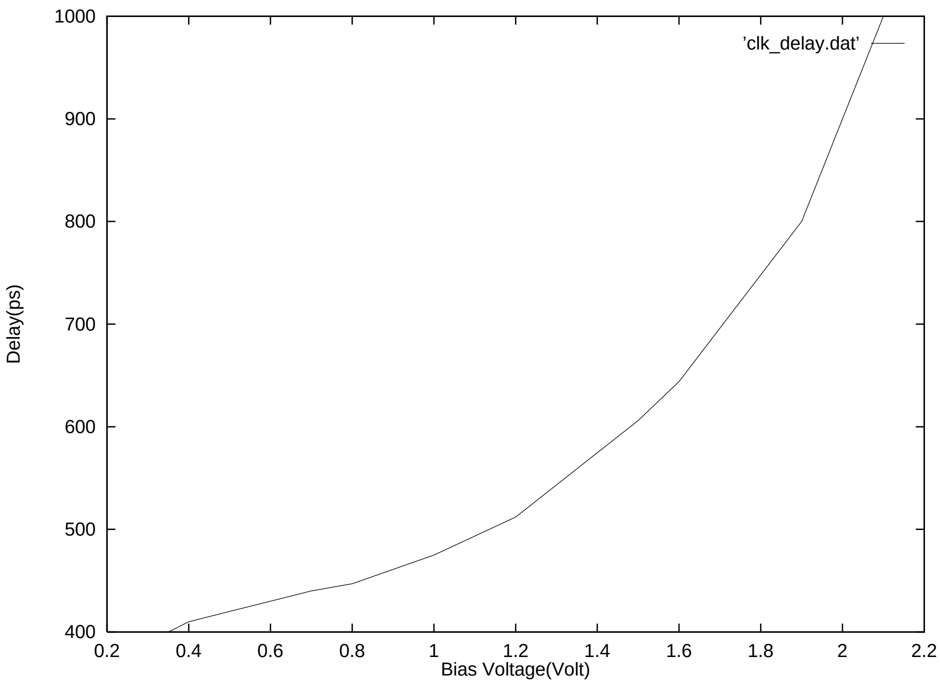


Figure 19: Measured clock delay *vs.* bias voltage

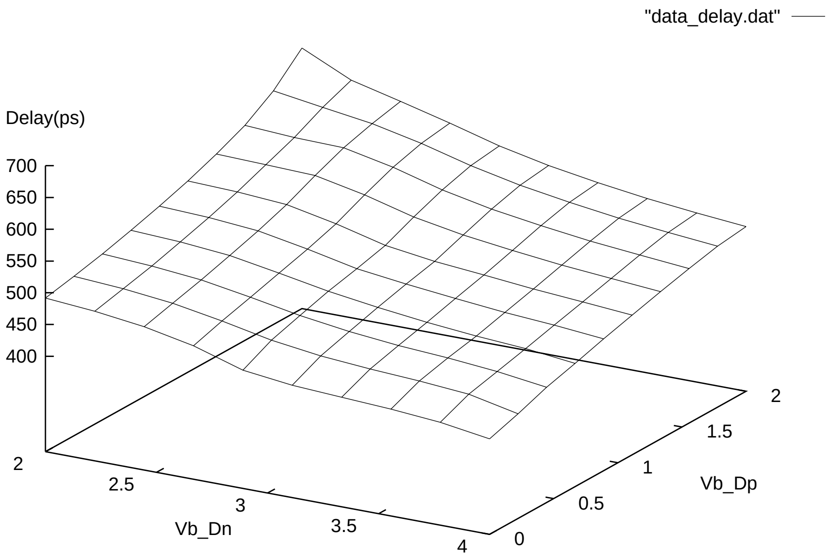


Figure 20: Measured data delay *vs.* bias voltages

in Fig. 20. Due to failure of data delay locking through the data DLL, the data delay bias voltages were controlled externally. To measure the data delay, we provided different input patterns at different data rates and adjusted the bias voltages externally until the data output was correct. Because the clock delay is already known, the data delay can be calculated.

The measured output signals in each channel with an input data rate of 417Mb/s are shown in Fig. 21. Waveform degradation comes from a wire inductance and power supply noise. Fig. 22 shows a measured output in all channels through a HP1662 logic analyzer with an input pattern of '0101...' at 417Mb/s. The measured outputs in each channel (labeled as $Lab1_0$ to $Lab1_7$) are '0' in channel 1 ($Lab1_0$ and $Lab1_1$), '1' in channel 2 ($Lab1_2$ and $Lab1_3$), '0' in channel 3 and '0' in channel 4.

To measure jitter in the data recovery circuit, two different methods are involved. The transition edge of the input data can be moved by controlling a external variable delay on the data signal generator. First, the time interval until the output starts to show error bit recovery was measured. Then by comparing the measured time interval with the expected valid time window, jitter was calculated. In the 416.7Mbits/s data (2.4ns NRZ) the expected nominal valid window time is 2.4ns. The average measured time interval of correct data recovery was 2.28ns, which indicates jitter of about 120ps. Another method to measure jitter in the circuit is that by moving the clock transition edge near the one bit data sampling window, the time interval can be measured between stable data recovery outputs.

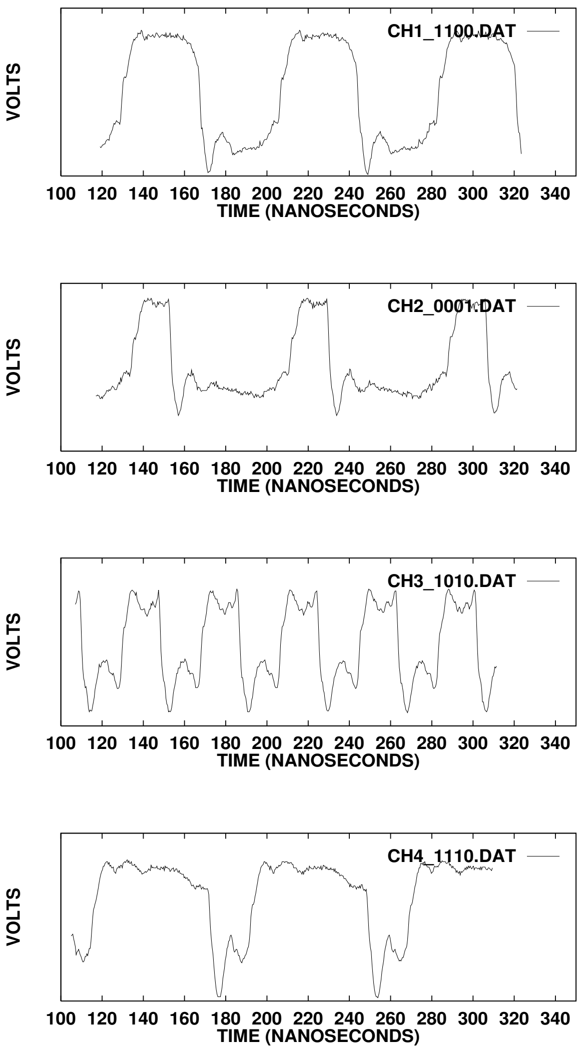


Figure 21: Measured output waveforms in each channel

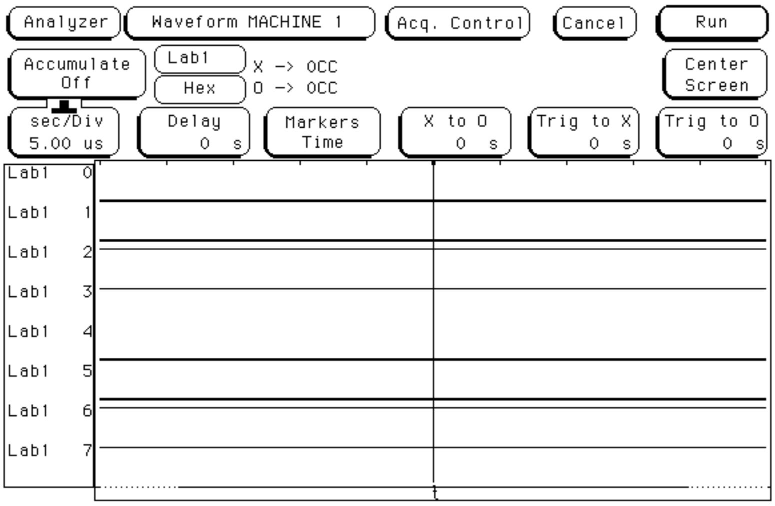


Figure 22: Output waveform measured by HP1662 with input'0101...'

The measured jitter in the latter way also shows about 120ps. By introducing intentional jitter using the data pulse width control function in the external signal generator, the jitter tolerance of the chip was tested. A minimum pulse width of 1.2ns input data (50% of nominal data pulse width) was recovered successfully. Therefore jitter tolerance of the circuit is 1.2ns.

A VCO circuit with guard ring was implemented for testing the effectiveness of the guard ring for noise reduction relating to the jitter. The measured jitter did not show any dependence on whether the VCO was on and off. This shows that a guard ring is a necessary block for mixed mode circuit to reduce noise. The jitter as a function of the power supply was also measured. Jitter becomes less pronounced as the power supply increases.

At the rate of 417 Mbits/s the tolerable error range of the sampling input data is $\pm$1.2ns, which is 8 sampling point error tolerance. If there are more or less than 8 sampling points from a total of 64 sampling points in any bit of four input data streams, it is possible that data bit could be mistakenly recovered. Therefore careful design is required so that the variable delay units are controlled under the tolerance. Due to the 1:8 demultiplexed structure, bit error rate of the chip could not be quantified. Instead, the output patterns for different inputs were tested and in all case, all data were recovered successfully.

6 Conclusions

A data recovery circuit using the matched delay sampling technique has been presented. By utilizing the fine sampling resolution in the sampler, which is limited by a difference in delays of data and clock delay elements, high speed data recovery without a high-speed clock has been achieved. The data recovery circuit using an external clock at 1/4 of the data frequency was designed and fabricated in MOSIS 1.2 μm CMOS. Test results show that the circuit can take 417Mb/s NRZ input data and makes a 1:4 demultiplexing of serial input data into four output streams in parallel with 800mW power consumption at 4V power supply. Clock delay elements were controlled by bias voltages generated by a delay locked loop circuit and data delay elements were controlled by external adjustment.

The scheme could be modified as a high speed data/clock recovery circuit if a clock generated from an on-chip VCO is used. This capability is needed if the applications spanned over various data rates which would be required for a transceiver design. The data recovery scheme with the matched delay sampler also shows that it is feasible to handle higher data rates of over 1 Gb/s if a more aggressive CMOS technology is used.

References

[1] P. Wallace, R. Bayruns, J. Smith, T. Laverick, and R. Shuster. A GaAs 1.5Gb/s Clock Recovery and Data REtiming Circuit. *ISSCC*, pages 192–193, 1990.

[2] A. Pottbacker and U. Langmann. An 8GHz Silicon Bipolar Clock-Recovery and Data-Regenerator IC. *ISSCC*, pages 116–117, 1994.

[3] L. Lai and R.C. Walker. A Monolithic 622Mb/s Clock Extraction and Data Retiming Circuit. *ISSCC*, pages 144–145, 1991.

[4] S. Enam and A.A. Abidi. NMOS IC's for Clock and Data Regeneration in Gigabit-per-Second Optical Receivers. *ISSCC*, pages 144–145, 1991.

[5] J. Sonntag and R. Leonowich. A monolithic CMOS 10MHz DPLL for Burst-Mode Data Retiming. *IEEE International Solid-State Circuit conference*, pages 194–195, February 1990.

[6] M. Bazes and R. Ashuri. A novel CMOS Digital Clock and Data Decoder. *IEEE Journal of Solid State Circuits*, pages 1934–1940, December 1992.

[7] B. Kim, D. N. Helman, and P. R. Gray. A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2 μm CMOS. *IEEE Journal of Solid State Circuits*, pages 1385–1394, December 1990.

[8] Bin Guo, A. Hsu, Y. Wang, and J. Kubinec. A 125Mbs CMOS all digital Data Transceiver Using Synchronous Uniform Sampling. *ISSCC*, pages 112–113, February 1994.

[9] K. Lee, S. Kim, G. Ahn, and D. Jeong. A CMOS Serial Link for Fully Duplexed Data Communication. *IEEE Journal of Solid State Circuits*, pages 353–364, April 1995.

[10] W. Liu, M. Clements, and R. K. Cavin. The Matched Delay Technique:Theory and Practical Issues. *Tech. Report, NCSU-VLSI-93-23, North Carolina State University*, 1994.

[11] J. Kang, W. Liu, and R. K. Cavin. A Monolithic 625Mb/s Data Recovery Circuit in 1.2μm CMOS. *Custom Integrated Circuits Conference*, pages 625–628, May 1994.

[12] M. Clements, W. Liu, J. Kang, and R. K. Cavin. Continual sampling and storage with a Matched Delay Sampler. *Electronic Letters*, pages 463–465, March 1994.

[13] C.T. Gray, W. Liu, W.A.M. van Noije, T. A. Hughes, and R. K. Cavin. A sampling technique and its CMOS implementation with 1 Gbit/s Bandwidth and 25 ps Resolution. *IEEE Journal of Solid State Circuits*, pages 340–349, March 1994.

[14] J. Yuan and C. Svensson. High Speed CMOS Circuit Technique. *IEEE Journal of Solid State Circuits*, pages 62–70, 1989.

[15] C. Zukowski and K. Shum. A Matched-Delay CMOS TDM Multiplexor Cell. *ICCD*, pages 352–355, 1988.

[16] D. Erdman, D. Rose, and G. Nifong. CAzM: Circuit Analyzer with Macromodeling User's Guide. *Tech. Report. MCNC*, June 1990.