

# The Design and Simulation of Single-bit-data Blind Oversampling Data-recovery Circuit for USB2.0 Interface

Ma Xiaoyan<sup>1</sup>, Zou Hao<sup>2</sup>, Li Yujuan<sup>1</sup> & Xu Tingting<sup>1</sup>

<sup>1</sup> College of Information Engineering, Tai Shan Medical University, Tai'an, Shandong, China

<sup>2</sup> Tai'an power supply company, Tai'an, Shandong, China

Correspondence: Ma Xiaoyan, College of Information Engineering, Tai Shan Medical University, Tai'an 271016, Shandong, China. Tel: 86-135-8383-4589. E-mail: maxiaoyan81@126.com

Received: November 26, 2012

Accepted: December 14, 2012

Online Published: December 25, 2012

doi:10.5539/mas.v7n1p61

URL: <http://dx.doi.org/10.5539/mas.v7n1p61>

## Abstract

In order to propose a new single-bit-data blind oversampling data-recovery circuit for the USB2.0 interface, the circuit is designed and simulated correctly by using dynamic sampling window to sample the signal. Based on the dynamic sampling window, the circuit selects the right data from the 5X sampled data and decides the proper next sampling window on the basis of both the edge of sampled data and the position of current sampling window. The proposed circuit by using dynamic sampling window simplifies the sampling algorithm and reduces the scale of circuit compared to conventional circuits. Moreover, this proposed circuit is available on account of the simulation result at the frequency of 480MHz and test vectors of 1000,000 bit based on 0.18 um CMOS technology library.

**Keywords:** blind oversampling, data-recovery, dynamic sampling window

## 1. Introduction

In modern serial data transmission systems, clock synchronization signals are generally not used for transmission, which are replaced by asynchronous mode between transmitter and receiver, in order to reduce the number of the signal lines. In asynchronous mode, the clock signal is unable to achieve strict synchronization, with a possible solution of using blind oversampling data recovery circuit.

There have been several studies on the design of blind oversampling data recovery circuit. The traditional blind oversampling data recovery circuit was firstly introduced in earlier studies (Yang, Farjad-Rad, & Horowitz, 1998; Kyeongho et al., 1995; Kim & Jeong, 2003). Then Part et al. proposed a relatively new circuit design using data selection of the 5X oversampled data sampled from a single data bit, which could reduce more than half numbers of the transistors (Park et al., 2008). This circuit included an X5 bit Sampler, a coarse data recovery block, an Add/Drop FIFO, and a sampling arbitration circuit. The general principle of this circuit was that the 5-bit oversampled data were sampled from the serial input single-bit data with 5 times sampling frequency by the 5X bit sampler. Corresponding to the phase of level jumping, 1-bit data was chosen from 5-bit input data and then transferred into the FIFO. This Add/Drop FIFO played a very important role in getting an accurate data. Because in practical data transmission, a data may be omitted or repeatedly sampled by the coarse data recovery circuit owing to the signal jitter or the frequency offset between the transmitter and receiver. In the Add/Drop FIFO block, if data recovery circuit missed a sampling data, a new data would be added in Add/Drop FIFO, which was modulated by the control circuit. While if a data was repeatedly sampled, which would be cut off in Add/Drop FIFO. In this way, the data transmission accurate would be guaranteed.

However, blind oversampling data recovery circuit has been challenged by the high transmission speed USB2.0 interface, which is used in communication between computer and peripheral equipment. Currently, USB 2.0 interface has been proved as a relatively simple and reliable interface for short-range data communication and applied in several special integrated chips. And there would be more and more demand for this kind of digital device integrating USB interface. In order to satisfy this increasing demand with a lower cost, reducing the circuit scale of USB2.0 interface would help to cut down the power consumption. Thus a smaller and simpler circuit would be preferred.

In this paper, we propose a more simple single data bit blind oversampling data recovery circuit, with a simulation results in the frequency of 480MHz. In this circuit, the input serial signal is sampled by a dynamic sampling window, and the jump phase of the current data as well as the next dynamic sampling window are determined by the jump phase of both current sampling data and the data in its front-adjacent sampling window. Generally, USB 2.0 protocol uses NRZI coding, which ensures the length of continuous high level will not be longer than 6 bits in the serial data (Compaq & Intel, 2000). Thus the jump data are able to be used in synchronizing transmitter and receiver, also used to control the frequency shift between the transmitter and receiver. Compared to the traditional blind oversampling data recovery circuit and the circuit reported by Park et al, our circuit design and algorithm are relatively simpler and easier, with a much smaller hardware size.

## 2. Circuit Structure

### 2.1 The Traditional Blind Oversampling Data Recovery Circuit

Figure 1 shows the diagram of traditional blind oversampling data recovery circuit in the USB receiver. The circuit samples the input data with 3 times oversampling frequency using a 24-phase-multiphase clock. The 24 bits oversampling data are transmitted into FIFO1 and the edge detector simultaneously. Then the received bit data would be converted into a byte data by a bit/byte controlling synchronizer. The data are processed into 8 bytes in FIFO2 and transmitted to the output as the recovery data. Because there are two FIFO in this circuit, more hardware resources are required.

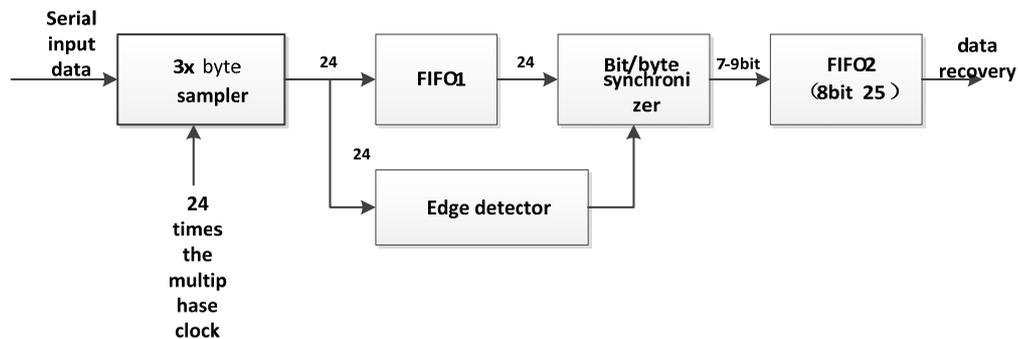


Figure 1. The diagram of traditional blind oversampling data recovery circuit

### 2.2 Blind Oversampling Data Recovery Circuit Reported by Park et al. (2008)

Figure 2 presents the structure of the blind oversampling data recovery circuit reported by Park et al. The serial input data are first sampled with 5X sampling frequency by a 5X sampler. Then this sampling data are transmitted into FIFO by controlling MUX through a phase selector. Considering to the signal jitter and frequency shift between the transmitter and receiver, the sampling data may be lost or repeatedly sampled. Thus an error detector and an Add/Drop FIFO are adopted to add or cut off a data based on the detected error, which in turn to promise an accurate recovery data obtained.

This circuit is simpler than the traditional one. However, the algorithm of both the phase selection and the error detector are relatively complex. Also the implementation of the Add/Drop FIFO is quite difficult.

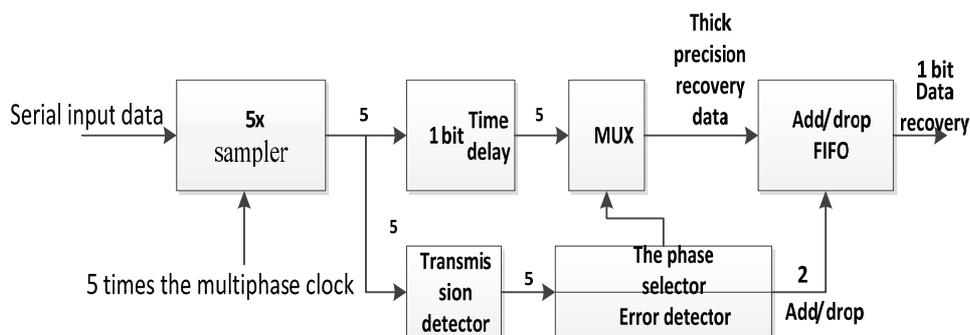


Figure 2. The diagram of blind oversampling data recovery circuit in (Park et al., 2008)

### 2.3 The New Blind Oversampling Data Recovery Circuit in Our Study

The circuit in our study is using a modifying phase sampling algorithm on the basis of the blind oversampling data recovery circuit reported by Park et al. (2008). A dynamic rather than a fixed sampling window is used to sample input signals. This dynamic sampling window is shifted based on the jumping phase of the input data, thus it could be used to reduce or even avoid the errors from data missing or repeatedly sampling owing to the clock synchronization. In this way, the circuit will be simplified by using a general FIFO instead of the Add/Drop one, as well as without the error detector. The structure of this circuit is showed in Figure 3.

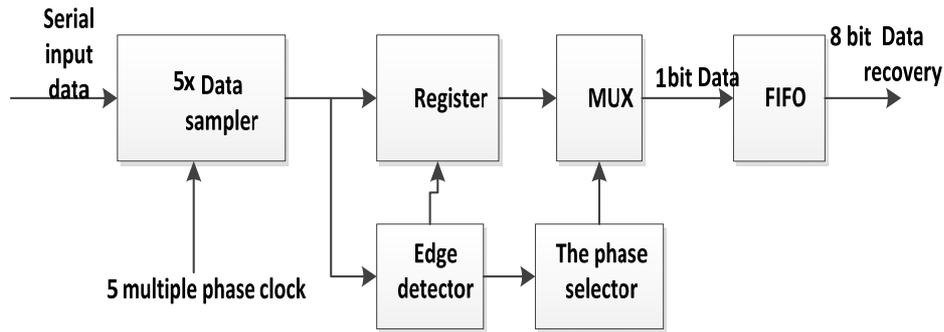


Figure 3. The diagram of a modified blind oversampling data recovery circuit in our study

This circuit includes a 5X data sampler, a shift register, an edge detector, a phase selector, a 7-1 MUX, and a FIFO (convert 1 bit serial data into 8 bit parallel data). Five sampling clocks in 5X data sampler work in the frequency of 480MHz, with the adjacent lock phase difference of  $2\pi/5$ . It is worthy to point that this circuit works in a 5-clock-input-signal alternately sampling mode, which could reduce the high frequency interference without using 2.4 GHz clock signal. In addition, in the sampler, a buffer constituted by two level inverters is used to cushion the sampling results to solve the problem caused by the edge sampling data and get a correct level of the sampling data.

The principle of the circuit is that 1bit serial input data is firstly sampled 5 times at a same time-interval by the 5x bytes sampler, which is used a 5 times multiphase clock as a reference clock. Then the received sampling data are simultaneously transmitted to the edge detector and the shift register. In the edge detector, the last bit of the previous sampling data and the new sampling 5 bits data are stored to calculate the jumping edge position of this 6-bit data. Then the calculation results are transferred into the phase selector. In the shift register, the sampling data are shift stored according to the sampling sequence. This shift register is at least 7 bits to meet the requirement of the dynamic window. Based on the locations of rising and falling edges, as well as the last sampled data position, the current sampling position can be obtained by the phase selector, which is also used to control MUX to choose the right data as an input to the FIFO. Lastly in FIFO, the 1bit data is converted into 8bit data as the recovered data.

### 3. Sampling Data Selection

In this circuit, five clocks are used to sample the serial input data and different shifts are obtained after sampling, with the same shift between adjacent sampling data. Figure 4 shows the sampling principle of the 5X phase clock. Five sampling data (D[0]~D[4]) are received in a sampling window at the rising edge of the clock.

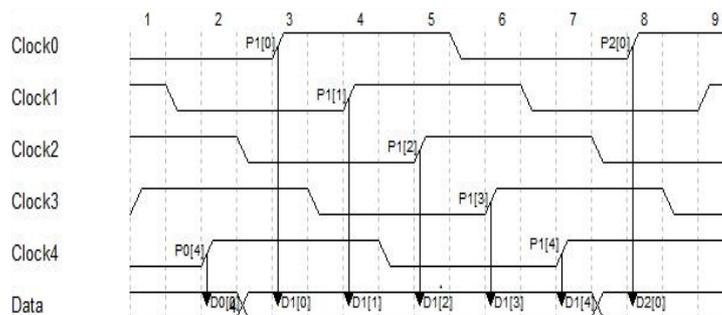


Figure 4. The sequence diagram of the 5X phase clock

The phase selection algorithm in our study is modified from the design reported by Park et al. (2008). In their study, a fixed sampling window was adopted for blind oversampling data recovery circuit. Five clocks, from clock0 to clock 4, sequentially sampled the input data, which would then determine the sampling phase and finally constitute the fixed sampling window P[0]~P[4]. In this design, because the frequency offset between the transmitter and receiver is cumulative, fixed sampling window may contain the adjacent data information, which would make data sampling algorithm more complex and also increase the hardware cost. In our study, the fixed sampling window is replaced by a dynamic one, the effect caused by frequency offset would be partially reduced and most current sampling data would fall into the sampling windows. The position of the current sampling window is determined by the last sampling results. In addition, it is worthy to point that, more data need to be stored in the register using the dynamic sampling window, at least 6 bits, adding 1 bit delay, so 7 bits in total, which can be achieved with a 7-bit shift register.

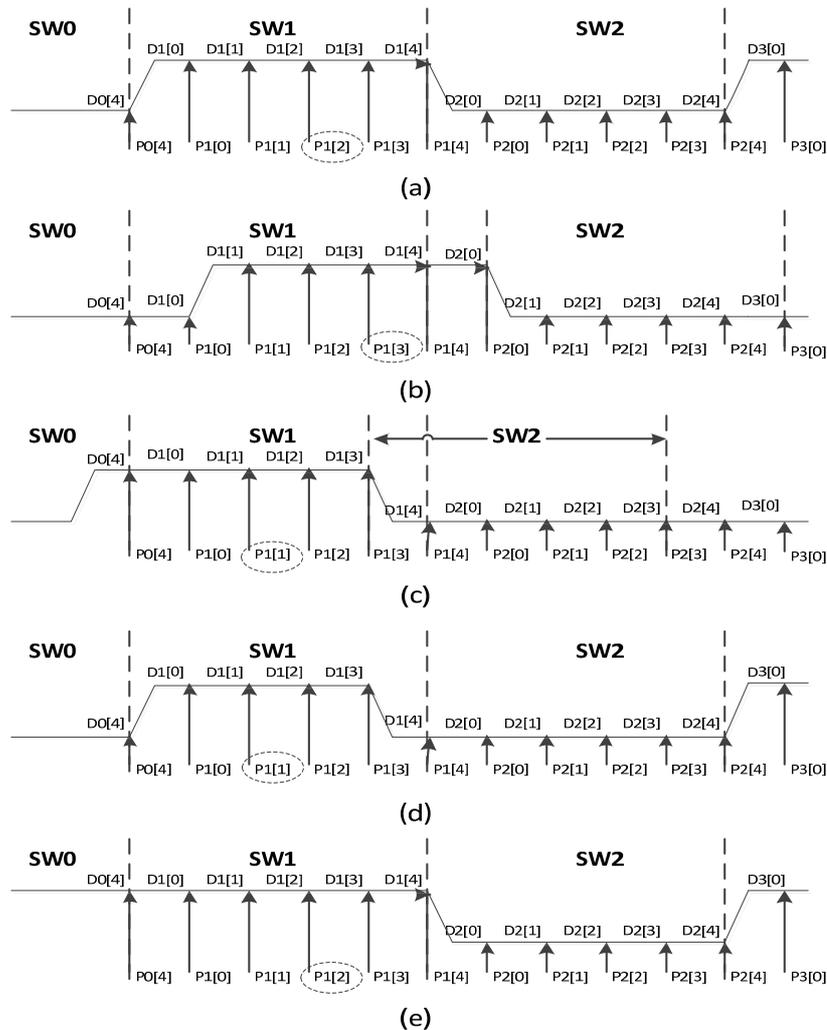


Figure 5. Sequence diagram of the data sampling

Figure 5 presents the sequence diagram of the data sampling selection. Supposing the current sampling window is SW1, if the jumping edge of the bus data falls between P0[4] and P1[0], the sampling result will be D1[2], which is used to determine the next sampling window, that is P1[4]~P2[4], shown in Figure 5(a). If the jumping edge of the bus data falls between P1[0] and P1[1], the sampling result will be D1[3] and the next sampling window will be P2[0]~P2[4], shown in Figure 5(b). If the jumping edge of the bus data falls between P1[3] and P1[4], the sampling result and the next sampling window will be D1[1] and P1[3]~P2[3], respectively, shown in Figure 5(c). If there are two jumps in the sampling window in P0[4]~P1[0] and P1[3]~P1[4], respectively, then the sampling result will be D1[1], determining the next sampling window P1[3]~P2[3], shown in Figure 5(d). If no jumping data in the sampling window, the sampling result will be D1[2], and the next sampling window is

P1[4]~P2[4], shown in Figure 5(e). However, if the sampling result is not in the list above, there may be a greater data jitter or a larger frequency shift between the transmitter and receiver, which is impossible to recover an accurate data. These different cases results are also summarized in Table 1.

Table 1. The sampling results in current sampling window P0[4] ~P1[4]

	Case1	Case2	Case3	Case4	Case5
Bus data jump	P0[4]~P1[0]	P1[0]~P1[1]	P1[3]~P1[4]	P0[4]~P1[0], P1[3]~P1[4]	Don't jump
Sampling results	D1[2]	D1[3]	D1[1]	D1[2]	D1[2]
The next sampling window	P1[4]~P2[4]	P2[0]~P3[0]	P1[3]~P2[3]	P1[3]~P2[3]	P1[4]~P2[4]

#### 4. Simulation and Analysis

In this paper, HSPICE is adopted to implement the circuit simulation, with a 0.18um technology library. The FIFO block is removed in order to get a higher simulation speed.

##### 4.1 The Generation of Test Vector

Firstly, a simulation is conducted for signal jitter as well as the frequency offset between transmitter and receiver. The sampling frequency is fixed at 480MHz and the frequency of the serial input data is  $480 \times (1 \pm 1/1000)$  MHz. A pseudo random data are generated using MATLAB to modulate the random jitter of the serial input signal. Based on this method, a serial input data with the length of 1,000,000,000-bit are generated as the test vector, which are used in simulation for the blind oversampled data recovery circuit.

##### 4.2 Simulation Results

Using frequency at 480MHz, the simulation results show that this circuit can be used to accurately recover the serial input data, even the signal with a frequency offset and a signal jitter. Some simulation results are presented in Figure 6. As can be seen, the input signals are recovery accurately, which are comparable with the out data. The simulation results also suggest that this circuit design can be applied in the receiver accordance with USB 2.0 protocol.

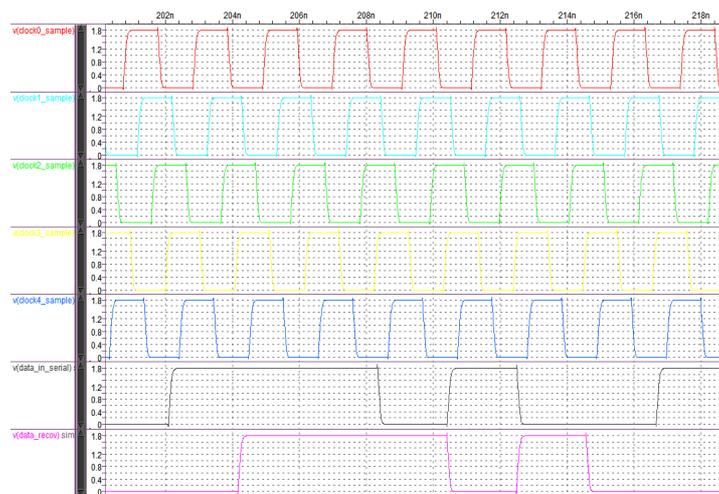


Figure 6. Some simulation results

#### 5. Conclusion

In this paper, we proposed a new blind oversampled data recovery circuit, including a 5X sampler, an edge detector, a phase selector, a MUX block, and a FIFO block. Through modifying the traditional phase selection algorithm, we used a dynamic sampling window instead of the previous fixed sampling window to sample the

input data, which can simplify the process of data recovery and also reduce the errors from data missing or data repeatedly sampling in the traditional recovery processing owing to the signal jitter and the frequency offset between transmitter and receiver in asynchronous communication.

We also conducted the circuit simulation with frequency of 480 MHz, with accurate data recovery results. So it suggests that this circuit can be used in USB 2.0 system with a less cost and a more simple design, which is more suitable for integration in SOC chips.

### References

- Compaq, I., & Intel, L. (2000). Universal serial bus specification revision 2.0 in USB Implementers\ Forum.
- Kim, J., & Jeong, D. K. (2003). Multi-gigabit-rate clock and data recovery based on blind oversampling. *Communications Magazine, IEEE*, 41, 68-74. <http://dx.doi.org/10.1109/MCOM.2003.1252801>
- Kyeongho, L., Sungjoon, K., Gijung, A., & Jeong, D. K. (1995). A CMOS serial link for fully duplexed data communication. *IEICE Transactions on Electronics*, 78, 601-612.
- Park, S. H., Choi, K. H., Shin, J. B., Sim, J. Y., & Park, H. J. (2008). A single-data-bit blind oversampling data-recovery circuit with an add-drop FIFO for USB2. 0 high-speed interface. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55, 156-160. <http://dx.doi.org/10.1109/TCSII.2007.911791>
- Yang, C. K. K., Farjad-Rad, R., & Horowitz, M. A. (1998). A 0.5- $\mu\text{m}$  CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling. *IEEE Journal of Solid-State Circuits*, 33, 713-722. <http://dx.doi.org/10.1109/4.668986>