**Cleveland State University**
**EngagedScholarship@CSU**

Electrical Engineering & Computer Science Faculty Publications

Electrical Engineering & Computer Science Department

5-2003

# Coding, Decoding, and Recovery of Clock Synchronization in Digital Multiplexing System

Hansheng Wang
*Tsinghua University*

Xiaoyi Qin
*Tsinghua University*

Lieguang Zeng
*Tsinghua University*

Fuqin Xiong
*Cleveland State University*, f.xiong@csuohio.edu

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

Part of the Systems and Communications Commons

**How does access to this work benefit you? Let us know!**

Original Citation

Repository Citation

# CODING, DECODING, AND RECOVERY OF CLOCK SYNCHRONIZATION IN DIGITAL MULTIPLEXING SYSTEM

**Hansheng Wang,** *Tsinghua University*

**Xiaoyi Qin,** *Tsinghua University*

**Lieguang Zeng,** *Tsinghua University*

**Fuqin Xiong,** *Cleveland State University*

**Abstract** **High-speed broadband digital communication networks rely on digital multiplexing technology where clock synchronization, including processing, transmission, and recovery of the clock, is the critical technique. This paper interprets the process of clock synchronization in multiplexing systems as quantizing and coding the information of clock synchronization, interprets clock justification as timing sigma-delta modulation ($T\Delta - \Sigma M$), and interprets the jitter of justification as quantization error. As a result, decreasing the quantization error is equivalent to decreasing the jitter of justification. Using this theory, the paper studies the existing jitter-reducing techniques in transmitters and receivers, presents some techniques that can decrease the quantization error (justification jitter) in digital multiplexing systems, and presents a new method of clock recovery.**

*Index Terms*—**Clock recovery, clock synchronization, digital multiplexing, jitter, $T\Delta - \Sigma M$.**

## I. INTRODUCTION

**I**N DIGITAL communication networks, in order to enlarge capacity and improve efficiency, several low-speed tributary signals are combined to form one high-speed composite signal for transmission in one broadband channel. Digital multiplexing is the technology of combining digital signals, and the device is called a digital multiplexing system. Digital multiplexing systems include plesiochronous digital hierarchy (PDH), synchronous digital hierarchy (SDH), and so on. Its performance greatly affects the performance of the entire digital communication network. In plesiochronous multiplexing, justification is needed because tributary clock and composite clock are not synchronous, and then jitter of justification is introduced. Jitter of justification is the most important and complicated theoretic problem in digital multiplexing, and there are two main jitter-analyzing methods: power spectrum method [1] and 1/p method [2], [3]. These two methods were all presented for PDH where there is only one stuff bit in one frame. While in

SDH, pointer adjustment introduced by frequency difference is essentially positive/zero/negative justification, whose stuff bits can be 8 (in TU-12) or 24 (in AU-4). Therefore, we first expand the two methods to a more general case where the number of stuff bits could be greater than one, and use the results as the reference for comparison in this paper.

Let $f_m$ be the average read clock frequency which is generally derived from the high-rate composite clock at the multiplexer, let $f_l$ be the actual frequency of the tributary clock, and the stuff ratio is defined as $\rho = (f_m - f_1)/SF_s$ and $|\rho| \le 1$ [1]. The sign of $\rho$ indicates whether the justification needed is positive (increase in $f_l$) or negative (decrease in $f_l$). $F_s$, the justification frame frequency, is the maximum frequency in which justification can take place. $S$ is the number of justification bits per justification frame ($S = 1$ in PDH, and $S = 1$, 8, or 24 in SDH). The period of justification frame is defined as $T_s = 1/F_s$. $F_s$ is decided by multiplexing hierarchy, and may not equal the frame frequency.

Similar to [1], jitter of justification for $S$ greater than one can be expressed as

$$\Phi_s(t) = \Lambda - S + \xi + S\rho t - S[\xi + \rho[t + \tau]] \qquad (1)$$

where $t$ is time variable normalized to $T_s$, $\Lambda$ is justification threshold, $\zeta$ is initial phase which is uniformly distributed on the interval $[0, S]$, the random variable $\tau$ is the time before $t = 0$ when a justification opportunity occurs. Function $[x]$ truncates the number $x$ to its integer part, for example, $[1.6] = 1$, $[-1.6] = -1$. Referring to [1], the Fourier transform of $\Phi_s(t)$ is the power spectrum

$$\Phi_s(f) = \sum_{n=1}^{\infty} \left( \frac{\rho S}{2n\pi} \right) \{\delta(f + n) + \delta(f - n)\}$$
$$+ \sum_{n=1}^{\infty} S \left( \frac{1}{2n\pi} \right)^2 \text{sinc}^2 \pi f$$
$$\times \{rep\delta(f - n\rho) + rep\delta(f + n\rho)\} \qquad (2)$$

where $repx(f) = \sum_{k=-\infty}^{\infty} x(f - k)$, and $f$ is frequency normalized to $F_s$. The first term of (2) only contains relatively high-frequency (multiple of $F_s$ since $f$ is normalized to $F_s$) components that are easy to filter, the second term contains relatively low-frequency components that are difficult to filter and are the main source of jitter.

For the 1/p analyzing method in [2] and [3], this paper expands it by introducing $S$. Referring to [2], when $\rho = q/p + \Delta\rho$ ($p$ is relatively prime to $q$), $\Phi_s(t)$ has the high-frequency jitter whose frequency is $F_s/p$, and the low-frequency envelope
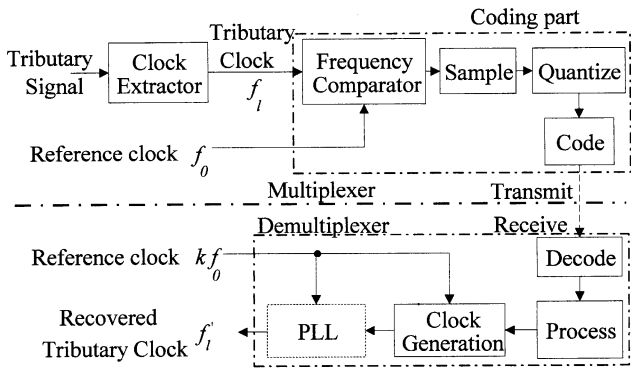
Fig. 1. Theoretic model of processing clock in digital multiplexing system.

whose period is nearly $1/(|\Delta\rho|pF_s)$ and whose amplitude is $S/p$. Ignoring the dc term, we have

$$\Phi_s(t) = \frac{q}{p}St + \Delta\rho St - S\left[\frac{q[t]}{p} + \Delta\rho[t]\right]. \qquad (3)$$

$t$ is also normalized to $T_s$, and is an integer at justification time. When $\Delta\rho$ is small enough that the low-frequency jitter cannot be filtered, the amplitude of remnant jitter is $A_{\mathrm{jpp}} = S/p$, and the frequency is $p|\Delta\rho|F_s$. This is the peak-to-peak jitter of justification.

The power-spectrum method can only estimate the root mean square (RMS) amplitude of the jitter, while peak-to-peak value is usually measured and is used as a quality indicator. The 1/p method estimates the peak-to-peak jitter value, but the result is not accurate. This paper proposes a theoretic model of processing clock synchronization in digital multiplexing, and the model interprets the process of clock synchronization from a point of view of coding and decoding the information of clock synchronization. In the multiplexer (transmitter), information from a tributary clock is sampled, quantized, and coded, and then it is transmitted to the receiver in the composite signal. In the demultiplexer (receiver), the clock information can be obtained from the composite signal, and be decoded, processed, and recovered, and then the tributary clock is obtained. It is clear that the major error source is the quantization process; thus, the essence of justification jitter is the quantization error. In fact, to reduce jitter of justification is to reduce the quantization error.

This paper is organized as follows. A theoretic model of processing clock synchronization in a complete digital multiplexing system is introduced in Section II. Coding of clock synchronization is considered in Section III. Decoding or recovery of clock synchronization is considered in Section IV.

## II. THEORETICAL MODEL OF PROCESSING CLOCK SYNCHRONIZATION IN DIGITAL MULTIPLEXING SYSTEM

In Fig. 1, we propose a theoretical model of clock processing in a digital multiplexing system, where the top half of the diagram is the multiplexer and the bottom half is the demultiplexer. This theoretical model is by no means the block diagram of a real-word clock synchronizer in a digital multiplexing system. We just use it to interpret the clock synchronization from a novel point of view.

In order to transmit information on the clock, coding is needed so that binary code can be generated for transmission. This process is similar to the process of normal signals, such as voice. The coding part is composed of a frequency comparator, sampler, quantizer, and encoder. In the comparator, a reference clock $f_0$ is compared with the tributary clock $f_l$. The reference clock can be synchronous with the composite clock, or can be independent. The frequency error is sampled, quantized, and coded. The binary code is inserted in the composite signal for transmission. The sample period and quantization step are different in different multiplexing hierarchies. For example, in SDH and PDH, the sample period is the minimum period in which justification is permitted, and the quantization step is related to the number of stuff bits.

In the decoding part of the demultiplexer, the binary code of clock synchronization from the multiplexer is decoded. Then clock synchronization can be processed further (such as low-pass filtering), and the frequency error between the tributary clock and the reference clock $f_0$ can be obtained. In the clock-generation block, a clock whose frequency is the same as the tributary clock is generated. The demultiplexing reference clock must be synchronous with the multiplexing reference clock, but its frequency may be different by a factor $k$, and $k$ is not necessarily an integer, but should be fixed and be able to be realized in the clock-generation block. Therefore, if the frequency error between the tributary clock and the reference clock $f_0$ is known, the clock-generation block can generate the tributary clock from $kf_0$. In general, $f_0$ is generated from the composite clock in transmitter, then the clock $kf_0$ can be easily obtained from the composite clock in receiver. In theory, if $f_0$ is independent in transmitter and $kf_0$ can be obtained in receiver, tributary clock can also be generated. If jitter of the clock from the clock-generation block can meet the requirement, this clock can be the recovered clock $f_l'$; otherwise, the phase-locked loop (PLL) is needed to filter out jitter further. In Section IV, a novel method that needs no PLL will be introduced.

In summary, in this theoretical model, a digital multiplexing system transmits a complete digital signal, including clock and data. While transmitting data, a digital multiplexing system also accomplishes coding and decoding tributary clock synchronization so that the tributary clock can be transmitted to the receiver correctly.

## III. CODING OF CLOCK SYNCHRONIZATION IN DIGITAL MULTIPLEXING SYSTEMS

### A. Principle of Timing Sigma-Delta Modulation $(\mathrm{T}\Delta - \Sigma\mathrm{M})$

In justification of clocks in PDH and SDH, phase comparison of two clocks is equivalent to integration of the frequency error between two clocks, and positive or negative justification makes phase error between the tributary clock and reference clock reduce or increase $S$ unit interval (UI). The process is equivalent to integrating the error between the sample value of $\rho$ and the code value of $\rho$, where $\rho = ((f_0 - f_1)/(SF_s))$. $\mathbf{f_0}$ is not limited by the definition in (1), and can be not only an average read clock $\mathrm{f_m}$. In fact, $\mathrm{f_0}$ can be any independent clock, if $|f_0 - f_1| \leq SF_s$. The code value being 1 means positive justification, being 0 means no justification, and being $-1$ means negative justifica-
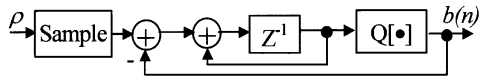
Fig. 2.   Diagram of $T\Delta - \Sigma M$.
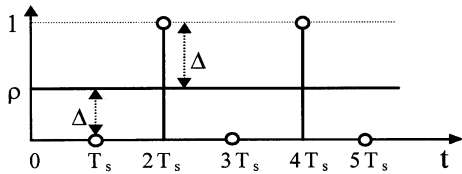


Fig. 3.   Principle of $T\Delta - \Sigma M$.

tion. So justification in digital multiplexing can be described as sigma-delta modulation in Fig. 2. In positive justification, the quantization function $Q[\cdot]$ is

$$Q[x] = \begin{cases} 1, & (x \leq 1) \\ 0, & (x < 1). \end{cases}$$

In positive/zero/negative justification, $Q[\cdot]$ is

$$Q[x] = \begin{cases} 1, & (x \geq 1) \\ 0, & (-1 < x < 1) \\ -1, & (-1 \geq x). \end{cases}$$

As shown in Fig. 2, coding clock synchronization is sigma-delta modulation $(\Delta - \Sigma M)$ of stuff ratio $\rho$ (which reflects the frequency error between the tributary clock and reference clock), we call this timing sigma-delta modulation $(T\Delta - \Sigma M)$. The waveform of $T\Delta - \Sigma M$ is shown in Fig. 3, where the small circles are code values and $\Delta$ is the error between the code value and $\rho$. This error is introduced by quantization. From (1) and the definition of quantization function $Q$, at $iT_s$, when $S\rho t - S[\rho[t]] \geq S$, the code value is 1; when $S\rho t - S[\rho[t]] \leq -S$, the code value is $-1$; when $S\rho t - S[\rho[t]]$ is smaller than $S$ and larger than $-S$, the code value is 0. So the code value of $T\Delta - \Sigma M$ is

$$b(i) = -[\rho i] - [\rho(i - 1)] . \tag{4}$$

$b(i)$ can be 1, 0, and $-1$. At $iT_s$, $\Delta$ is

$$\Delta(i) = \rho - [\rho i] + [\rho(i - 1)] . \tag{5}$$

$\Delta$ reflects frequency error between before and after quantizing, so jitter $\Phi_s(t)$ is

$$\Phi_s(t) = SF_s \int_0^1 \Delta(\tau) d\tau . \tag{6}$$

$SF_s\Delta(i)$ is frequency error between the quantized signal and the unquantized signal at time $t$ on the interval $[iT_s, (i+1)T_s]$. And on the interval, the phase error (jitter) is

$$\Phi_s(t) = \Phi_0(i) + SF_s\Delta(i)t, \quad iT_s \leq t < (i+1)T_s$$

where $\Phi_0(i)$ is the phase error at $iT_s$. That is

$$\Phi_0(i) = SF_s \sum_{k=0}^{i-1} \Delta(k)(i-1)T_s = S\rho(i-1) - S[\rho(i-1)] .$$

Normalizing time $t$ to $T_s$, we obtain

$$\Phi_s(t) = S\rho t - S[\rho[t]] \tag{7}$$

Equation (7) is essentially the same as (1). Therefore, $T\Delta - \Sigma M$'s quantization error introduces jitter of justification, and it also proves that $T\Delta - \Sigma M$ can describe justification in PDH and SDH. Referring to (2), with $f$ not being normalized to $F_s$, the power spectrum of $\Phi_s(t)$ in (7) is

$$\Phi_s(f) = \sum_{n=1}^{\infty} \left( \frac{f_0 - f_1}{2n\pi F_s} \right) \{\delta(f + nF_s) + \delta(f - nF_s)\}$$

$$+ \sum_{n=1}^{\infty} S \left( \frac{1}{2n\pi} \right)^2 \text{sinc}^2 \pi f \left\{ \text{rep}\delta \left( f - \frac{n(f_0 - f_1)}{S} \right) \right.$$

$$\left. + \text{rep}\delta \left( f + \frac{n(f_0 - f_1)}{S} \right) \right\} \tag{8}$$

where $\text{rep}x(f) = \sum_{k=-\infty}^{\infty} x(f - kF_s)$. Equation (8) is essentially the same as (2).

Now the $T\Delta - \Sigma M$ theory is used to interpret the clock synchronization in PDH and SDH. In one justification frame, there are several indication bits of justification to indicate positive, negative, or zero justification. In the demultiplexer, justification status can be judged through the indication bits. The indication bits are, in fact, the result of coding tributary clock synchronization. In previous applications, $T\Delta - \Sigma M$'s reference clock $f_0$ is obtained from the composite clock $f_h$, and $T\Delta - \Sigma M$'s code value is transmitted using the indication bits. The receiver can get the composite clock from composite signal, then the tributary clock can be recovered because the indication bits carry the result of $T\Delta - \Sigma M$. Thus, the reference clock can be obtained easily in the demultiplexer, and extra overhead to transmit code value of the reference clock is not needed. However, there still exists deficiency. Because $T\Delta - \Sigma M$ code value is transmitted using indication bits, the reference clock must be a composite clock. For example, in SDH signal relay, when a virtual container (VC) signal maps into another administration unit (AU) or tributary unit (TU), another $T\Delta - \Sigma M$ coding is needed because of the change of composite clock. But the coded clock has jitter, and jitter accumulation is introduced. For fear that accumulated jitter is too large, the number of relays are limited and then the scale and flexibility of the network is affected.

In theory, $T\Delta - \Sigma M$'s code value can be transmitted separately from transmitter to receiver. For example, $T\Delta - \Sigma M$'s reference clock can be any suitable clock, and the $T\Delta - \Sigma M$'s code value can be transmitted in separate overheads. If a clock in synchronization with the reference clock of the transmitter exists in the receiver, the demultiplexer can recover the tributary clock according to the code value. Then with relays, there is no jitter accumulation because multiple $T\Delta - \Sigma M$ operations are not needed. However, this requires a synchronized clock in the demultiplexer and overheads in the data frames.

### B. Methods of Reducing Quantization Error of Clock Synchronization Information in Transmitters

According to the discussion above, justification jitter in a digital multiplexing system is from a quantization error of clock
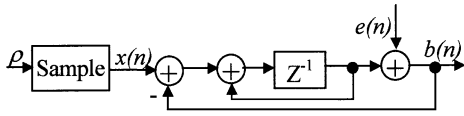
Fig. 4. Diagram of $\text{T}\Delta - \Sigma\text{M}$ with the quantization error.
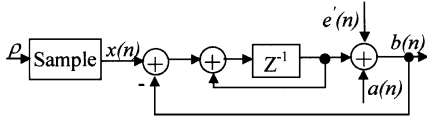


Fig. 5. $\text{T}\Delta - \Sigma\text{M}$ diagram of stuff threshold modulation.

synchronization information. In this section, based on this viewpoint, we discuss some methods of reducing justification jitter (quantization error of clock synchronization information).

Fig. 4 is the diagram of $\text{T}\Delta - \Sigma\text{M}$ with consideration of the quantization error. $x(n)$ is the sample sequence of $\rho$, and $e(n)$ is the sequence of quantization error. From Fig. 4, we can get the transfer function

$$B(z) = z^{-1}X(z) + \left(1 - z^{-1}\right)E(z). \tag{9}$$

The error $\Delta$ mentioned above is

$$\Delta(z) = \left(1 - z^{-1}\right)E(z) \tag{10}$$

$$\Delta(n) = e(n) - e(n-1). \tag{11}$$

In the demultiplexer, $b(n)$ is processed by a low-pass filter (LPF) to get the final result. Assume that the LPF is an ideal filter with cutoff frequency $f_c$, then without considering the dc part, the final power of remnant jitter is

$$\sigma_j^2 = \int\limits_{-f_c}^{f_c} \Phi_s(f)df = \int\limits_{-f_c}^{f_c} j2\pi f\Delta(f)df. \tag{12}$$

From (12), in general, the smaller the $f_c$ is, the smaller the remnant jitter is. For a fixed $f_c$, the less the frequency components that are lower than $f_c$ in $\Phi_s(f)$, the smaller the remnant jitter is. In transmitters, many jitter-reduction methods have been proposed, such as sawtooth stuff-threshold modulation [4], [5], random stuff-threshold modulation [6], model method [7], and stuff ratio-detachment method [8]. Various stuff-threshold modulation methods are compared in [9]. The essence of all these jitter-reduction methods is to change the frequency spectral structure of error $\Delta$: reducing the low-frequency part of the spectrum, and/or moving part of the low-frequency spectrum to higher frequencies that are easier to filter using an ordinary PLL. Fig. 5 is the $\text{T}\Delta - \Sigma\text{M}$ diagram of the (sawtooth/random) stuff-threshold modulation. The stuff-threshold modulation is equivalent to adding a modulation sequence $a(n)$ in $\text{T}\Delta - \Sigma\text{M}$, and $e'(n)$ is the new quantization error. Then (9) is changed into

$$B(z) = z^{-1}X(z) + (1 - z^{-1})\left(E'(z) + A(z)\right). \tag{13}$$

The factor affecting the remnant jitter is $a(n) + e'(n)$, where $a(n)$ corresponds to high-frequency jitter and is easy to filter. Thus, only $e'(n)$ affects the final remnant jitter. For sawtooth stuff-threshold modulation, $a(t) = at - [a[t]]$, where $a$ is the slope ratio and $t$ is normalized to $T_s$.
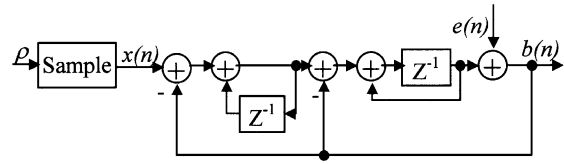


Fig. 6. Principle diagram of two-stage $\text{T}\Delta - \Sigma\text{M}$.
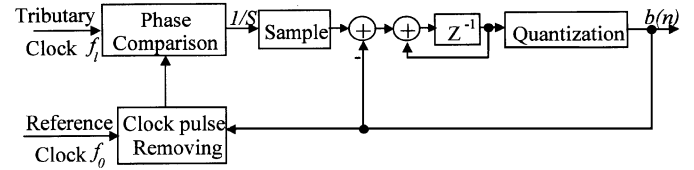


Fig. 7. Realization diagram of two-stage $\text{T}\Delta - \Sigma\text{M}$.
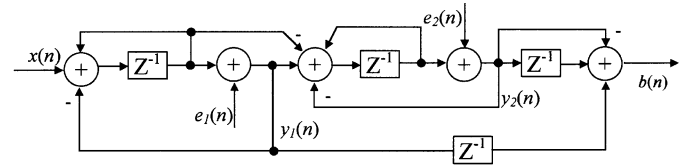


Fig. 8. Diagram of alternative MASH $\Sigma\Delta$ modulator.

In speech coding, sigma–delta modulation $(\Delta - \Sigma\text{M})$ is an important coding method, and has been researched thoroughly. Some methods of changing $\Delta - \Sigma\text{M}$'s circuit structure to reduce $\Delta - \Sigma\text{M}$'s quantization error, such as multistage $\Delta - \Sigma\text{M}$, have been presented [10], [11]. From the above discussion, the coding of clock synchronization information in digital multiplexing systems is essentially the same as $\Delta - \Sigma\text{M}$. Methods of quantization error reduction in $\Delta - \Sigma\text{M}$ are also applicable for reducing quantization error (justification jitter) in $\text{T}\Delta - \Sigma\text{M}$. Thus, we can have multistage $\text{T}\Delta - \Sigma\text{M}$.

Figs. 6 and 7 are, respectively, the principle diagram and the actual realization diagram of the two-stage $\text{T}\Delta - \Sigma\text{M}$. Actually, the method of adaptive threshold modulation in [12] is identical to two-stage $\text{T}\Delta - \Sigma\text{M}$. From Fig. 6, the transfer function of the two-stage $\text{T}\Delta - \Sigma\text{M}$ is

$$B(z) = z^{-1}X(z) + (1 - z^{-1})^2 E(z) \tag{14}$$

and then

$$\Delta(z) = (1 - z^{-1})^2 E(z). \tag{15}$$

Compared to the $\text{T}\Delta - \Sigma\text{M}$, the two-stage $\text{T}\Delta - \Sigma\text{M}$ effectively adds a high-pass finite impulse response (FIR) filter to the quantization error, reducing low-frequency spectral components and eventually the remnant jitter. The difference in $\text{T}\Delta - \Sigma\text{M}$'s number of stages is equivalent to the difference in the high-pass filter's order. The higher the number of stages, the higher the high-pass filter's order, and the better the effects of filtering and jitter reduction. Fig. 8 is the diagram of the multistage $\text{T}\Delta - \Sigma\text{M}$ applied in digital multiplexing, which is called the alternative multistage noise shaping (MASH) $\Sigma\Delta$ modulator in [13]. Its number of stages is higher than two and its jitter performance is better than two-stage's. It is noted that when the number of stages is more than two, the circuits are not always stable and are stable only under certain conditions [11], [13], [14].
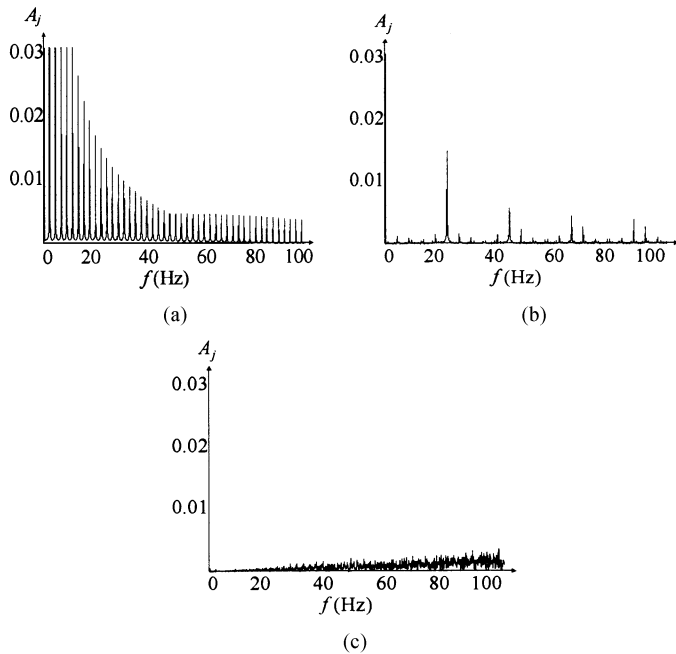
Fig. 9. Jitter frequency spectra.



Fig. 10. Novel recovery method without PLL.

Using (5), (6), (8), and (15), jitter frequency spectra of several methods with $\rho = 0.0011$ and $F_s = 2$ kHz are plotted in Fig. 9. As it is the low-frequency jitter that mostly affects the final remnant jitter, Fig. 9 only gives out the spectra below 100 Hz. Fig. 9(a) is the frequency spectrum of $T\Delta - \Sigma M$ without any process of reducing quantization error. Fig. 9(b) is the frequency spectrum of model method or sawtooth stuff-threshold modulation method, and $\rho_1$ of the model method or $a$ of the sawtooth stuff-threshold modulation is 0.1. Fig. 9(c) is the spectrum of two-stage $T\Delta - \Sigma M$. According to Fig. 9, due to the use of quantization error-reduction methods, the low-frequency spectra of (b) and (c) are much less than (a). Thus, these methods can reduce the remnant jitter.

## IV. CLOCK RECOVERY IN DIGITAL MULTIPLEXING SYSTEM

Conventionally, there is no further processing after the decoding of clock synchronization, the clock produced by the clock-generation block does not satisfy the requirement, and a PLL is needed for filtering jitter. In SDH, the low-frequency large-amplitude jitter induced by pointer justifications is difficult to filter using an ordinary PLL. Hence, several processes need to be added, or the PLL needs a special design for very narrow bandwidth and relatively wide capture and synchronizing range.

As shown in Fig. 1, the procedure of decoding $T\Delta - \Sigma M$ is to obtain $T\Delta - \Sigma M$'s code value $b(i)$ in each $T_s$ according to indication bits, then use $b(i)$ to process reference clock $f_0$ to get clock $f'_l$ on the interval $t \in [iT_s, (i+1)T_s]$. When there is positive or negative justification in the justification frame, $S$ pulses of the reference clock $f_0$ will be reduced or increased. So the frequency of $f'_l$ in $[iT_s, (i+1)T_s]$ is

$$f_l(i) = f_0 - SF_s b(i). \tag{16}$$

$f'_l$'s jitter is $S$ UI. If time $t$ is normalized to $T_s$, the time-domain equation of the clock's jitter is (7). Known from (16), the result
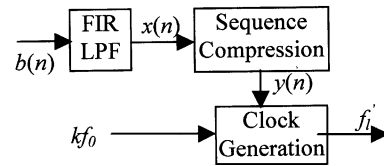
of decoding is used to generate the clock without any further process. Therefore, the jitter of the generated clock is still large.

Pointer justification in SDH introduces 8-bit or 24-bit large jitter, and the simple recovery method mentioned above cannot meet requirements. Some methods such as the bit-leak method [15], self-adaptive bit-leak method [15], and random phase modulation method [16], [17] were proposed. In these methods, $b(i)$ are all processed after decoding; and in the clock-generation block, according to the results of processing, $f'_l$ with smaller jitter is generated.

The methods mentioned above all need a PLL to filter the remnant jitter further, as their processes of the clock synchronization only decrease jitter to some extent, and the jitter is still too big to satisfy the requirement. In the following, we will present a novel recovery method that needs no PLL. The key element of this method is a FIR LPF with very narrow bandwidth that can directly filter the jitter to satisfy the requirement. Fig. 10 is the diagram of this novel method. The transfer function of the FIR LPF is as follows:

$$H(z) = \frac{X(z)}{B(z)} = \frac{1}{m} \sum_{i=1}^{m} z^{-i} \tag{17}$$

where $m$ is the length of the FIR LPF, and $x(n)$ is the output of the FIR LPF, $X(z)$ and $B(z)$ are $x(n)$'s and $b(n)$'s $Z$ transform, respectively. In the sequence compression block, $x(n)$ are sampled to form $y(i)$, that is, $y(i) = x(mi)$. The sampling period is $mT_s$ and thus, the period of the bits of $y(i)$ is $mT_s$. With sequence compression, the FIR LPF is simple and just a block of accumulating $b(n)$ every $mT_s$. The clock generation block generates tributary clock $f'_l$ according to $y(i)$, and $f'_l$ keeps constant in the time range of $[imT_s, (i+1)mT_s]$

$$f_l{}'(i) = f_0 - SF_s y(i). \tag{18}$$

Fig. 11 is the diagram of one kind of system implementation. Suppose $k$ is 16. With the period of $mT_s$, a $+/-S$ counter operates according to $b(n)$. In a justification frame period $T_s$, when $b(n) = 1$, the $+/-S$ counter increases by $S$; when $b(n) = -1$, the $+/-S$ counter decreases by $S$; when $b(n) = 0$, the $+/-S$ counter holds its result. Every $mT_s$, the counting result is sent to a register, and then the $+/-S$ counter is cleared to begin the next period's counting. Therefore, the $+/-S$ counter's result in the $i$th period of $mT_s$ is $mSy(i)$. And $y(i) = (1/m)\sum_{n=1}^{m} b((i-1)m + n)$. According to (4), we can get

$$y(i) = \frac{([m\rho i] - [m\rho(i-1)])}{m}. \tag{19}$$

The adder also adds a constant $16mS$ to $y(i)$. Hence, within a period of $mT_s$, $16mS(1 + y(i))$ positive edges of Register
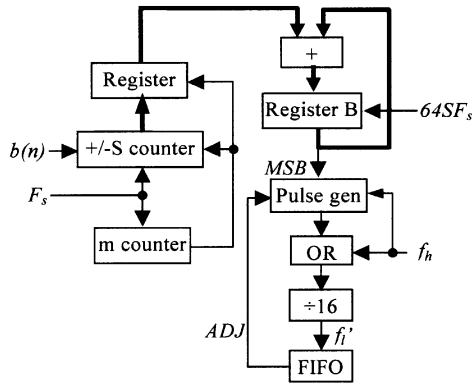
Fig. 11.  Implementation diagram of timing recovery.



Fig. 12.  Frequency spectrum of the undulation.

B's most significant bit (MSB) are evenly distributed. The pulse generator (labeled as "Pulse gen" in the figure) generates pulses whose width is one period of the clock $f_h$. $f_h$ is a high-speed clock gotten from composite clock, whose frequency is $16(f_0 + SF_s)$, here, $f_0$ is the tributary clock frequency when there is not any justification. The OR logic can make $f_h$ deduct $16mS(1 + y(i))$ cycles within $mT_s$ so that the frequency of its output clock is $16(f_0 - Sy(i)F_s)$. Then, after being divided by 16, the recovered tributary clock can be obtained. The first-in, first-out (FIFO) can give adjustment (ADJ) signal to Pulse gen. When FIFO is full or empty, ADJ can make Pulse gen reduce or add one deducted cycle. In this method, ADJ can only affect Pulse gen at the beginning because the error between $f_l'$ and $f_l$ is less than one cycle within $mT_s$, and the long-time average value of $f_l'$ equals that of $f_l$.

Jitter of the recovered tributary clock $f_l'$ is made up of high-frequency jitter and low-frequency undulation. The pulse-removing operation generates a high-frequency sawtooth waveform, which is approximately $kSy(i)/T_s$ Hz in frequency, and $1/k$ UI in amplitude. As the existence of $T\Delta - \Sigma M$ quantization error, there exists an error between $y(i)$ and $\rho$. So there is also an error between $f_l'$ and $f_l$ : $\Delta f_l(i) = f_l'(i) - f_l = SF_s(\rho - y(i))$. This error leads to a low-frequency undulation in addition to the high-frequency jitter. From (19), at the time $imT_s$, the amplitude of the undulation is

$$A_1(i) = \sum_{k=1}^{i-1} \Delta f(k)mT_s = (i-1)Sm\rho - S\left[(i-1)m\rho\right].$$

$$(20)$$

Set $A_2(i) = A_1(i+1) - A_1(i)$, then

$$A_2(i) = Sm\rho - S\left([im\rho] - [(i-1)m\rho]\right).$$

$$(21)$$

Therefore, the time-domain waveform of this undulation is

$$\Phi_{\text{low}}(t) = A_1(i) + \frac{A_2(i)}{mT_s}\left(t - (i-1)mT_s\right)$$

$$(22)$$

where $(i-1)mT_s \le t < imT_s$. The frequency spectrum of the undulation is the Fourier transform of $\Phi_{\text{low}}(t)$.

Fig. 12 is the frequency spectrum curves of the undulation with different $m$ on the condition of $S = 8$, $\rho = 1/896$, and $T_s = 1/(8)$ kHz. From these curves, we can see that the component of undulation whose frequency is higher than 10 Hz is quite small and can be ignored. And the component whose frequency is lower than 10 Hz is wander and is much less than valid value
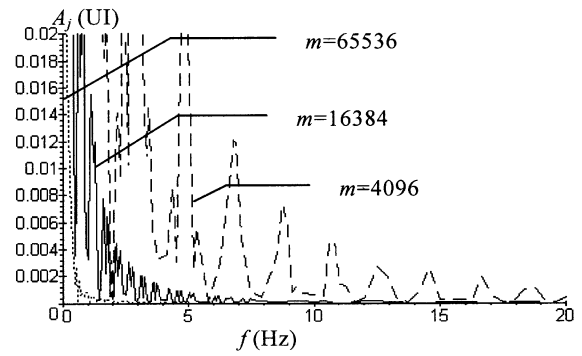
(because wander can affect network only with larger value, the lower its frequency is, the larger its valid value is). Therefore, the undulation can be ignored. In addition, with increase of $m$, the effect of the undulation decreases.

## V. CONCLUSION

This paper interprets the processing of clock synchronization as quantizing and coding the clock, interprets justification as $T\Delta - \Sigma M$, and interprets the jitter of justification as quantization error. According to this theoretic model, decreasing the jitter caused by justification is equivalent to decreasing the quantization error. From the point of view of this theory, the paper examines some former jitter-decreasing methods, and proposes to use quantization error-reduction methods of other fields in digital multiplexing systems. It is shown that these methods can reduce the remnant jitter. In addition, in the demultiplexer, clock synchronization recovery is the procedure of decoding $T\Delta - \Sigma M$. Its main parts are decoding, processing, clock generation, and PLL. Based on the theory, a novel clock synchronization recovery method without the PLL is presented. This method can efficiently filter the low-frequency large-amplitude jitter induced by pointer justifications, and the circuit realization is simpler without the PLL. The novel clock-recovery method depicted in Fig. 10 has been successfully applied to PDH and SDH application-specific integrated circuits designed by Tsinghua University, Beijing, China.

## REFERENCES

[1] D. L. Duttweiler, "Waiting time jitter," *Bell Syst. Tech. J.*, vol. 51, no. 1, pp. 165–207, 1972.
[2] Y. Sun, *Digital Multiplexing Technique*, 2nd ed.  Beijing, China: Post and Telecommunication Press, 1991.
[3] Y. Matsuura, S. Kozuka, and K. Yuki, "Jitter characteristics of pulse stuff synchronization," in *Proc. IEEE Int. Conf. Communications*, 1968, pp. 259–264.
[4] R. G. Kusyk, T. E. Moore, and W. A. Krzymien, "Spectral analysis of waiting time jitter in the presence of stuff threshold modulation," *Electron. Lett.*, vol. 26, no. 8, pp. 526–528, 1990.
[5] G. L. Pierobon and R. P. Valussi, "Jitter analysis of a double-modulated threshold pulse stuff synchronizer," *IEEE Trans. Commun.*, vol. 39, pp. 594–602, Apr. 1991.
[6] B. Wang, "Waiting time jitter reduction by the optimum random stuff threshold modulation," in *Proc. Int. Conf. Communication Technology*, Beijing, China, 1996, pp. 24.03.1–24.03.4.
[7] L. Zeng and C. Feng, "A novel justification method—model method for positive/zero/negative justification," *J. China Inst. Commun.*, vol. 7, no. 3, pp. 1–8, 1986.

[8] ——, "A novel method of reducing waiting time jitter in positive justi-fication," *J. China Inst. Commun.*, vol. 13, no. 1, pp. 45–51, 1992.

[9] S. S. Abeysekera, A. Cantoni, and V. Sreeram, "A comprehensive analysis of stuff threshold modulation used in clock-rate adaptation schemes," *IEEE Trans. Commun.*, vol. 46, pp. 1088–1096, Aug. 1998.

[10] J. C. Candy, "A use of double integration in sigma-delta modulation," *IEEE Trans. Commun.*, vol. COM-33, pp. 249–258, Mar. 1985.

[11] P. M. Aziz, H. V. Sorensen, and J. V. Spiegel, "An overview of sigma-delta converters," *IEEE Signal Processing Mag.*, vol. 13, no. 1, pp. 61–84, 1996.

[12] M. J. Klein and R. Urbansky, "Network synchronization—a challenge for SDH/SONET?," *IEEE Commun. Mag.*, vol. 31, no. 9, pp. 42–50, 1993.

[13] S. S. Abeysekera, "Sigma-delta architectures for pulse-stuff synchro-nizers," *Int. J. Circuit Theory Applicat.*, vol. 25, no. 2, pp. 457–467, 1997.

[14] ——, "Analysis of adaptive threshold modulation used in pulse stuffing and pointer adjustment schemes," *IEEE Trans. Commun.*, vol. 49, pp. 1835–1844, Oct. 2001.

[15] R. G. Kusyk, W. A. Krzymien, and T. E. Moore, "Analysis of techniques for the reduction of jitter caused by SONET pointer adjustments," *IEEE Trans. Commun.*, vol. 42, pp. 2036–2050, Feb.-Apr. 1994.

[16] H. Sari and G. Karam, "Cancellation of pointer adjustment jitter in SDH networks," *IEEE Trans. Commun.*, vol. 42, pp. 3200–3207, Dec. 1994.

[17] T. Wang, "Efficient methods to reduce jittering induction in pointer ad-justment in SDH networks," *Digital Commun.*, pp. 37–40, 1996.

**Lieguang Zeng** received the B. Sc. (Eng.) degree from Tsinghua University, Beijing, China, in 1970.

Since 1970, he has been with the Department of Electronic Engineering, Tsinghua University, Beijing, China, where he is currently a Professor. From 1974 to 1998, he researched speech coding, and from 1979 to 1990, he researched PDH networks and PDH ASIC design. From 1991 to 2001, he was the Group Leader who was responsible for SDH network research and SDH ASIC design. Currently, his research efforts are concentrated on 10 Gb/s MAN and its ASIC design. His research interests include telecommunications, high-speed networks, and ASIC design.

**Hansheng Wang** received the B. Sc. (Eng.) degree from Tsinghua University, Beijing, China, in 1994, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 1998.

From 1998 to 2000 he was with the State Key Lab on Microwave and Digital Communication of Tsinghua University as a lecturer. Since 2001, he has been with Palm, Inc., Milpitas, CA. His research interests include telecommunications, SDH and ATM networks, and ASIC design.

**Xiaoyi Qin** received the B. Sc. (Eng.) degree from Tsinghua University, Beijing, China, in 1996, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2001.

Since 2001, she has been with Palm, Inc., Milpitas, CA. Her research interests include telecommunica-tions, SDH and Ethernet networks, access networks, and ASIC design.

**Fuqin Xiong** (S'87–M'88–SM'96) received the B.Sc. degree in communication engineering in 1970 and the M.Sc. degree in communication and elec-tronic system engineering in 1982 from Tsinghua University, Beijing, China, and the Ph.D. degree in 1989 from the Department of Electrical Engineering, University of Manitoba, Winnipeg, MB, Canada.

From 1970 to 1978 he was an Assistant Lec-turer, and from 1983 to 1984 a Lecturer in the Department of Radio-Electronics Engineering, Tsinghua University, Beijing, China. From 1984 to 1985 he was a Visiting Scholar at the Department of Electrical Engineering, University of Manitoba. He joined the Department of Electrical Engineering, Cleveland State University, Cleveland, OH, in 1990 as an Assistant Professor. He was promoted to Associate Professor in 1995, and to Professor in 2002. In 1997, he was on sabbatical leave for one semester at the City University of Hong Kong and the summer in Tsinghua University, Beijing, China. He was a NASA-ASEE Summer Faculty Fellow in 2000, 2001, and 2002 at NASA Glenn Research Center, Cleveland, Ohio. His research interests are in communication engineering, particularly in modulation and coding techniques for digital communications, including satellite communications and mobile communications. He has been directing several research projects sponsored by NASA Glenn Research Center, Cleveland, Ohio. He is the author of the book *Digital Modulation Techniques* (Boston, MA: Artech House, 2000) and a contributor to *The Wiley Encyclopedia of Telecommunications*, 2002.

Dr. Xiong is a regular reviewer for IEEE and IEE.