TH10: IT-business & Computer Systems

TH11: Bio Technologies

TH12: Video/Audio/Speech Processing

PS1: Poster Session I

Proceedings of ICEIC2008, June 24-27, 2008, Tashkent, Uzbekistan

Precise Network Synchronization Technique Using Phase Adjustment and External Filtering

[*]Chang-Kyung Seong, [**]Seung-Woo LEE, and [*]Woo-Young Choi
[*]Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
[**]NIP Technology team, Electronic Telecommunication Research Institute, Daejoen, Korea
wchoi@yonsei.ac.kr

Abstract – ***Network synchronization is extremely important for many applications. In order to improve the performance of Pulse Addition and Swallowing (PAS) method, which considers only TOD (Time-Of-Day) adjustment, we investigate clock rate synchronization. To achieve the extension of frequency resolution and deterministic control gain in clock rate control, the clock phase is directly adjusted. Also, digital LPF is used to mitigate fluctuation of clock rates due to granularity of TOD. The proposed structure is modeled and simulated with MATLAB. Simulation results for cascaded ten nodes show that the peak-to-peak time offset of 3UI is reduced to about 0.21UI by using LPF.***

1. Introduction

Time and frequency synchronization between distributed network nodes has many important benefits by making network elements share the same time information and clock frequency. Applications such as real-time AV streaming, automatic control or manufacturing, wireless communication, and positioning systems especially require precise synchronization due to hard timing constraints.

Several approaches have been considered for network synchronization. Most of them are based on two-way message exchange using the time stamp specified in IEEE 1588 [1]. In order to adjust time rate, PAS method has been most widely used in synchronization systems [2]. In addition, DCO (Digitally-Controlled Oscillators) or DCXO (Digitally-Controlled Crystal Oscillators) have been used to generate variable clock frequency in ADPLL (All-Digital Phase Locked Loops) applications [3][4]. The PAS method, however, does not adjust the frequency of clock signal but only the increment rate of TOD. Although use of DCO or DCXO allows adjustment of clock frequency, the resulting frequency resolution is not sufficient and complex hardware is required to overcome this. It is also a problem that their characteristics are not deterministic due to PVT variations.

Granularity of TOD also degrades the performance of synchronization system. With inherently quantized TOD, the time offset between two nodes fluctuates even in synchronization due to the uncertainty of time information.

We propose a new structure to reduce the fluctuation of time offset caused by finite frequency resolution and granularity of TOD. Section II and III describe time synchronization model and problems causing peak-to-peak fluctuation of time offset, respectively. Section IV proposes two ideas to overcome these problems and section V wraps up the paper with conclusion.

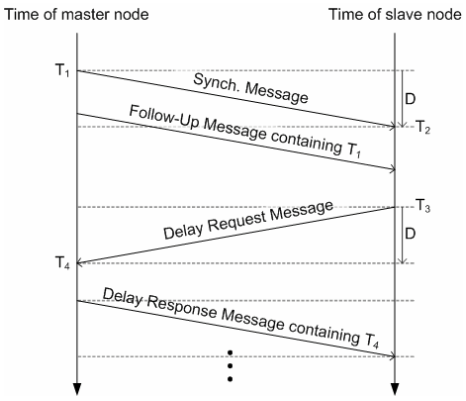


Figure 1: One set of message exchange between master and slave nodes to measure time offset and delay

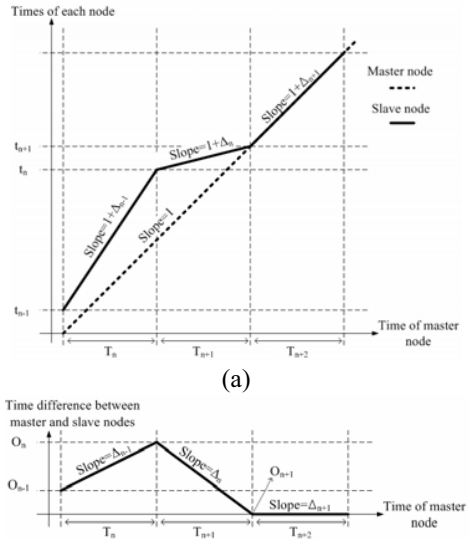
2. Time Synchronization Model

Figure 1 shows the two-way message exchange procedure between master and slave nodes described in IEEE 1588 PTP (Precision Time Protocol) [5]. This diagram has two independent time axes, master and slave nodes. Each axis corresponds to TOD value based on its own free-running oscillator. In order to synchronize slave's own clock frequency and TOD value to those of master node, slave node measures time offset from master and network delay.

One set of procedure consists of four messages. At the start of a set, master node sends *Synch Message* to slave node. When the message arrives at slave after propagation and delay (D), the slave node marks and stores the arrival time T_2 referring to slave's TOD value. After a while, *Follow-up Message* containing the departure time of the *Synch Message* (T_1) is sent to slave node. In order to get delay time (D) and separate it from time offset (O), slave node sends *Delay Request Message* to master node at T_3, again. When master node detects this message, master node stores the arrival time T_4 referring to master's TOD value. By returning T_4 back with *Delay Response Message* from master node, slave node is able to calculate O and D as shown in equation (1) and (2), separately. Note that the equations, however, are held for symmetric delays in both directions.

$$O = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad \text{- (1)}$$

$$D = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \quad \text{- (2)}$$



(a)

(b)

Figure 2: Simplified synchronization model with clock rate adjustment: (a) times of each node vs. time of master node (b) time offset between slave and master nodes vs. time of master node.

Figure 2 (a) and (b) illustrate simplified synchronization models. Here, only two nodes are supposed for simplicity. In figure 2 (a), horizontal and vertical axes represent the time of master node and the time of each node, respectively. Note that the slope means the normalized clock rate of each node. Dashed line is the time of master node with unity slope. On the other hand, segmented solid lines are the time of slave node. Slave node adjusts its own clock rate after every message exchanges. Without perfect synchronization, the n-th segment of solid line has slightly different slope from that of master node due to frequency offset, Δ_n. Figure 2 (b) is a modified version of figure 2 (a). Now vertical axis represents time offset between slave and master node. By denoting the time offset at the n-th message exchange as O_n, simple equation is derived as follows

$$\frac{O_{n+1}-O_n}{T_n}=\Delta_n \quad \text{- (3)}$$

The frequency offset, Δ_n, that the synchronizer should have after the n-th message exchange is derived from equation (3) and its recursive version as follows

$$\Delta_n=\Delta_{n-1}+\frac{O_{n+1}-O_n}{T_n}-\frac{O_n-O_{n-1}}{T_{n-1}} \quad \text{- (4)}$$

Finally, an equation (5) is derived by forcing the next time offset, O_{n+1}, to be zero,

$$\Delta_n=\Delta_{n-1}-\frac{O_n}{T_n}-\frac{O_n-O_{n-1}}{T_{n-1}} \quad \text{- (5)}$$

3. Problems in Synchronization Systems

A. Finite frequency resolution

In digitally-controlled clock synchronization systems, the amount of controllable frequency step is limited. Quantization errors due to the finite frequency step continuously accumulate time errors during a message exchange interval. The amount of error accumulation is proportional to the message exchange interval. As network systems demand longer message exchange intervals, more accurate and precise frequency control is necessary.

The minimum frequency adjustment step should be able to allow fine tuning, so that the time error accumulation during an exchange interval is under 0.5 UI [2], or

$$T\cdot\Delta f<0.5 \quad \text{- (6)},$$

where T is the normalized message exchange interval and Δf is the frequency resolution. In IEEE 1588, nominal values of clock rate and message exchange interval are 25MHz and 2 seconds, respectively. As a result, T is 5×10^7 UI and Δf must be less than 0.01ppm. IEEE 1588 allows the free-running frequency offset up to ±100ppm. Therefore, all nodes should be able to change their frequency rate up to ±200ppm in the worst case. In order to adjust frequency from 0.01ppm to 200ppm, slave nodes must the control frequency rate with at least 15-bit digital code [2].

Several approaches have been used for rate adjustment. Among these, the PAS method is the most straightforward, and has been used in most IEEE 1588 implementations [2]. This method adjusts the rate of TOD by increasing or decreasing the increment of TOD, depending on whether the clock rate should get faster or slower. If faster clock rate is needed, larger increment of TOD is occasionally selected, and vice versa. Despite the ease of implementation, the accuracy of synchronization is limited by the resolution of increment TOD. And PAS does not provide the adjustment of the clock rate.

As tools to digitally control the oscillation frequency, DCOs and DCXOs have been widely used in ADPLLs. Since it is hard to make the resolution of DCO or DCXO more than 15-bit, ΔΣ modulators are used to increase effective frequency resolution [3][4]. However, DCOs integrated in a silicon die suffer from fluctuation of free-running frequency, poor phase noise, and PVT variation, yielding non-deterministic DCO gain and oscillation range. Although DCXOs have more stable oscillation frequency and better phase noises compared to DCO, they are expensive to implement and also have non-deterministic gain.

B. Granularity of TOD

Time information, or TOD, is generated by a digital counter triggered by its own system clock signal. Without using PAS, the minimum step of TOD value is fixed to the amount of 1 UI of system clock signal. Since real (or analog) departure and arrival times (T1, T2, T3 and T4) are quantized to integer multiple of UI, they can be decomposed into two parts: the TOD $[T_k]$ and quantization error e_k,

$$[T_k]=T_k+e_k, \quad (k=1,2,3,4) \quad \text{- (7)}$$

where [X] rounds X to the nearest integer toward $+\infty$, and $0\le e_k<1$ in UI. Calculated time offset by message exchange (O') is also quantized from real time offset (O) as shown in equation (8);

$$O=\frac{([T_2]-[T_1])-([T_4]-[T_3])}{2}-\frac{(e_2-e_1)-(e_4-e_3)}{2} \quad \text{- (8)}$$
$$=O'-E$$

where $O'=\frac{([T_2]-[T_1])-([T_4]-[T_3])}{2}$, and $E=\frac{(e_2-e_1)-(e_4-e_3)}{2}$.

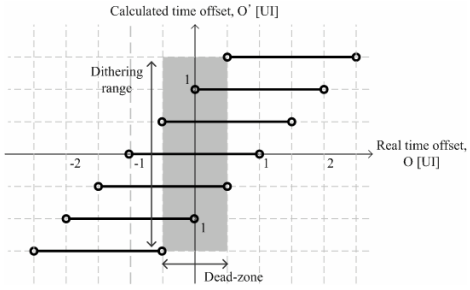


Figure 3: Effect of finite time resolution

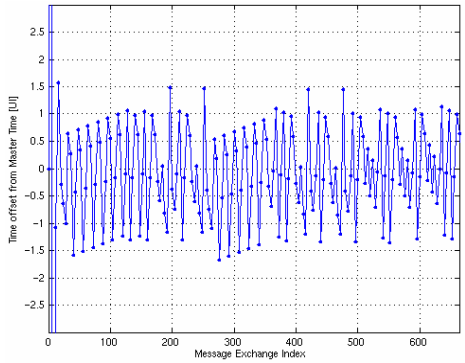


Figure 4: Behavioral simulation of time synchronization with TOD

Figure 3 plots the effect of equation 8. It is observed that calculated time offset (O') will have one among three or four values even for one real time offset value (O), depending on the condition of four variables, e_1, e_2, e_3 and e_4. This uncertainty due to granularity of TOD degrades synchronization performance. Since the calculated time offset (O') under ±0.5UI is not detectable, this range can be considered as dead-zone for detection. Therefore, resulted real time offset (O) can fluctuate up to ±1.5UI.

In order to show the effect, a behavioral model of synchronization system was realized using MATLAB and simulations were performed with granulated TOD instead of real analog time. Following assumptions are made ain simulations:

· Slave node has different initial TOD value and free-running frequency with master node;

· Message exchanges are executed every 2×10^8 clock cycles. (2 sec with 100-MHz clock);

· Up- and down-link Network delays are symmetric;

· The granularity of TOD is 1 UI.

As expected, figure 4 shows that time offset fluctuates around zero with peak-to-peak variation of ±1.5UI.

4. A New Structure

A. Phase adjustment

The mechanism of a new approach with direct control of phase is depicted in figure 5. Straight solid line represents slave's time offset from master with free-running frequency offset. Segmented solid lines are the result of phase shifts to adjust the frequency of slave clock. Suppose that clock signal can be shifted by 1/N UI in phase, or minimum phase shift

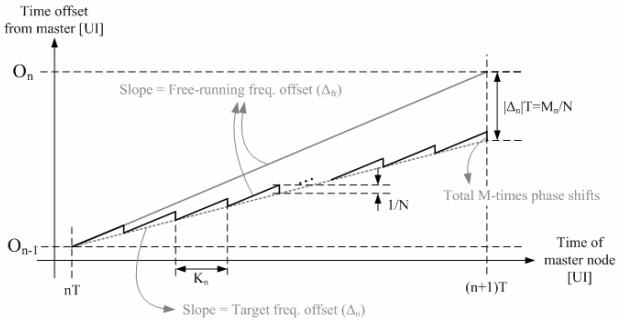


Figure 5: Generation of effective frequency offset using phase adjustment

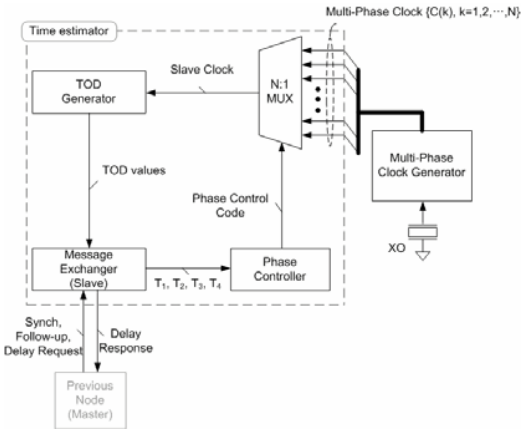


Figure 6: Block diagram of clock synchronization system using phase adjustment

step is 1/N UI. In the figure, there are total M-times shifts of unit phase step at every K_n-th clocks to generate frequency deviation as much as target frequency offset Δ_n from the free-running line. The accumulated time offset during T is the same as the amount of total phase shift as shown in equation 9,

$$|\Delta_n|T = \frac{M_n}{N} \cdot \text{ - (9)}$$

And, the period of phase shift, K_n, is derived as follows.

$$K_n = \frac{T}{M_n} = \frac{1}{|\Delta_n| \cdot N} \text{ - (10)}$$

By handling the phase of clock signal instead of frequency, effective frequency resolution becomes almost infinite. The minimum effective frequency offset is generated by only one step of phase shift during one message exchange interval. Moreover, control gain is deterministic and immune to PVT variation unlike oscillators mentioned above. Although non-uniform phase difference of N clocks can cause unwanted phase error, the total sum of phase error is zero. The drawback of this approach is that the phase of slave clock periodically jumps as much as 1/N UI and the peak-to-peak jitter of clock is 1/N UI. However, this effect becomes negligible by choosing large N.

Figure 6 shows a part of block diagram of proposed clock synchronization system. External Crystal Oscillator (XO) provides a reference clock with free-running frequency offset. N-phase clocks having the same frequency with a reference clock are generated by the multi-phase clock generator. One of them is dynamically selected as a slave clock by N:1 MUX according to phase control code from phase controller (PC).

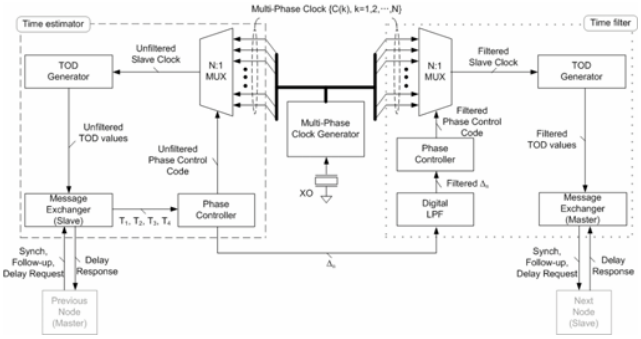


Figure 7: Complete block diagram of proposed clock synchronization system

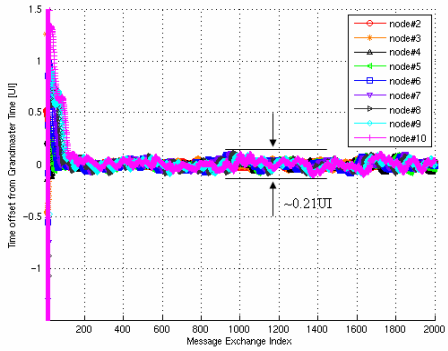


Figure 8: Time offsets of serially chained nine nodes from the time of grandmaster. (Filtered TOD value)

TOD value is counted up by TOD generator triggered by the slave clock. A block named slave message exchanger makes handshaking with master node and delivers results such as T_1, T_2, T_3 and T_4 to PC. Finally, the PC calculates Δ_n and K_n and counts up or down the phase control code at every K_n clock cycles.

B. External filtering

Intuitively, low-pass filtering can be considered to reduce dithering. By inserting an LPF into time estimator loop, high-frequency dithering is obviously reduced. However, LPF does not change detecting resolution of TOD-based message exchange, maintaining the total peak-to-peak value of dithering as ±1UI. We used an external LPF instead of internal one.

A complete block diagram of proposed clock synchronization system is given in figure 7. The system consists of XO, multi-phase clock generator, and two parts with dashed and dotted boxes, respectively. The part within dashed box aforementioned in figure 4 is named as a time estimator (TE) because its TOD value is not absolutely the same with master's one but dithers around it. With this reason, the word 'unfiltered' is attached in front of all names of nodes in the TE block.

On the other hand, the part with dotted box is newly added and named as time filter (TF). This part has a similar structure to the TE except for addition of digital LPF. Digital LPF in TF block receives Δ_n from PC in TE and filters high frequency component out to mitigate fluctuation of time offset. Filtered Δ_n in turn is fed into a PC in the TF block and used for calculation of phase shift period (K_n). Finally, filtered slave clock with filtered frequency offset is generated and results in a new TOD value. This TOD value is used for another message exchange with the next slave node.

It is remarkable that deterministic characteristics of the phase control method make external filtering possible. With mismatch between two devices that generates frequency offset within TE and TF respectively, control signal make different effects for each one so that TF generates clock signal with different frequency.

In another MATLAB simulation, the function of TF with FIR filter is newly modeled in network nodes. It is assumed that ten network nodes are serially chained with master-slave relationship. (The maximum number of nodes is specified as ten in IEEE 1588 standard [5].) Each node (except for grandmaster) tries to synchronize its time and frequency to its prior node. Other conditions of the previous simulation were maintained. Figure 8 shows the time offsets of each node from the time of grandmaster with FIR LPF having cut-off frequency of 0.028 Hz with 32 taps and 16-bit resolution. The simulation shows that the final node has the worst case peak-to-peak time offset of 0.21 UI.

5. Conclusions

We propose a new structure of time synchronizer for network synchronization. Finite frequency resolution and granularity of TOD are pointed out as the cause of time offset fluctuation. By directly controlling the clock phase, effective frequency resolution can be maximized and control gain becomes deterministic. With fixed control gain, a new approach with an external filter block is proposed and verified using MATLAB simulation. The simulation with serially chained ten nodes shows that the fluctuation of time offset is reduced from 3UI to 0.21UI by external filtering.

Acknowledgments

This work was supported by the IT R&D program of MIC/IITA. [2007-S012-02, Multimedia Convergence Network on Chip Technology Development]

References

1. Geoffrey M. Gardner and Kees den Hollander, "Analysis of Clock Synchronization Approaches for Residential Ethernet," *Proceedings of the 2005 Conference on IEEE 1588*, Oct. 10-12, Zurich.
2. John C. Eidson, *Measurement, Control, and Communication Using IEEE 1588: Chapter 5 Practical Issues in Implementing IEEE 1588,* Springer, 2006.
3. Do-Hwan Oh, Deok-Soo Kim, Suhwan Kim, Deog-Kyoon Jeong, and Wonchan Kim, "A 2.8-Gb/s All-Digital CDR with a 10-bit Monotonic DCO", *ISSCC Dig. Tech. Papers*, pp.222-223, Feb., 2007.
4. Jerry Lin, "A Low-Phase-Noise 0.004-ppm/Step DCXO With Guaranteed Monotonicity in the 90-nm CMOS Process," *IEEE J. Solid-State Circuits*, pp. 2726-2734, Dec., 2005.
5. IEEE Std™ 1588-2002, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control System*, IEEE, Nov. 8, 2002.