International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:12, No:1, 2018

FPGA Implementation of Adaptive Clock Recovery for TDMoIP Systems

Semih Demir, Anil Celebi

***Abstract*—**Circuit switched networks widely used until the end of the 20th century have been transformed into packages switched networks. Time Division Multiplexing over Internet Protocol (TDMoIP) is a system that enables Time Division Multiplexing (TDM) traffic to be carried over packet switched networks (PSN). In TDMoIP systems, devices that send TDM data to the PSN and receive it from the network must operate with the same clock frequency. In this study, it was aimed to implement clock synchronization process in Field Programmable Gate Array (FPGA) chips using time information attached to the packages received from PSN. The designed hardware is verified using the datasets obtained for the different carrier types and comparing the results with the software model. Field tests are also performed by using the real time TDMoIP system.

***Keywords*—**Clock recovery on TDMoIP, FPGA, MATLAB reference model, clock synchronization.

I. Introduction

CIRCUIT switched networks, widely used until the end of the 20th century, leave their place to PSN. This transformation that is continuing over two decades is carried out by TDMoIP equipment which interconnects circuit switched equipment to package switched equipment. TDMoIP systems are responsible for moving synchronous clock signals of asynchronous networks on circuit switched systems. For this purpose, there are several clock synchronization methods that have been proposed in the literature. Improved clock synchronization algorithms are used on real time systems. Proposed methods aim to provide the most accurate results, in the shortest time and with the least processing power. These methods are also robust to the undesired effects such as sequence shift, variable delay, and package corruption.

Packages arriving at the receiver are not necessarily in the order in which they are transmitted to the PSN. Thus, the packages received are held at the receiver for a period of time for delay compensation, and are aligned accordingly. Reordered packages are separated by estimated clock frequency of sender system, then they are sent to circuit switched network (CSN). Otherwise, Jitter buffer with certain size in which packages are held can be in overflow or underrun state. General structure of a TDMoIP system is illustrated in Fig. 1.

There are three considerable methods proposed in the literature for clock synchronization problems in TDMoIP systems. The first approach is the GPS as presented in [1]. Although high accuracy clock synchronization can be achieved with GPS method, the integration of a GPS into each system is costly. The second approach for clock synchronization is Precision Time Protocol (PTP) [2]. Although PTP methods have no any other requirements, it cannot be applied in case of transmission at different speed of carrier data on same ethernet port. The third method is called adaptive clock synchronization. An estimation by using the difference between timestamps of received packages in the receiver system or variance of jitter buffer level in certain period [3]-[9] is performed in this method.

In [3], both timestamp and jitter buffer level are used in order to achieve clock synchronization. The buffer level changes are compensated after sender's clock signal is acquired. In order to estimate the frequency to be used in the receiver, time-based phase locking method is used in [4]. Package switch network delay and frequency change tendency parameters are used in the method proposed in [5]. The difference between the clock signals of both the receiver and sender is estimated by using the maximum likelihood method for the mean square error value of the parameters. In reference [6], time synchronization processes are performed by filtering the time information of the packages that are over-delayed on the jitter buffer in the receiver unit. Unlike other methods, both synchronous and adaptive clock synchronization approaches have been utilized in [7]. There are also other methods that provide clock synchronization with only jitter buffer level [8], [9].

Time synchronization processes are defined by causal mathematical models that produce results that are dependent on a particular input, which is indispensable in their performance. Implementation of clock signal estimation process within a short time and with a high accuracy is highly affected from the performance of the mathematical infrastructure. Furthermore, the hardware platform on which these processes are being carried on are also have considerable impact on this process.

TDMoIP systems must be synchronous with circuit switched systems. For this reason, TDMoIP systems contain application specific integrated circuits (ASIC) or FPGA. FPGA can be a good choice for non-standard applications where a fabricated ASIC does not provide any flexibility since it is already fabricated without reconfiguration capability. In this study, clock synchronization process is implemented on FPGA by using labeled timestamp information on the packages received from PSN. Estimation approach is

S. D. is with the Department of Electronic and Communication Engineering, Kocaeli University, Kocaeli, Turkey (phone: 090-539-2201517; e-mail: d.semihdemir@gmail.com).

A. C. is with the Department of Electronic and Communication Engineering, Kocaeli University, Kocaeli, Turkey (e-mail: anilcelebi@kocaeli.edu.tr).

implemented by the hardware architecture proposed in this work.

The remaining of this paper is organized as follows. Clock synchronization method with timestamping is explained in Section II. In Section III, hardware architecture is explained in detail. Experimental results are presented in Section IV.

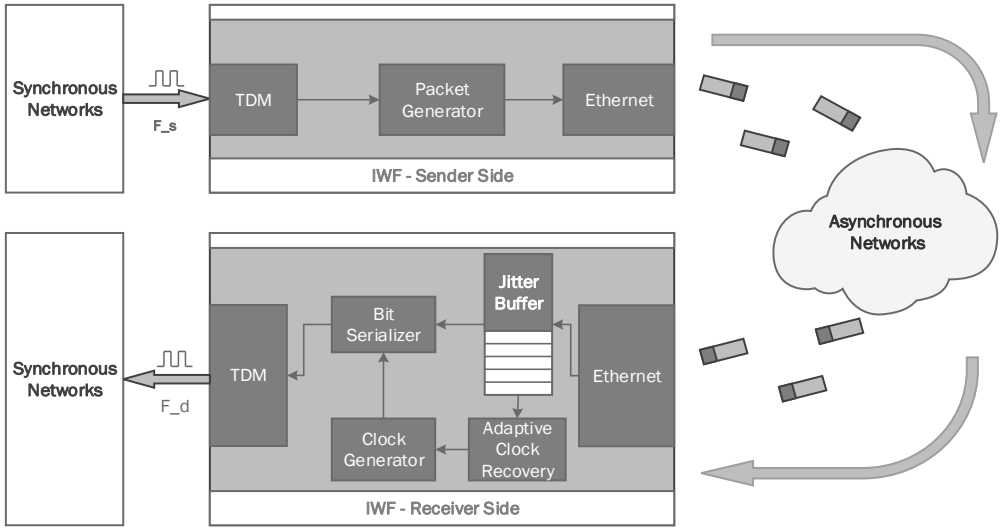

Fig. 1 General structure of a TDMoIP system

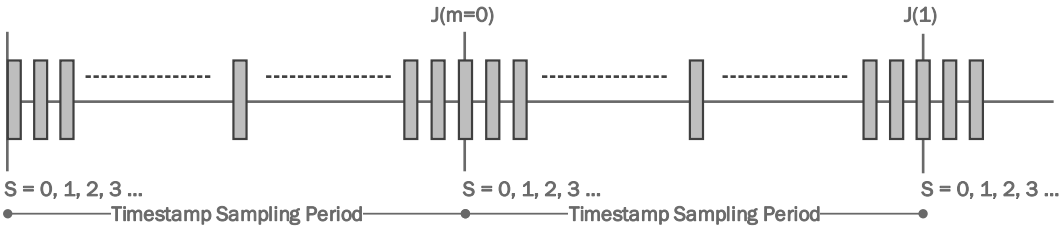

Fig. 2 Clock synchronization algorithm – sampling periods

II. Mathematical Model of the Adaptive Clock Synchronization

In this section, mathematical method used for clock frequency estimation in the proposed hardware architecture is explained in detail. The adaptive clock synchronization algorithm provides precisely corrected value of clock frequency. To achieve this, it utilizes the timestamps generated by local clock that indicates arrival time of ethernet packages in receiver side.

As shown in Fig 2, real time system gathers the timestamp information from ethernet packages during timestamp sampling period. The obtained timestamp information is evaluated at the end of sequential clock adjustment period, and then, clock estimation is performed. The S variable points to the labeled timestamp of a package in each synchronization group and the m variable represents number of completed clock synchronization period from the beginning. It is labeled with a timestamp until J(m) packages are placed in jitter buffer.

$$P = \frac{f_s}{f_d} \tag{1}$$

The variable f_s indicates the frequency of the clock signal that may differ according to the type of the TDM on the sender side. The ethernet package is generated with the f_d clock on sender side. The frequency f_d indicates the frequency of the service clock signal for the TDMoIP systems on the receiver side. P indicates the ratio of the frequency of the receiver and sender clock signals. The timestamp value of the ethernet package arriving at the receiver side, generated by the service clock signal, can be expressed as follows.

$$T_{ji} = C_{f_d} + f_d \, x \left(\tau_{fixed} + \tau_{ji} + \frac{(jx2\,xN+(i-1))x\,l}{f_s}\right) \tag{2}$$

l: Ethernet packet length (bits), N: Total sample count in one period, τ_{fixed}: On IP networks constant delay (s), τ_{jk}: IP networks variable delay-jitter (s), R_{ji}: timestamp of j.sample of i.window, C:Offset value of timestamp, f_d: Clock frequency of receiver device, f_s: Clock frequency of sender device. However, timestamp value is a counter value counted by service clock frequency in real time systems. Two consecutive decision windows are established to calculate the ratio between the frequency values of the receiver and sender clock signals. The collected time information of the packages

is set as the first window, when half of the total number of samples is reached. Time information for the second window is collected until the total number of samples is reached. In (3), the clock signal ratio between the sender unit and the receiver unit is obtained by using the time information of the Ethernet packages. Due to the fact that the examples in a long window are taken into account, the algorithm inherently behaves as a filter that passes low frequencies.

$$T_{diff} = \frac{1}{N} x\left(\sum_{k=N+1}^{2N} T_{ji} - \sum_{k=1}^{N} T_{ji}\right) \quad (3)$$

$$P_j = \frac{1}{Nxl} x T_{diff}$$

The operation of increasing the total number of samples in a window, decreasing the size of the ethernet packages, and getting the arithmetic mean of the results obtained after repeating a certain amount of this sequence, has the effect of reducing the error margin in the TDMoIP clock synchronization process.

Although the ratio between the clocks signs is estimated at high accuracy using the generated time information, this algorithm is weak against buffer overflow or discharge probabilities in high-order package switching networks. In order to solve this problem, the P_j ratio, which is calculated in the case that the running buffer reaches these threshold values, is compensated by determining the threshold values at the jitter buffer level. After completion of an estimation period, the clock signal is corrected, and time samples are taken in accordance with the new clock frequency in the next synchronization period. In the clock synchronization decision intervals, it is observed that the system is more stable and the clock frequency of receiver system has lower jitter and wander requirements as given in ITU-T G.823 [10] when more samples are collected and less frequent clock frequency is changed.

III. RTL Design of Adaptive Clock Synchronization

This section describes the implementation of clock synchronization in FPGAs for TDMoIP systems. In order to estimate the time difference, labeled time information is used during the reception of ethernet packages that is received from the PSN. The received ethernet packages are queued in the jitter buffer and it is sent to the synchronous CSN with the estimated clock frequency as result of the clock synchronization process. If clock frequency of the receiver system differs from the clock frequency of the sender system, as a result overflow or underrun situations can be occurred in the jitter buffer.

In order for the clock synchronization process to be highly precise, the variable delay effect due to the PSN must be kept minimum. To eliminate the variable delay effect, a high number of samples of time information are required. Clock synchronization register transfer level (RTL) design is implemented with Verilog hardware description language. The reference model is implemented with MATLAB. The reference model is used to verify the functional correctness of the clock synchronization hardware. E1 / E2 / E3 carrier types are used to provide synchronous communication in TDM networks.

Division operation is avoided wherever possible during FPGA implementation of mathematical equations. If it is inevitable to carry out the division process then the operation is tried to convert into power of two's form. In this work, the division operation is removed with appropriate techniques.

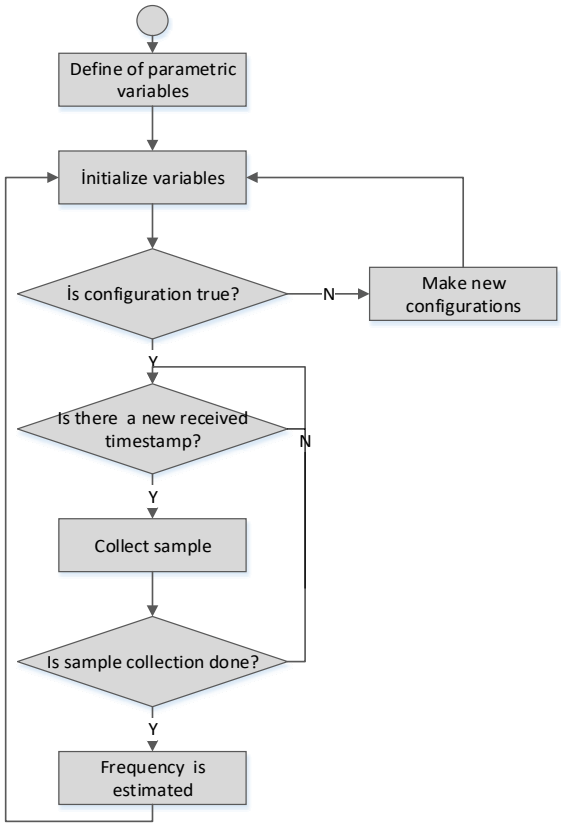

Fig. 3 Algorithm for adaptive clock synchronization RTL

TABLE I
Configuration Parameter of Clock Synchronization IP Block

Parameter	E1 Type	E2 Type	E3 Type
Package Length (byte)	2^8 - 2^10	2^8 – 2^10	2^8 – 2^10
Number of Packages	2^11 – 2^19	2^11 – 2^19	2^11 – 2^19
Jitter Buffer Delay(ms)	1… 10^3	1… 10^3	1… 10^3
Estimation Period (sec.)	2 – 524	2 – 524	2 – 524

The division operations are implemented by shift operations. 64-bit wide variables are utilized in the clock synchronization algorithm to achieve the target accuracy. The minimum number of bits for the operations is chosen as 114 for maintaining the accuracy but defined 128 bits for practicality. Thus, all of the numbers are expanded by 50 bits. The least significant 64 bits of the results are taken into account at the end of the process. Clock synchronization algorithm is shown in Fig 3. The proposed hardware architecture can perform clock synchronization for E1, E2 and E3 type carriers. The parameters used for different types of

carriers are given in Table I.

As a result of the estimation process within the FPGA, the conversion of the estimated frequency value is given in (4) parts per million (PPM) unit.

$$f_{est} = \frac{f_{est\_by\_fpga} \, x \, 10^6}{2^{50}} \; PPM \qquad (4)$$

Table II shows the resource utilisation of the clock synchronization hardware on the Xilinx Zynq 7000 series SoC "xc7z020clg484-1".

TABLE II
FOR CLOCK SYNCHRONIZATION, RESOURCE USAGE OF THE DESIGNED HARDWARE (POST - SYNTHESIS)

Resource	Estimation	Available	Utilization %
LUT	10876	53200	20.44
FF	78	106400	0.04
IO	127	200	63.50
BUFG	12	32	37.50

IV. FUNCTIONAL VERIFICATION

The time stamps for E1, E2, and E3 carrier types are generated from the ethernet packages. These data are then used to verify the functionality of clock synchronization implementations on TDMoIP, MATLAB, and FPGA platforms. Time stamp data are generated for a slot of 16384 samples. Five slots are used for verification. Timestamps are generated by using a TDMoIP device installed on the field. The datasets generated for E1, E2 and E3 carrier types are used for verification which is implicitly used by the TDMoIP. These data are then also used for verification of MATLAB and FPGA implementations. The results are illustrated in Table III. Frequency estimates of three different platforms for E3 carrier type are illustrated in Fig. 4.

TABLE III
ESTIMATION RESULTS OF THREE DIFFERENT PLATFORMS

	FPGA		*MATLAB*		*Processor*
	Expanded 50 bit	*PPM*	*Expanded 50 bit*	*PPM*	*PPM*
E3	12913796096	11,46975501	12913815552	11,46977229	11,667237
	5057251328	4,491741493	5057195520	4,491691925	4,128165
	2809140224	2,495017725	2809147904	2,495024546	2,796388
	1333101568	1,184032044	1333072384	1,184006123	1,921939
	794478592	0,70563874	794458112	0,70562055	0,661558
E2	12350545920	10,96948836	12350590976	10,96952838	11,003752
	5375451136	4,774359695	5375442944	4,774352419	4,207819
	3002597376	2,666842192	3002292224	2,666571163	2,801929
	1437429760	1,276694093	1437509632	1,276765033	1,874928
	695193600	0,617455953	695126016	0,617395926	0,611522
E1	12141936640	10,7842061	12142272512	10,78450441	10,772236
	5538316288	4,919013008	5538308096	4,919005733	4,273293
	3184181248	2,828121069	3183779840	2,827764547	2,878435
	1591132160	1,413209247	1590657024	1,412787242	1,934645
	540229632	0,479820301	540418048	0,479987648	0,672164

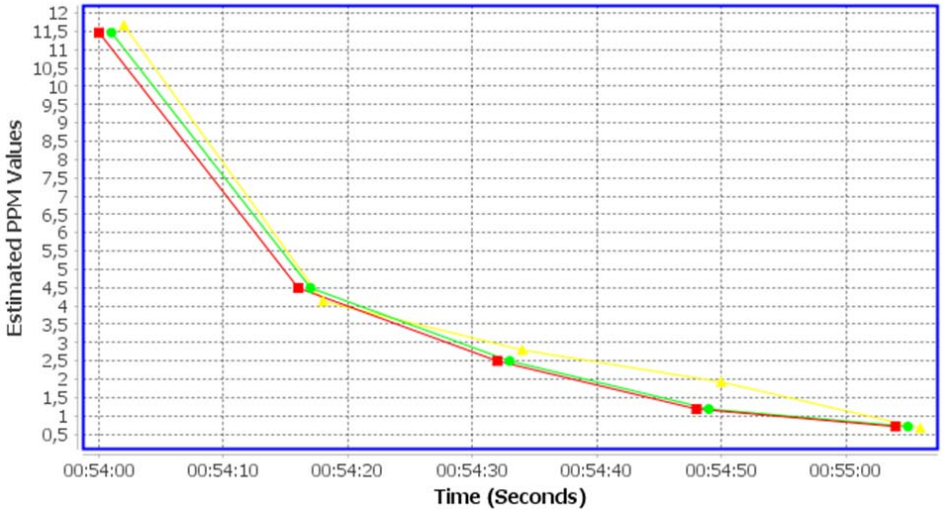

Fig. 4 E3 Traffic estimation result

V. CONCLUSION

In this study, an adaptive clock synchronization in TDMoIP systems is implemented on FPGA using Verilog hardware description language. Real time TDMoIP data are used for verification. According to the results, it is concluded that FPGAs can considerably give fast response for time critical running systems compared to the CPU based implementations in the current TDMoIP systems.

REFERENCES

[1] Lewandowskı W., Azoubıb J., and Klepczynskı W. J. "GPS: Primary Tool for Time Transfer", Proceedings of the IEEE, vol. 87, no. 1, January 1999.

[2] Giorgi G., Narduzzi C. "Performance Analysis of Kalman-Filter-Based Clock Synchronization in IEEE 1588 Networks", IEEE Transactıons on Instrumentatıon and Measurement, vol. 60, no. 8, August 2011.

[3] Zhu S., Xu Y., Zhang J., Wang Y., "The Study ard Analysis of joint Adaptive Clock Recovery,Mechanism for TDMOIP", IEEE International Conference on Networking, Sensing and Control, 2008.

[4] Aweya J., Ouellette M., Montuno D. Y., Felske K., "Circuit emulation services over Ethernet—Part 1: Clock synchronization using timestamps", Internatıonal Journal Of Network Management, 14: 29–44, 2004.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[5] Trump T., "Estimatıon Of Clock Skew In Telephony Over Packet Swıtched Networks" IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000.
[6] Umayabashi, M., Cui Z., Takagi K., Iwata A., "Proposal of High Stable Clock Recovery Method Based on Queuing-Delay Measurement Scheme for TDM PWE3 over PSN", OSA/OFC/NFOEC, 2009
[7] Shen Y., Ge N., "Dual-Stage Clock Recovery for TDM in Packet Networks", Proceedings ofISCIT, 2005.
[8] Shen G., Nizam M. H. M., Liu E., Gui L., Xu X., "Fast and Accurate Clock Recovery in Packet Switched Networks", National 973 program, 2004
[9] Shuai L., Yueming L., Yuefeng J., "A Buffer Level Clock Recovery Based On Least Mean Squares In Ptn" IC-BNMT, 2009
[10] ITU-T Rec. G.823, The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy, 2000.