Network Synchronization Coding for Wireless Sensor Network Device

Pattem Sampath Kumar
Department of ECE,
Malla Reddy Institute of Technology,
Hyderabad, AP, India.

K.Jaya Sankar
Department of ECE,
Vasavi college of Engineering,
Hyderabad, AP, India.

V.Sumalatha
Department of ECE,
JNTU college of Engineering,
Ananthapur, AP, India.

Abstract
Wireless sensor networks are emerging at faster rate in practical industrial applications. During the communication process the packets are transferred via wireless media. The precision of data transferred is dependent on the synchronization approach followed. However in case on wireless sensor network where the devices are located at different range, processing variant of measured data, the Precision Clock Synchronization Protocol for distributed Network and Control operation is needed . The IEEE-802.3is defined as the communication standard in such network. In this standard the protocol is defined for clock synchronization based on link to link synchronization. However in distributed network It isrequired to enable accurate synchronization of clocks with varying precision, resolution and oscillator stability in a distributed manner. In this paper a synchronization approach based on a hierarchical synchronization process of master and slave principal.

Keyword
Clock synchronization, distributed system, data precision, wireless sensor network, master-slave hierarchical synchronization.

I. INTRODUCTION

Wireless sensor networks have been proposed extensively over thepast several years as a means of alleviating instrumentation costsassociated with structural health monitoring of civil infrastructure.However, low data throughput, unacceptable packet yield rates, andlimited system resources have generally plagued many deployments bylimiting the number of sensors and their sampling rate. The sensornetworks present challenges in three broad areas: energy consumption,network configuration and interaction with the physical world.Therefore, the development of sensor networks requires technologiesfrom three different research areas: sensing, communication, andcomputing (including hardware, software, and algorithms). The nextgeneration of the structural health monitoring sensors needs to be lowcost,low-power, self-healing, self-organized, and compact

Many of today's instrumentation used for wireless sensor operation require sophisticated timing of a variety of I/O functions (such as analog input, analog output, and digital I/O). PXI and data acquisition (DAQ) hardware offer an ideal solution to many of these timing with their very precise, modular, and easy-to-program synchronization features. A common class of test systems with some of the most demanding timing requirements is the simulation-response system. The stimulus-response system is characterized by reacting to one or more simultaneous excitation pulses by emitting one or more simultaneous signal pulses, usually after a known and very short time delay. In many cases, this stimulus-wait-response cycle repeats rapidly and continuously. In all cases, the instrumentation challenge of creating such a stimulus-response system is synchronizing the various output stimulus signals with respect to the acquisition of the response signals with a precisely timed delay.

In a wireless sensor network, synchronization is achieved by using data rate synchronization, clock frequency synchronization, or clock recovery synchronization techniques. The communication system includes a conversion clock at a transmitter and a conversion clock at a receiver. The data rate synchronization techniques adjust data rates by resampling digital data sent from the transmitter to the receiver to compensate for frequency variations in the conversion clocks. The clock frequency synchronization techniques use time stamps to adjust the receiver conversion clock frequency to substantially match the transmitter conversion clock frequency. The clock recovery techniques use a reference clock to adjust the transmitter conversion clock frequency and the receiver conversion clock frequency to be approximately the same.

II. WIRELESS SENSOR NETWORK

Recently, wireless sensor networks have emerged that can becharacterized by local processing capabilities that minimize the amountof data transmitted in a single- or multi-hop strategy to extend thelifetime and robustness of

Manuscript received June 5, 2014
Manuscript revised June 20, 2014

the network. The multi-hop Wisden system[6], which uses the small mica motes developed at theUniversity of California at Berkeley [7], provides anexample. In this system, by avoiding the transmission of lengthy timehistories, battery life of the wireless nodes can be extended, while theissues of strict time synchronization and loss intolerance aremarginalized. The BriMon system [8] provides aneasy to deploy, long term and low maintenance system using batteryoperated wireless sensor motes as an alternative for communication anddata logging needs. While such developments in wireless sensornetworks have demonstrated their potential to provide continuousstructural response data to quantitatively assess structural health,manyimportant issues including network lifetime and stability, reliability,time-synchronization, distributed processing and overall effectivenesswhen using low-cost sensors must be realistically addressed.

The most important challenges need to be solved at the networklevel itself, apart from the coordination needed with the application layer.

Specific problems include:

1) Self-formation: The network topology should be self-adjustable, i.e.addition of new sensor devices should be handled automatically in thenetwork without manual intervention. Similarly, the sensor devicesmay drop out of the network if enough energy is not harvested. In thiscase, the rest of the network should be able to adjust and find alternateroutes for transmitting the information and coordinating the sensingactivities.

2) Time Synchronization: The distributed sensors collect information fortransmission to the local base station. The various sensors should betime synchronized such that the events causing the observation can becorrelated and uniquely identified.

3) Transceiver frequency: Selection of a suitable RF band for lowpower,non-interfering, high-throughput operation is needed.

4) Prioritization: The communication between the sensor nodes and thelocal base station could be synchronous (periodic update messages) orasynchronous (as a result of an anomalous event, which could create atrigger for the monitoring system). The messages sent asynchronouslyshould be allocated higher priority in the network, because of theiralert-like nature.

5) Hierarchy: The sensor network needs to be organized in an adaptivehierarchy based on the application requirements. The hierarchyamong the sensor nodes can ease the routing as well as provide thecapability to make distributed decisions. Distributed decisionsminimize the transmission of unnecessary raw data to the local basestation.

6) Information Storage and Retrieval: In the case of communicationfailure (with the remote central information server) due to inadequatepower for communication or interruption in communication link, thesensor network needs to be designed with limited storage capability.This storage capacity can be optimized with respect to the type andnumber of sensors.

7) Protocol Design: Standard communication protocols needs to becustomized in order to: (1) Minimize the communication overhead,and (2) Make the sensing system reliable and robust. Thiscustomization will provide application oriented network featuresunique to the continuous monitoring system.The next generation of the wireless networks for sensor applicationsneeds to be designed accordingly to resolve these important challenges.

III. COMMUNICATION MODEL

In order to provide decision making capability and perform complexsignal processing at the sensor level, dedicated reconfigurable System on-Chip (SoC) devices are closely coupled with the micro-electromechanicalsensors (MEMS). Recently, FPGA based ultrasonic signal processing hardware havebeen successfully used in real-time flaw detection [11],ultrasonic data compression [12] and parameterestimation applications [9] demonstrating its versatility. Inaddition, power and area efficient implementations based on recursivefilter structures for subband decomposition have been proposed [10]. These implementations are especially suitable for ultra-lowpower smart sensors used in structural monitoring applications.

IV. HIERARCHICAL SYNCHRONIZATION

The synchronization protocol is organized as follows;

1: Master-slave hierarchy has to organize between clocks (based on observing the clock property information contained in multicast Sync messages).

2: Each slave must be in synchronization to its master (based on Sync, delay_Req, Follow-Up, and Delay_Resp messages exchanged between master and its slave).

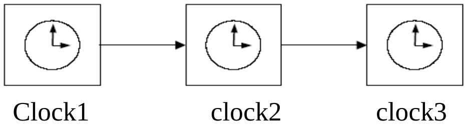


Fig1: Hierarchy of Master-Slave clocks

Clock1: Grand master clock

Clock2: slave to the grand master clock and master to its slave

Clock3: slave to its master

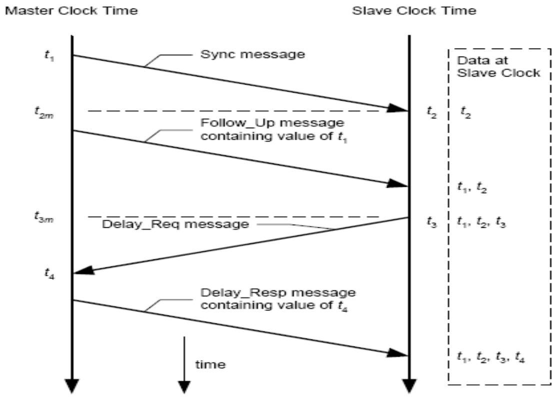


Fig 2: Offset and Delay measurement

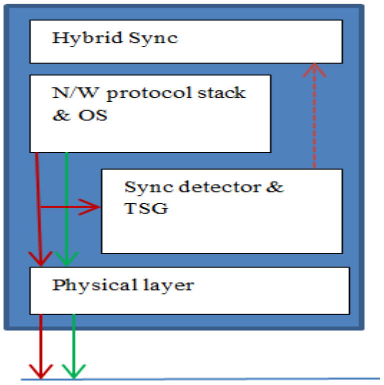


Fig 3: Message exchange for offset correction

In the above the master clock sends two messages:
1. Sync message (red color)
2. Follow up message (green color)
The arrow with red color denotes timestamp point and the line with dotted red color denotes time at which sync message passed the timestamp point (t1).
a) The characteristics of Sync Messages defined as:

- It is issued by clocks in the 'Master' state
- It is having clock characterization information
- It is having an estimate of the sending time (~t1)
- The receipt time is noted whenever it was received by slave clock
- It can be distinguished from other legal messages on the network
- For best accuracy these messages can be easily identified and detected at or near the physical layer and the precise sending (or receipt) time recorded

b) Similarly the characteristics of Follow-Up messages are defined as:

- It is issued by clocks in the 'Master' state
- Always associated with the preceding Sync message
- Contain the 'precise sending time= (t1)'as measured as close as possible to the physical layer of the network
- When received by a slave clock the 'precise sending time' is used in computations ratherthan the estimated sending time contained in the Sync message

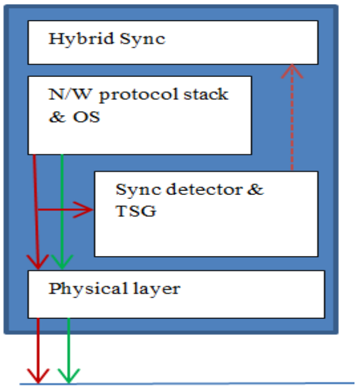


Figure 4: Message exchange for propagation delay correction

In the above figure the master clock receives delay request message (red color line) and it sends delay response message (green color line), similarly the dotted line with red color denotes time at which the delay request passed the time stamp point(t4). When there is a delay at slave the slave clock sends a delay request message to master clock and then the master clock gives response called delay response and the respective characteristics are defined as follows:

c) Delay_Req Messages:

- Issued by clocks in the 'Slave' state
- The slave measures and records the sending time (t3)
- When received by the master clock the receipt time is noted (t4)
- Can be distinguished from other legal messages on the network
- For best accuracy these messages can be easily identified and detected at or near the physical layer and the precise sending (or receipt) time recorded

d) Delay_Resp messages:

- Issued by clocks in the 'Master' state
- Always associated with a preceding Delay_Req message from a specific slave clock
- Contain the receipt time of the associated Delay_Req message (t4)
- When received by a slave clock the receipt time is noted and used in conjunction withthe sending time of the associated Delay_Req message as part of the latency calculation

Best Master Clock Selection

This phase gives the description about the best selection of master clock. The most precise clock in the network synchronizes all other clocks. There are two kinds of roles for the selected master clock: master (the one, which synchronizes the others) and slaves (those being synchronized). In principle, any clock can play either the master or the slave role. The precision of a clock is categorized by the protocol in classes (stratum). The highest class is an atomic clock that has the stratum value 1. The "best master clock algorithm"automatically gives the selection of the best clock in the network.

Selecting a Master Clock

The clock which is going to act as master clock should have the following characteristics:

- A clock at startup listens for a time called TIME_RCP_SYNC.
- A master clock (clock in the PTP_MASTER state) issues periodic Sync messages in a period called sync interval.
- A master clock may receive Sync messages from other clocks (who for the moment think they are master) which it calls 'distant masters'
- Each master clock uses the Best Master Clock algorithm to determine whether it should remain master or yield to a distant master.
- Each non-master clock uses the Best Master Clock algorithm to determine whether it should become a master.

WIRELESS SENSOR NODE TOPOOLGY

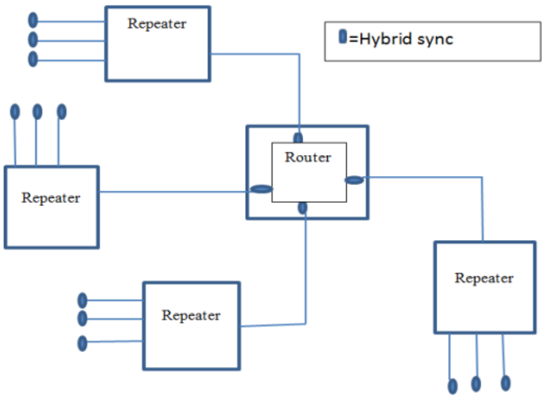


Figure 5: Boundary Clock

The following procedure was performed during the multiple subnet synchronization and master clock selection.

- Sync, Follow-Up, Delay_Req, or Delay_Resp messages should not passed by boundary clocks, thus network segment is concerned as far as the synchronization is occurred.
- A port of a boundary clock acts just like an ordinary clock within the subnet with respect to synchronization and best master clock algorithm
- The boundary clock internally selects the port that sees the 'best clock' as the single slave port. This port is a slave in the selected subnet. All other ports of the boundary clock internally synchronize to this slave port.
- Boundary clocks define a parent-child hierarchy of master-slave clocks.
- The best clock in the system is the Grand Master clock.
- If there are cyclic paths in the network topology the best master clock algorithm reduces the logical topology to an acyclic graph.

In IP multicast communication and is not restricted to Ethernet, but can be used on any network technology that supports multicasting. In point to pint (PTP) communication the network is scaled for a large number of PTP nodes because a master can serve many slaves with a single pair of Sync and Follow_ up messages. Multicast communication offers also the advantage of simplicity. IP address administration does not need to be implemented on the PTP nodes. For this reason, Delay_Req and Delay_Resp as well as the management messages, which are in fact point-to-point messages, use also multicast addressing. All other nodes but the wanted destination has to filter out these messages.

PTP Implementation

a) Time Stamping Methods

Synchronization accuracy directly depends on time stamp accuracy. There are different options to take time stamps:

The most accurate method is to detect PTP frames with hardware assistance. Ingress and egress frames pass the Media Independent Interface (MII), where frames can easily be captured and decoded. Below the MII, data is 4B5B coded, scrambled, and therefore not directly interpretable. The accuracy of this method is limited by the Physical chip timing characteristics.

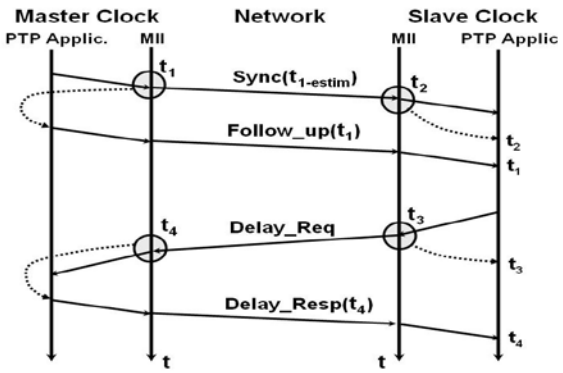


Figure 6: Time stamp transfer (circles indicate time stamping)

Without any hardware assistance, the next best place for time stamping is the network driver. Egress frames are stamped at the very latest moment before the frame is handed over to the MAC controller. Ingress frames are stamped at the entry point of the network interface interrupt service routine. The accuracy of this method is limited by the operating platform timing characteristics (e.g. interrupt latency, CPU performance) and load dependent.

Stamping on the application layer is best located at the socket interface. It does not require any modification to system software. The influence of protocol stack and load allows only moderate accuracy.

Performance Criteria

The precision of the delay and offset calculations depend on the precision of the time stamps. They should reflect the send and receive time as precise as possible. The slave‘s offset and delay calculation is based on the difference of time stamps taken at two different places (i.e. master and slave node). Therefore, the two clocks should use the same scale, i.e. the same tic interval. This is achieved by drift compensation: the slave clock‘s rate is accelerated or slowed down. A slightly different tic interval will degrade the result.

It is assumed that the message transit delay is the same for both directions. At a first glance, this is the case in an Ethernet link. However, going into the details, there are a few non-idealities: Cables used for Ethernet have a minor asymmetry by design, to reduce far end cross talk (FEXT). Ethernet transceivers have asymmetric transmit and receive paths. If their timing characteristics are clearly specified within a small range, the asymmetry can be taken into account by calculation as inbound and outbound latency correction constants. On the long run, conditions may change due to reconfiguration (leading to a very different delay) or environmental conditions (temperature). How fast the clocks can react depends on the frequency of sync and delay measurement and the dynamic behavior of the servos controlling the slave clock.

To sum it up, performance depends on:

- The symmetry of communication channel (i.e. same delay in both directions and constant over a longer period of time)
- drift compensated clocks (i.e. adjusted time base in master and slave clocks)
- time stamp accurateness
- time stamp firmness
- sync interval
- clock stability
- clock control loop characteristics
- Synchronization Methods compared with PTP

There have been various previous methods of synchronizing clocks distributed over a network: the most common are the Network Time Protocol (NTP) and the simpler Simple Network Time Protocol (SNTP) derived from it. These methods are quite common in LANs or in the Internet and allow accuracy right down to the millisecond range. Another possibility is the use of radio signals of the GPS satellites. However, this requires relatively expensive GPS receivers for every clock and the appropriate antennae on the roof and the necessary cabling. Although this provides a high precision clock, it is often impractical for reasons of cost and effort.

The Objective of this paper is to prove that the speed of communication has improved by implementing the Hierarchical Synchronization protocol over the existing communication link. For this purpose we need to develop the communication link without Hierarchical Synchronization protocol and with the Hierarchical Synchronization protocol.

V. OPERATIONAL MODELING

Block diagram with Hierarchical Synchronization protocol differs from the actual diagram in the interface part. The interface block is split into 2 units.

1. Synchronization unit
2. Clock Generation unit

The synchronization unit is used to select the master clock from the existing clocks and it sends that information in a setup packet to the Clock generation unit where it compares the master clock with the clocks from the individual systems and corrects the device clocks to the master clock thereby enhancing the speed of communication.

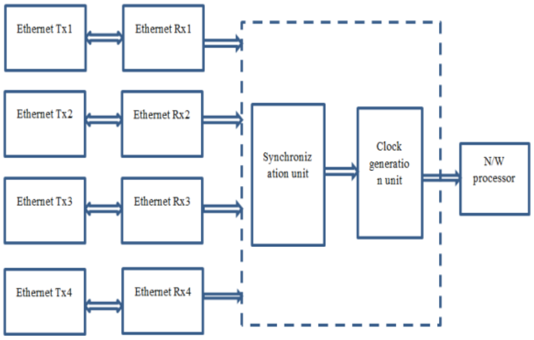
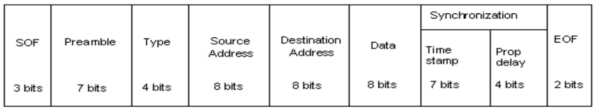


Fig 7: Block diagram of proposed hierarchical synchronization approach

The block diagram above is the top module for communication with the help of hierarchical synchronization protocol. It is performed in three stages,1) Communication Link, 2) Interface, 3) Network Processor.

Communication link is the first stage. Since the objective is to prove that speed is enhanced while various devices connected are operating at different frequencies,the link designed here consists of 4 channels each channel operating at different frequencies. Each channel having a transmitter and a receiver. In this the Ethernet is using as medium the transmitter and receiver should also Ethernet transmitter and Ethernet receiver.The transmitter device collects data from the hosts and transmits that data through the receiver to the network processor for the corresponding control signals. The data from the host is taken as input and the packet is formed by the frame builder unit. CRC generator is used to calculate the Frame Check Sequence and add it to the packet. Then the build packet is transmitted through the serial to parallel converter to the receiver.

Frame Format



SOF	Preamble	Type	Source Address	Destination Address	Data	Synchronization		EOF
						Time stamp	Prop delay	
3 bits	7 bits	4 bits	8 bits	8 bits	8 bits	7 bits	4 bits	2 bits

Fig 8: Frame format developed protocol

The frame format of the setup packet is shown above. A synchronization field is added to the packet which provides the information that is necessary for synchronizing in addition to the packet format having in conventional frame format. It consists of 2 internal fields, time stamp and the propagation delay. These two fields specify the offset value and the propagation delay value of the slave clocks from the master clock. The devices which receive this packet will take these two fields and corrects its clock to the master clock. In this design the propagation delay is assumed to be zero since no physical medium is involved. The flow of the design is similar to that without synchronization.

The above figure gives the graphical representation of the design process carried out in the proposed approach. The host machine which needs to communicate with the processor sends its data to the FIFO of the transmitter device and depending on the status of the control signal read or write operation is performed and then the data is forwarded to the CRC generator where the FCS is generated and also to the frame builder where the frame is build and then the parallel data is converted to serial for transmission and send over the network.

The data is transmitted over the network and at the receiving side the data is reconverted into parallel form and this data is fed into the FIFO and send to the frame reader where the is decomposed into different fields. The destination address field is fed to AML to check if the frame is addressed to this particular receiver or not. If the address matches then the FCS is fed to CRC checker to check for its correctness and if the frame is proved to be received without any error then the frame is processed else the frame is discarded.

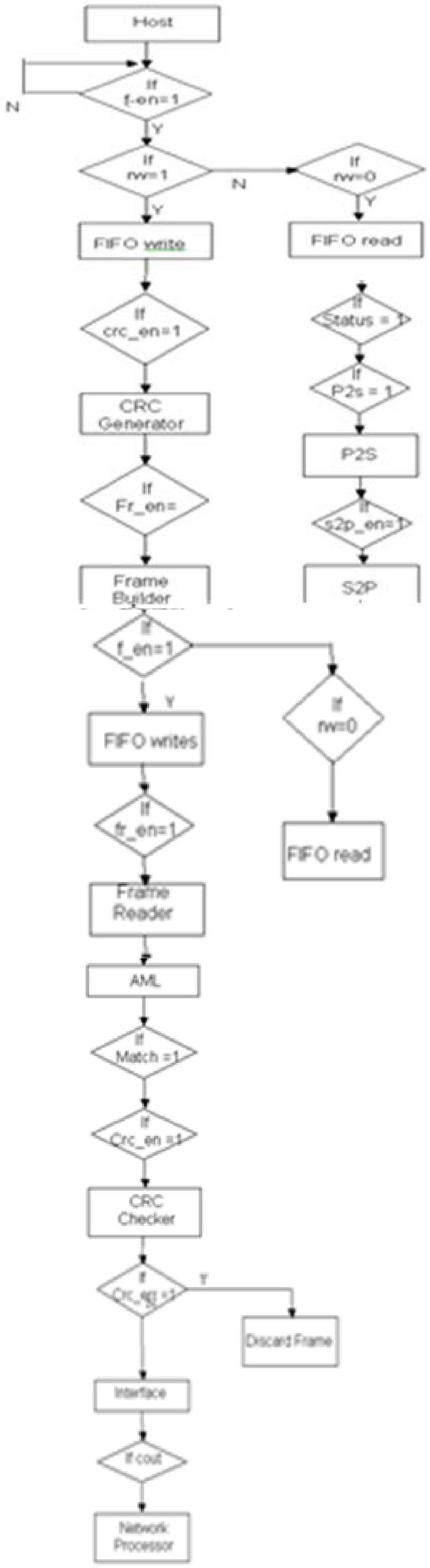


Fig 9: operational flowchart of the wireless sensor operation

VI. SIMULATION OBSERVATIONS

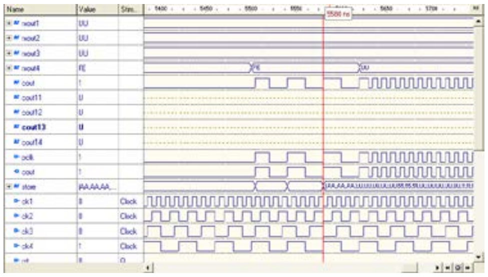


Fig 10: Timing simulation result for the communication unit following conventional link synchronization approach.

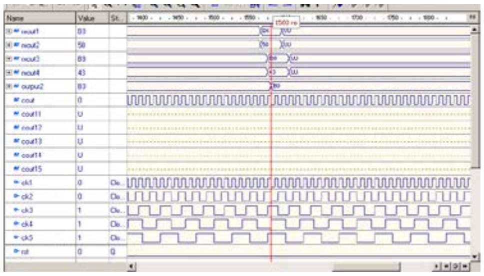
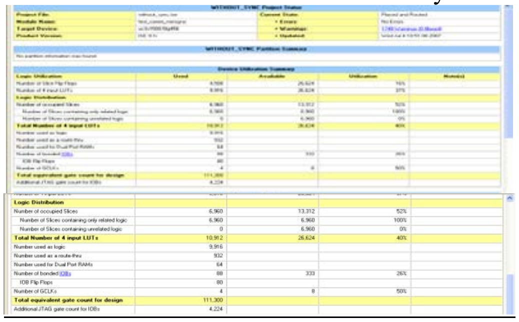


Fig 11: Timing observation result for the developed communication unit following proposed hierarchical coding unit.

DLC=3bytes
Best master clock (BMC)= min (tstamp)
Default clock = clk1 (under asynchronous mode communication)
Frequency selection method = round robin
Total number of communicating nodes = 4
Total amount of data generated per node = 3 bytes
Total amount of expected data in processor = 12 bytes
Total time taken = 5580 ns (under non synchronous round robin based comm)
(Total time = processing time + comm Time)
Total time taken = 1445 ns (under 1588 synchronous mode comm)
Total time saved (ts)= 4135 ns
Total clock cycles saved = ts / BMC
=4135 / 10 ≈ 413 cycles



WITHOUT_SYNC Project Status

Project File:	[illegible]	Current State:	Placed and Routed
Module Name:	[illegible]	• Errors:	No Errors
Target Device:	[illegible]	• Warnings:	[illegible]
Product Version:	[illegible]	• Updated:	[illegible]

WITHOUT_SYNC Partition Summary

[illegible]

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	
Logic Distribution				
Number of occupied Slices	6,960	13,312	52%	
Number of Slices containing only related logic	6,960	6,960	100%	
Number of Slices containing unrelated logic	0	6,960	0%	
Total Number of 4 input LUTs	10,912	26,624	40%	
Number used as logic	9,916			
Number used as a route-thru	932			
Number used for Dual Port RAMs	64			
Number of bonded IOBs	88	333	26%	
IOB Flip Flops	80			
Number of GCLKs	4	8	50%	
Total equivalent gate count for design	111,300			
Additional JTAG gate count for IOBs	4,224			

Fig12: Synthesis report of the design without synchronsation

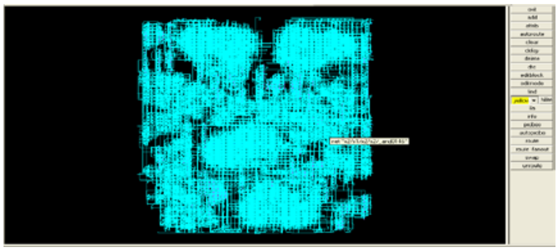
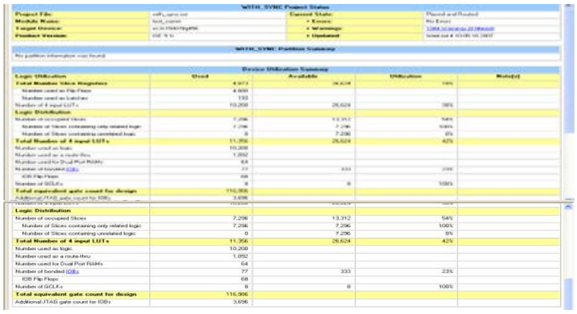


Fig 13: Implementation routing and placement logic for developed approach on vertex 2p FPGA device



WITH_SYNC Project Status

Project File:	[illegible]	Current State:	Placed and Routed
Module Name:	[illegible]	• Errors:	No Errors
Target Device:	[illegible]	• Warnings:	[illegible]
Product Version:	[illegible]	• Updated:	[illegible]

WITH_SYNC Partition Summary

[illegible]

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	[illegible]	[illegible]	[illegible]	
Number used as Flip Flops	[illegible]			
Number used as Latches	[illegible]			
Number of 4 input LUTs	10,200	26,624	38%	
Logic Distribution				
Number of occupied Slices	7,296	13,312	54%	
Number of Slices containing only related logic	7,296	7,296	100%	
Number of Slices containing unrelated logic	0	7,296	0%	
Total Number of 4 input LUTs	11,356	26,624	42%	
Number used as logic	10,200			
Number used as a route-thru	1,092			
Number used for Dual Port RAMs	64			
Number of bonded IOBs	77	333	23%	
IOB Flip Flops	68			
Number of GCLKs	8	8	100%	
Total equivalent gate count for design	116,906			
Additional JTAG gate count for IOBs	3,696			

Fig14: Logical synthesis report for the developed synchronization approach

VII. CONCLUSION

In this paper a new synchronization protocol for data transmission using hierarchical synchronization processing is proposed. In the proposed approach the timing synchronization is obtained more effectively by introducing the time stamp logic for driving operation. The proposed Hierarchical synchronization coding scheme derives the synchronization bits from the on-run master clock frequency and the sub clock frequency are more precisely been used for synchronization process. The developed synchronization reduces the operation timing of the data reception by faster synchronization of the master frequency and intern developing synchronous sub clock synchronization frequency.

REFERENCES

[1] D.Sanz, M.Ruiz, J.M.Lopez,R.Castro,J.Vega, E.Barrera, “IEEE1588 V2 Clock Distribution in FlexRIO Devices: Clock Drift Measurement”, 19th IMEKOTC4 Symposium and 17th IWADC Workshopon Advances in Instrumentation and Sensors InteroperabilityJuly 18-19, 2013

[2] Jae Hun Cho, Hoon Kim, Sihai Wang, Jaehoon Lee, Hanlim Lee, Seongtaek Hwang,Sungdae Cho, Yunje Oh, and Tae-Jin Lee, “A Novel Method for Providing Precise TimeSynchronization in a Distributed ControlSystem Using Boundary Clock”, IEEE Transactions on Instrumentation and Measurement 2009.

[3] Gabriel J. Garcia, Carlos A. Jara, Jorge Pomares, AimanAlabdo, Lucas M. Poggi and Fernando Torres, “A Survey on FPGA-Based Sensor Systems: Towards Intelligent and Reconfigurable Low-Power Sensors for Computer Vision, Control and Signal Processing”, Sensors2014.

[4] Thomas Beluch, Daniela Dragomirescu, Florian Perget, Robert Plana, “Cross-layered Synchronization Protocol forWireless Sensor Networks”, Ninth International Conference on Networks, Les Menuires : France ,2010.

[5] Cheng Fang Zhen,Wenyi Liu,Yongrui Liu, and Anbin Yan, “Energy-Efficient Sleep/Wake Scheduling for Acoustic Localization Wireless Sensor Network Node”,International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, Volume 2014.

[6] Xu, N., Rangwala, S., Chintalapudi, K.K., Ganesan, D., Broad, A., Govindan, R. and D.Estrin, “A wireless sensor network for structural monitoring”, Proceedings ofthe 2nd international conference on Embedded networked sensor systems, 2004.

[7] Horton, M., Culler, D., Pister, K. S. J., Hill, J., Szewczyk, R. and Woo, A.,“MICA:The Commercialization of Microsensor”, 2002.

[8] Chebrolu, K., Raman, B., Mishra, N., Valiveti, P. K. and Kumar R, “BriMon: Asensor network system for railway bridge monitoring”, 6th Annual InternationalConference on Mobile Systems, Applications and Services, MobiSys, 2008.

[9] Lu, Y., Oruklu, E. and Saniie, J. “Fast Chirplet Transform with FPGA-BasedImplementation”, IEEE Signal Processing Letters, vol.15, pp.577-580, 2008.

[10] Oruklu, E., Weber, J. and Saniie, J. “Recursive filters for subband decompositionalgorithms in ultrasonic detection applications”, IEEE Ultrasonics Symposium, pp.1881-1884, 2008.

[11] Weber, J., Oruklu, E. and Saniie, J., “Configurable hardware design for frequencydiversetarget detection”, IEEE 51st Midwest Symposium on Circuits and Systems,MWSCAS 2008, pp.890-893, 2008.

[12] Oruklu, E., Maharishi S. and Saniie, J.,“Analysis of Ultrasonic 3-D ImageCompression Using Non-Uniform, Separable Wavelet Transforms”, IEEE Ultrasonic Symposium, pp. 154–157 , 2007.