FPGA IMPLEMENTATION OF FLEXRAY CLOCK SYNCHRONIZATION MODULE IN NORMAL ACTIVE STATE OF POC MODULE

Naveena S George
M tech, ECE
Mar Athanasius College of Engineering,
Kothamangalam, Kerala

Shan Mary Cherian
M tech, ECE
St. Joseph's College of Engineering & Technology
Palai, Kerala

Sherin Mary Enosh
M tech, ECE
St. Joseph's College of Engineering & Technology
Palai, Kerala

Jiss Paul
Associate Professor, ECE
Mar Athanasius College of Engineering,
Kothamangalam, Kerala

*Abstract— **Modern automobiles are not merely mechanical devices, but they are becoming highly sophisticated by including more and more functions which are controlled by small digital computers known as Electronic Control Units (ECUs). As new functions are included, there is not only demand on ECU, but there is an increasing demand on Communication networks placed in the automobile. FlexRay is a scalable, flexible, high-speed, deterministic, error-tolerant communication technology that is designed to meet growing safety related challenges in the automobile industry. Synchronization between different nodes in a communication network is very essential for the proper working of a system. The clock synchronization service in FlexRay protocol suggests a distributed control system. In this paper the clock synchronization algorithm named Fault Tolerant Midpoint (FTM) Algorithm is studied and its implementation in Normal Active State of Protocol Operation Control (POC) Module is achieved with the help of Stratix IV FPGA.***

Keywords— FlexRay, Communication Controller, POC, Normal Active State, Clock Synchronization, FTM, Microtick, Macrotick, Communication Cycle.

I. INTRODUCTION

The FlexRay protocol is a safety oriented, time-triggered protocol that It supports both deterministic data that arrives in a predictable time frame and dynamic event-driven data that arrives in an unpredictable time frame. The development of FlexRay started as a response to meet the future demands on high speed networks and reliability in automotive applications. Current protocol standards like CAN and LIN will not be able to meet all these demands. The main feature of FlexRay is that it offers high bandwidth (up to 10 Mbit/s per channel), determinism and fault-tolerance. It is developed by the FlexRay consortium which consists of many car manufacturers like BMW, DaimlerChrysler, General Motors and Volkswagen. FlexRay protocol consists of a number of factors that makes it excellent for both safety related applications and high speed networking applications. One of those factors is the support for two channels for communication, which can be used to send either redundant data for reliability or for increasing the bandwidth.

The data transfer in FlexRay protocol is structured into the form of communication cycles. These communication cycles are periodic in nature and have got fixed length. The communication cycle consists of a static segment, a dynamic segment, a symbol window and the network idle time (NIT). Out of these static segment and NIT are mandatory and dynamic segment and symbol window are optional. The static segment is used to transmit cyclic information. It is included in every cycle, and uses the TDMA method. The dynamic segment is used to transmit non-regular data. It is optional in a FlexRay-cycle, and it utilizes the FTDMA method. The Symbol Window can be used to transmit test symbols, e.g. for the use with a bus guardian. The Network Idle Time is used by the nodes to perform the necessary time synchronization.

A FlexRay communication system (FlexRay Cluster) is made up of a number of FlexRay nodes which are also known as electronic control units (ECU) and a physical transmission medium (FlexRay Channel) interconnecting all of these FlexRay nodes. This FlexRay node consists of a Microcontroller (µC, also called as a host), a Communication Controller (CC), an optional Bus Guardian (BG), and one or two Bus Drivers (BD). The communication controller is referred to as a FlexRay controller. The bus driver is referred to as a FlexRay transceiver. The FlexRay transceiver couples the FlexRay communication controller to the physical transmission medium and its primary task is signal transformation. On the one hand, it transforms the logical signal stream received by the FlexRay communication controller into a physical signal stream. In the other direction, it transforms the received physical signal stream into a logical signal stream.

In this paper the FlexRay protocol is studied in detail and the performance of clock synchronization module in the normal active state of POC module is verified by implementing it in the Altera Stratix IV FPGA.

II. FLEXRAY COMMUNICATION CONTROLLER

The FlexRay communication controller executes the communication protocol defined in the FlexRay specification. The primary tasks of the FlexRay communication controller include framing, bus access, error detection and handling, synchronization, putting the FlexRay bus to sleep and waking it up, as well as coding messages which has to be transmitted and decoding messages which are received. The FlexRay communication controller in a FlexRay node consists of six modules as shown in the figure 1. They are controller host interface (CHI), protocol operation control (POC), coding and decoding (CODEC), media access control (MAC), frame and symbol processing (FSP), and clock synchronization process (CSP).

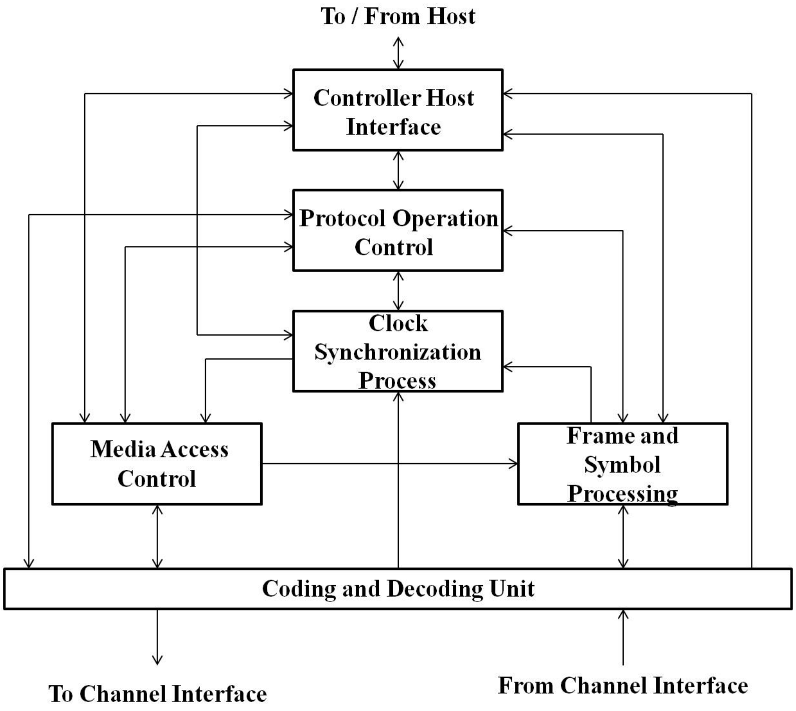

To / From Host
Controller Host Interface
Protocol Operation Control
Clock Synchronization Process
Media Access Control
Frame and Symbol Processing
Coding and Decoding Unit
To Channel Interface
From Channel Interface

Fig-1 FlexRay Communication Controller Architecture

The CHI module acts as a mediator between the host controller and the FlexRay communication controller within each node. The POC module controls the operational modes of the FlexRay modules. The CODEC module is responsible for encoding the communication elements into the form of a bit stream and for receiving communication elements, making bit streams and checking the correctness of the bit streams. The MAC module controls the access of FlexRay node to the bus. The FSP module is responsible for checking the correct timing of received frames and symbol, applying further syntactical tests to received frames, and checking the semantic correctness of received frames. The CSP module is responsible for generating timing units in the FlexRay communication controller, e.g., communication cycles. Moreover this module uses a distributed clock synchronization mechanism in which each node individually synchronizes itself to its cluster by observing the timing of transmitted frames from other nodes.

III. CLOCK SYNCHRONIZATION MODULE

Synchronization is the process of making different actions in a communication network with distributed clock system to agree on a same time reading. In FlexRay protocol each ECU’s are having their own local clocks whose values deviates from each other as time passes due to problems like voltage and temperature variations, even though they are initially synchronized with each other.

The basic time unit generated by the crystal oscillator in each ECU is known as Microtick. The durations of microticks are not constant, and to solve this problem a single, unique concept of time which is simplified to Global time is required. The Global time is expressed in terms of Macroticks. A fixed number of Macroticks are then combined together to form the communication cycles. Possible deviations in a clock take place in rate and offset parameters. These deviations are detected and corrected with the help of an algorithm named as Fault Tolerant Midpoint (FTM) algorithm.

IV. NORMAL ACTIVE STATE IN POC MODULE

POC is responsible for changing the mode of the core mechanisms of the protocol in response to changing conditions in the node. Figure 2 depicts an overview of the communication controller POC operations. Among the several states of POC module the Default config state is the startup state of communication controller. Config state sets up all the necessary conditions for the protocol's proper functioning. Ready state indicates it is ready for communication. If the node is in sleep mode Wakeup state can be used to make it active. During Startup state it initiates the communication process.

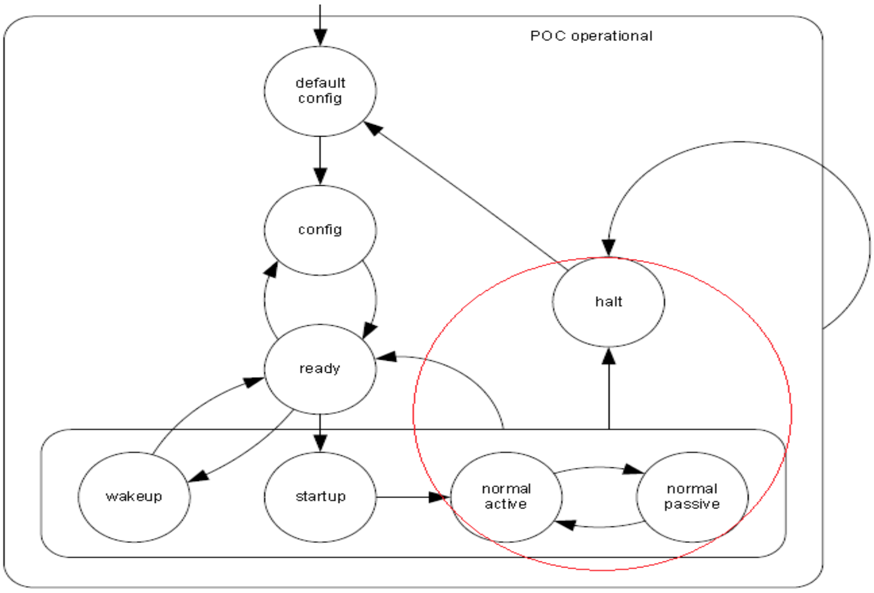


Fig-2 State diagram of FlexRay Protocol Operation Control

In Normal Active state the POC performs Synchronization calculations at the end of each cycle to determine whether the POC should change the modeling of the core mechanisms before the beginning of the next communication cycle. The state changes occur in accordance with the sync calculation results received during this state. If the sync calculation results are within bound and there are no errors the POC will remain in the active state itself. Else it will transfer to Passive or HALT conditions according to the error.

V. FAULT TOLERANT MIDPOINT (FTM) ALGORITHM

The functional principle of the Fault Tolerant Midpoint (FTM) algorithm is as follows.

i) Once the measurements are taken and the divergence values are calculated for both rate and phase differences, then these values are arranged in descending order.

ii) Then the most extreme values 'k' values from these measured divergences are removed. The algorithm determines the value of a parameter 'k', based on the details given in table I.

iii) The remaining one value from both the lists is then considered as the final rate correction value and final phase correction value.

TABLE I FTM ALGORITHM 'K' VALUE.

No of sync nodes	Parameter K
1 to 2	0
3 to 7	1
Greater than 7	2

VI. DESIGN APPROACH

A cluster consisting of three FlexRay nodes is considered for this paper. Each node is assumed to have a local clock and the frequencies of these clocks are 80 MHz, 100 MHz and 60 MHz respectively. The Macrotick deviation in each clock after some time is shown in Table II.

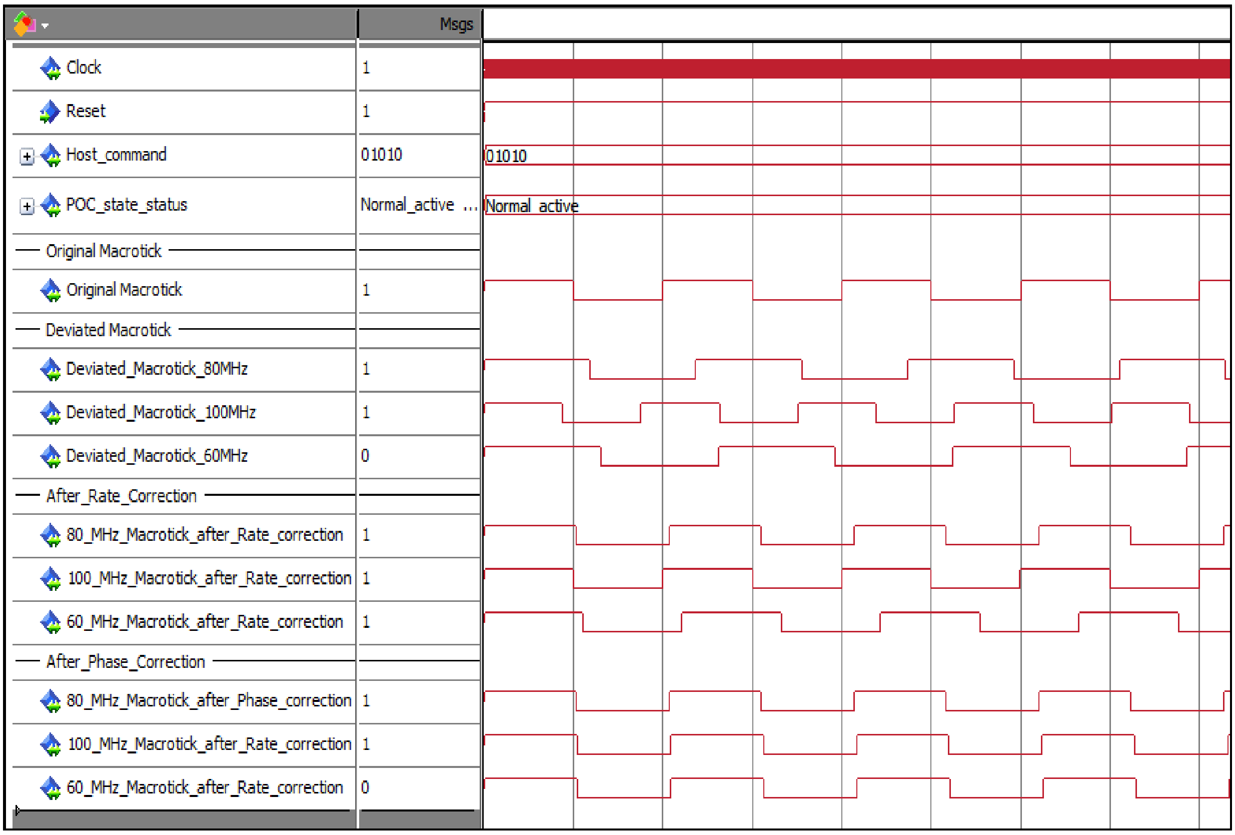
TABLE II MACROTICK DURATION AFTER DEVIATION.

Parameters	Values	Units
Macrotick duration (Required)	1	MHz
Duration of Macrotick generated by 80 MHz clock after deviation	844.3	KHz
Duration of Macrotick generated by 100 MHz clock after deviation	1.14	MHz
Duration of Macrotick generated by 60 MHz clock after deviation	762.13	KHz

The values of rate deviations are calculated by comparing the microtick counter values of the clocks before and after the possible deviations. The values of offset deviations are calculated by using a phase frequency detector. A phase frequency detector is a logic circuit that generates a voltage signal which represents the difference in phase between two signal inputs.

After obtaining the phase and rate deviations they are arranged in descending order to perform the FTM algorithm. Since, the cluster under consideration contains only three nodes the value of k will be 1. Therefore one extreme value from the greatest and smallest measured values of the list is removed. Then the remaining one value is taken as the final offset and rate correction value. As per this value, that much number of Microticks are added or subtracted from each deviated Macrotick to make them synchronous with other Macroticks.

The hardware description language used for developing the design was VHSIC Hardware Description Language (VHDL). Mentor Graphic's Modelsim was used for simulating the VHDL design. The simulation waveform of Macroticks before and after rate and phase correction is shown in figure 3.



	Msgs
Clock	1
Reset	1
Host_command	01010
POC_state_status	Normal_active ...
Original Macrotick	
Original Macrotick	1
Deviated Macrotick	
Deviated_Macrotick_80MHz	1
Deviated_Macrotick_100MHz	1
Deviated_Macrotick_60MHz	0
After_Rate_Correction	
80_MHz_Macrotick_after_Rate_correction	1
100_MHz_Macrotick_after_Rate_correction	1
60_MHz_Macrotick_after_Rate_correction	1
After_Phase_Correction	
80_MHz_Macrotick_after_Phase_correction	1
100_MHz_Macrotick_after_Rate_correction	1
60_MHz_Macrotick_after_Rate_correction	0

Fig-3 Waveform showing Macroticks before and after phase and rate correction.

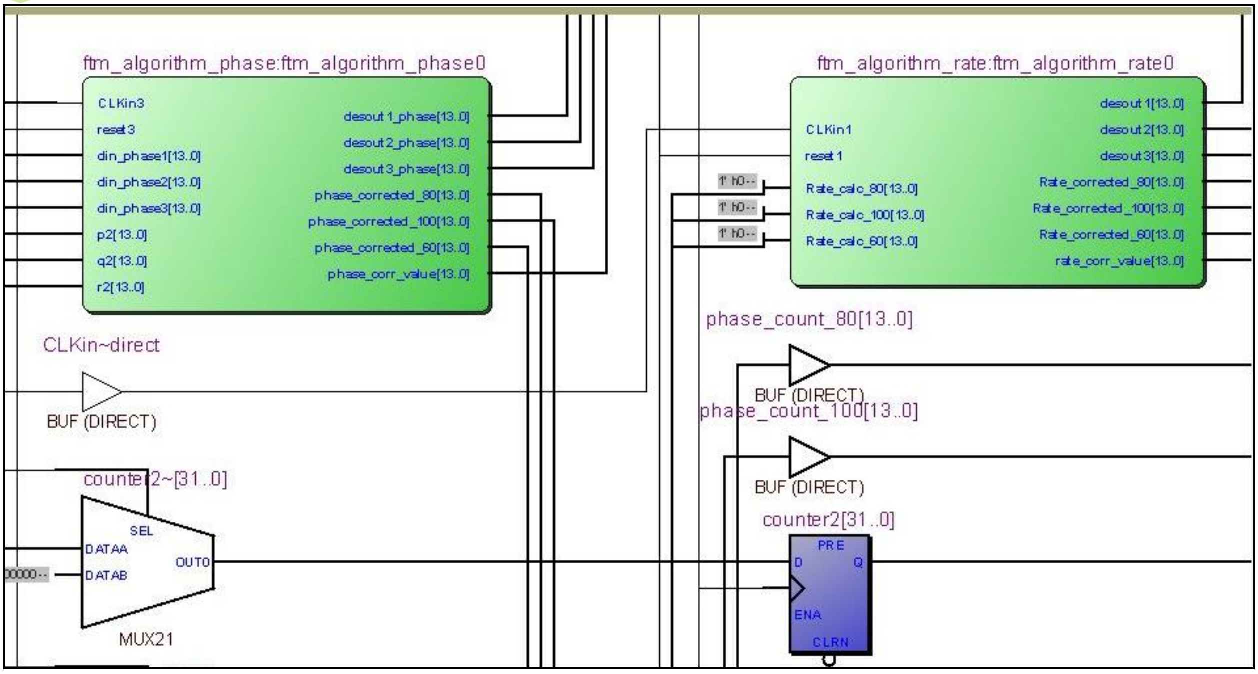


Fig-4 RTL View of FlexRay Clock Synchronization Module.

The RTL synthesis view of FlexRay clock synchronization module is shown in figure 4. It provides the schematic of the internal structure of a design netlist. Altera Quartus II 13.0.0.156 design software is used for synthesis of the Flex Ray POC and Clock Synchronization modules. The Stratix IV GX FPGA development board provides a hardware platform for developing and prototyping this design.

VII. CONCLUSIONS

The aim of this paper was to understand the concept and to provide a little contribution towards the discussion on the FlexRay Protocol operation and Clock synchronization process within a cluster. The POC module uses (Finite State Machine) FSM methodology to transfer its control from one state to another. It reacts to the host commands and protocol conditions by triggering coherent changes to core mechanisms in a synchronous manner, and provides the host with the appropriate status regarding these changes.

The simulation model for clock synchronization aims at the assessment of FlexRay clock synchronization algorithm (FTM) performance, by means of simulation experiments. The FlexRay clock synchronization algorithm is very well suited to solve the clock synchronization problem in the presence of changing clock drift rates with regard to the achievable precision within a single cluster.

REFERENCES

[1] Milind Khanapurkar, Jayant Y. Hande and Dr. Preeti Bajaj, “Approach for VHDL and FPGA Implementation of Communication Controller of FlexRay Controller”, IEEE, 2009.

[2] Alexander Hanzlik, “A case study done on the topic clock synchronization in FlexRay”, 2006.

[3] Chia-Yu Lu and Chang-Ming Wu, "A Hardware Design Approach of Sorting for FlexRay-based Clock Synchronization", IEEE, 2011.

[4] FlexRay Communications System Protocol Specification Version 3.0.1

[5] FlexRay Communications System Electrical Physical Layer Application Notes, Version 2.1 Rev. B, 2006

[6] Dominique Paret, “FlexRay and its Applications: Real Time Multiplexed Network”.

[7] Robert Shaw, Brendan Jackman, “Introduction to FlexRay as an Industrial Network ”, IEEE International Symposium on Industrial Electronics, 2008.

[8] Rodrigo Lange,” Hybrid FlexRay/CAN Automotive Networks”, Federal University of Santa Catarina, Brazil.

[9] C. Talbot and Shang ping Ren,” Comparison of Field Bus Systems, CAN, TTCAN, FlexRay and LIN in Passenger Vehicles”, Illinois Institute of Technology Chicago.

[10] Ugur Keskin,” In-vehicle communication networks a literature survey”, 2009.