ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Design of Frame Synchroinization in FPGA

Bhumi Reddy Maha Lakshmi
M.Tech (VLSI Design),
Sri Ventakeshwara College of Engineering, Chitoor.

Y.Murali Mohan Babu, PhD, M.Tech
Professor,
Sri Ventakeshwara College of Engineering, Chitoor.

ABSTRACT:

Data from spacecraft or aircraft or un-manned machine are transmitted to stations on earth in serial and encoded form.The transmission channels are generally noisy and the encoding enables to correct errorsduring transmission. Lossless real time data acquisition of such spacecraft data is very crucial for providing data to meet the user need. Frame Synchronization is one method to ensure that the data received at ground is lossless. In serial frame synchronization that is when each frame starts with an identical sync code, false sync due to replicas of the code randomly generated by the data is completely eliminated by appropriate frame synchronization logic. In this paper a method of detection of the frame synchronization code from the received PCM data, a synchronization confirmation as a guard by detecting a repetition of the frame synchronization codes, a confirmation of the subsequent frame identification signal based on the synchronization confirmation, and separation of the received multiplex data into respective fields is presented. After frame synchronization, individual measurands are identified according to the frame location.

KEY ELEMENTS: Frame synchronization, FPGA
SOFTWARE: XILINX
LANGUAGE: Verilog HDL

1. INTRODUCTION:

Frame synchronization is an essential element in digital communication systems. Itdetermines the boundary between data frames so that the information can be recoveredcorrectly from a stream of data. In modern communication systems data is not transferred as asimple stream of bits or bytes but in terms of frames or packets. In time multiplexed pulsecode modulated (PCM) telemetry binary data signals from several sources are grouped intoframes which have to be identified at the receiver in order to de-multiplex the data.

Framesynchronization is obtained by inserting in series (e.g. at the beginning of each frame), or inparallel (i.e. on a separate sync channel), a frame sync code (FSC). At the receiver, the framesynchronizer correlates the received signal with its own replica of the FSC for different bitshifts, until synchronization is acquired. The synchronization is then maintained by verifyingthe repetition of this code at each frame provided the frame length is fixed.The frame synchronizer correlates the received FSC with the referencesynchronization code. The main problem of false sync due to replicas or almost replicas ofthe FSC generated by the random data is avoided in this type of serial frame synchronization. Since the frame length is fixed the probability of false sync can be reduced byverifying the occurrences of the FSC on successive frames.

In serial frame synchronization aportion of each frame (e.g. the first k bits of the frame) consists of a code sequence, repeatedat each frame. After frame sync detection takes place and synchronization, individualmeasurands are identified according to the frame location. The decommutator identifies andextracts embedded asynchronous data stream (EADS) words. Thus frame synchronizer is avery crucial subsystem in the satellite data acquisition unit of satellite ground station. In thispaper two channel frame synchronization logic is designed and implemented on a StratixFPGA.

The hardware design consists of two major modules FrameSynchronization logic which in turn has modules like Data Simulator, Flywheel & FrameSync Strategy and Bit slip & correction logics. This design is realized in Verilog and the software used is Xilinx. Decommutator will identify and separate theindividual parameters from the incoming satellite PCM stream after frame synchronization.The validation of the modules is done with an inbuilt data simulator. The frame sync codesare selectable for catering to different satellites The FPGA used to implement the design isSPARTAN-II. After successful total compilation the program output file isloaded into the FPGA using a JTAG connector.

2. XILINX ISE SOFTWARE AND VERILOG HARDWARE DESCRIPTION LANGUAGE:

Xilinx ISE(Integrated Synthesis Equipment)is a software tool produced by Xilinx for synthesis and Analysis of HDL designs, enabling the developer to synthesize (compile) their designs, perform timing analysis, examine RTL diagrams, simulate design's reaction to different stimuli and configure the target device with the programmer.

The Verilog HDL was developed in 1984 by Gateway Design Automation under the leadership of Philip Moorby. Itquickly became a defacto standard in industry, but was limited by its status as a proprietary language. Gateway was later acquired by Cadence Design systems, Inc., which placed the language in the public domain under the auspices of open Verilog a standard hardware description language(IEEE standard 1364-1995)since then, the EDA industry has developed a wide variety of tools supporting verilog -based designs.

3.SPARTAN-3FPGA:

Spartan-3 architecture has the following components CLBs (Configurable Logic Blocks), IOBs (Input Output Block), BRAM (Block RAM), Multipliers, DCM (Digital Clock Manager).CLB has RAM-BASED LUTs for logic and also has storage elements configured as flip flops or latches. Each CLB has 4 slices.

A single slice has 2 LUTs, 2 storage elements, 2 multiplexers, carry logic and arithmetic gates. IOBs for input and output data. Block RAM IS 18K-bit dual port blocks used for data storage. Multiplier has2 18-bit binary inputs and has 36-bit output.DCM is used for distribute, multiply, divide or shift clock.

4. LOGIC IMPLEMENTED IN THE FPGA:

The total design comprises of Data Simulator logic, Frame Synchronization andassociated logic, are realized in the Stratix FPGA as shown in Fig1.

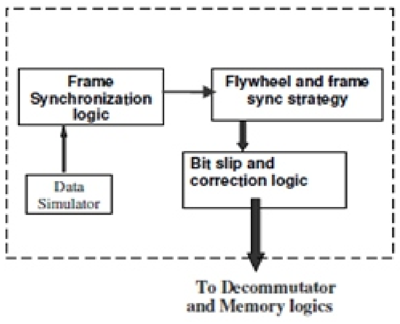
Frame Synchronization logic
Flywheel and frame sync strategy
Bit slip and correction logic
Data Simulator
To Decommutator and Memory logics

Fig1:Block diagram implemented in FPGA

5. DATA SIMULATOR LOGIC

The Data Simulator is to simulate the data. Data Simulator logic generates the FScode, with variable line count in the aux field and fixed video data pattern in a single-frame and for two channels . The required frequency is derived from a crystal oscillator.The twin channel's serial data and clock are connected to a RJ45 connector. Cat 6 cable isused to connect the simulator outputs to the Frame synchronization logic inputs.

6. FRAME SYNCHRONIZATION LOGIC:

Time Division Multiplexed (TDM) data is decommutated by first locating a fixedpattern and then determining that the pattern repeats at fixed intervals. The pattern is recognized by a correlator logic as shown in fig 3. that receives up to n sequential bitsand compares them to a programmed or fixed reference pattern, using a programmablemask to exclude don't care bit positions. Because telemetry data is often transmitted orstored imperfectly due to system noise constraints, absolute correlation is not alwayspossible. When determining the sync pattern location, we must often allow aprogrammable number of conflicts to occur in an otherwise acceptable pattern.

Thisnumber is referred to as the "sync pattern tolerance". In our design the tolerance bits arefrom zero to seven bits. The n bit correlation function is realized in the FPGA..Thiscorrelation logic compares the digital pattern stored in a reference pattern register withthe input data samples stored in the data input register. The number of matches iscalculated for every clock cycle. This number is called the correlation sum and is compared with a threshold value.

When the correlation score is >= the Threshold a FrameSync Detect pulse is generated. To prevent false detects a flywheel logic is included withstrategy which has a search, check and lock modes. When two consecutive syncs aredetected the logic will change from search to check and later to lock mode. Like wisewhen a sync loss occurs the logic will change from lock to check and when twoconsecutive sync loss occur the logic will revert to search mode.In our design the reference pattern register, mask register, correlation summationlogic, threshold logic are realized using macrocells. The LUTs in the macrocell can beupdated rapidly while the FPGA is in full operation. The design has been done for aframe sync pattern of 128 bit.

Eg: For a 128 bit frame sync code, the correlator compares all the 128bits of the inputsequence with the 128bit stored in the reference data register. On each clock pulse, theincoming bit is clocked into the left-most register and all bits are shifted by one register. Comparison is made bit by bit between the data input register and reference data registerand the correlation sum is computed. A perfect match will give 1111 output at thecorrelation output. While in case of 1 bit error the correlation sum will be 1110 and itgoes in this way until the tolerable limit of 0000 (i.e. 15 errors are tolerable for 127 bitframe sync code).If the number of errors are more than 15, than a loss pulse is generatedto indicate that the frame sync code does not match with the reference pattern. Ifsuccessive loss pulses are detected than the logic will go to loss mode.

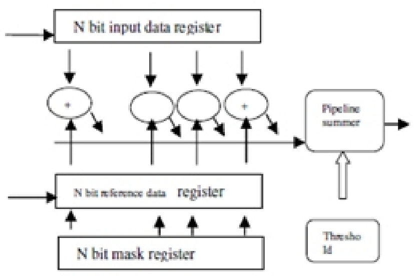

Fig2:Correlation principle diagram

7. FLY WHEEL AND FRAME SYNC STRATEGY:

The flywheel logic provides reliable frame synchronization and data decommutationby using programmable frame sync strategy counters, a programmable bit-slip window, andprogrammable bit error tolerances.

Frame synchronization occurs during all three states ofSEARCH, VERIFY, and LOCK. The design incorporated for our application is acombination of Fixed, and Adaptive, strategies as shown in Fig 4. The Fixed strategy tests fora programmable number of good or bad sync patterns to determine when state changes should occur.

The adaptive strategy is for use with noisy data when the sync pattern error toleranceis enabled.The fixed strategy tests the number of errors in the sync pattern and uses this value, relativeto previous values, to qualify a sync pattern. Four states have been identified as VERIFY,SEARCH, LOCK and CHECK for the frame sync logic to pass through. Initially the framesync logic will be in check mode.

In case 1 program for the sync strategy to transfer fromSEARCH to VERIFY, or in case 2 directly to LOCK. When the transfer is to VERIFY, youcan program additional flywheel frames (no acceptable sync pattern found in frame) before the strategy returns to SEARCH, and the number of consecutive good frames before thesynchronizer will go into LOCK.

This adds confidence that the real sync pattern has beenfound. In LOCK, you can pro- gram the number of consecutive bad frames before the synchronizer leaves LOCK. The transfer from LOCK is directly to SEARCH, thus allowing rapid sync reacquisition.

The Fixed strategy starts in SEARCH, where the bit stream is scanned for the programmedsync pattern. Upon detecting a good pattern (as qualified by the sync error tolerance) thesynchronizer enters

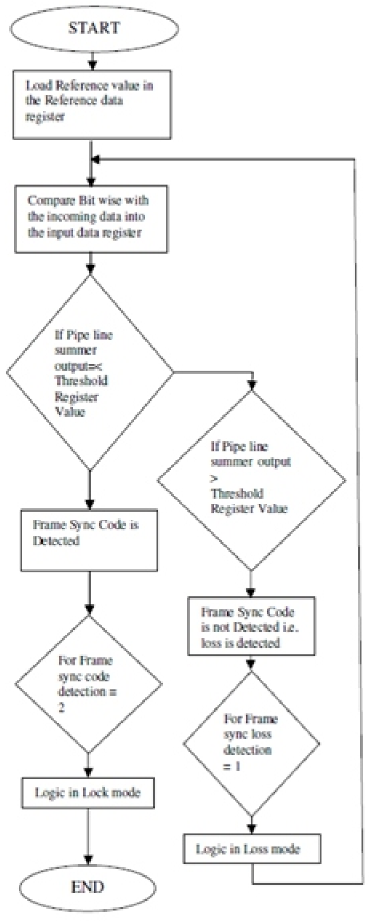
START
Load Reference value in the Reference data register
Compare Bit wise with the incoming data into the input data register
If Pipe line summer output=< Threshold Register Value
If Pipe line summer output > Threshold Register Value
Frame Sync Code is Detected
Frame Sync Code is not Detected i.e. loss is detected
For Frame sync code detection = 2
For Frame sync loss detection = 1
Logic in Lock mode
Logic in Loss mode
END

Fig3:Flow chart for the frame synchronization logic

the VERIFY state. A window is generated at the end of the frame by thewords per frame count and the bit slip window. If a good pattern is found within this windowthe consecutive Good Frames from VERIFY to LOCK count (2) is tested and, if equal, thesynchronizer will enter LOCK. The count of zero allows the strategy to transfer directly fromSEARCH to LOCK, bypassing the VERIFY state. When a good pattern is not found withinthe window, the consecutive Bad Frames from VERIFY to SEARCH (1) is tested and, ifequal, the synchronizer will return to SEARCH. In LOCK the correlator pattern is tested inthe End Of Frame window. If the number of consecutive frames with bad patterns matchesthe LOCK to SEARCH count (1), the synchronizer will return to SEARCH as shown in Fig .The flow chart shown in Fig3 Indicates the sequence followed in detecting the frame synchand also in case the frame sync pattern is not detected, it goes into loss mode. The Framesynchronization States are presented below in Fig4 .

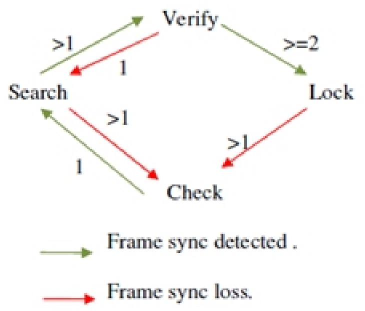

Fig4:Frame synchronization states

The flywheel and frame sync strategy allows us to record all the data received from thesatellite irrespective of the frame sync state into the system hard disk for further processing.The states will also ensure that the data recorded in the storage may still contain valid dataeven if the frame sync is not detected since the flywheel logic is in place and insynchronization with the frame window. Thus the imagery data may be good even in case theFrame sync data does not match with the reference pattern.

This states and strategy will avoidtotal frame loss thus improving the efficiency of data reception and enable post processing ofdata after real time data reception from the satellite. Thus the error tolerance will provideflexibility to the frame sync detection by enabling the allowable number of bit errors in theframe sync code and still acquire the data from the space craft.

So to maximize theprobability of synchronization lock with a particular acceptable errors and minimizing thelock probability if the data has got more errors will be the best suitable method for real timedata acquisition from satellites.The adaptive strategy functions as follows. In SEARCH, theframe sync logic will begin by searching for the frame sync pattern that meets theprogrammed frame sync error tolerance.

When a matched pattern occurs, the number ofdetected errors in the frame sync pattern will replace the contents of the sync error toleranceregister. The strategy sequencer remains in SEARCH as established by the words per framecount. The input stream is tested continually for a pattern with fewer errors than those sted,and if a sync pattern with less errors is found before end of frame, the words per framecounter is reinitiated and the new pattern error count replaces the sync pattern errortolerance.

When a frame passes that does not contain a pattern that is better than the currentsync pattern tolerance, and the pattern at the end of frame is acceptable, the frame sync unitwill advance to VERIFY. The strategy now works as in fixed mode, with the error toleranceequal to the number of errors detected in the best pattern encountered in SEARCH. If therequired number of good patterns is found in the frame sync window, the frame sync unit willadvance to LOCK.

If the frame sync errors are greater than the tolerance established (when leaving SEARCH), for the consecutive number of frames specified by the VERIFY toSEARCH count (when in VERIFY), or the LOCK to SEARCH count (when in LOCK), theframe sync unit reverts back to SEARCH mode. When SEARCH mode is re-entered, the syncerror tolerance is set to the initially programmed value and the pattern search proceeds aspreviously described.

8. BIT SLIP AND CORRECTION:

Frame synchronization normally consists of first looking for a pattern anywhere in thestream while in SEARCH, and then applying a frame length window to avoid false sync patternsthat may occur in the data stream. The frame is normally of fixed length and the pattern shouldrecur at a specific bit interval. However, when the input to the bit synchronizer lacks sufficienttransitions, and/or contains excessive noise, the bit synchronizer clock may drift and produceexcessive or insufficient clock pulses during the frame. This results in a Frame Bit Slipcondition,which can result in the loss of synchronization.

Bit Slip Correction permits to program the frame synchronizer to accept sync patterns occurringin bit positions adjacent to the expected position in the frame. The sync pattern may occur exactlyat the expected location (one bit window); one bit position early or late (three bit window), twobit positions prior or after the expected position (five bit window), or three bit positions prior orafter the expected position (seven bit window), and still be detected as an acceptable sync pattern.This feature enables the unit to maintain synchronization during excessive noise bursts or datadropouts in the input stream when the bit synchronizer cannot maintain synchronization with thePCM stream.

In our design the Bit Slip Window of 3 bits is designed (i.e. +/- 1 bit slip) may correct forthe resultant one bit sync error. However, if the sync pattern is not properly chosen (when thetelemetry encoding was designed), the pattern, shifted by several bit positions with random data,may look like an acceptable pattern (especially when the sync pattern error tolerance is large) andthe results may resynchronize the frame erroneously. Therefore, the frame synchronizers mustprovide a programmable sync strategy that will achieve rapid synchronization while, at the sametime, guarding against false sync patterns produced by the variable patterns encountered in thePCM stream.

RESULT:

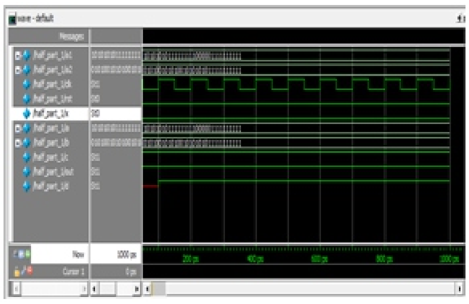

Fig5:Simulation results

Frame synchronization in FPGA, the device utilization is 67 slice flip-flops, 17 4-input LUT's are used.The errors removed from The received data using frame synchronization. Finally the extracted data is stored in BLOCK-RAM according to the frame location depending upon the FSC.

CONCLUSION:

The Frame Synchronization in FPGA and associated logic designed and developed is suitable for satellite data acquisition systems in the Ground segment. It is used for Monitoring environmental conditions, System and equipment status at remote locations. The errors removed from The received data using frame synchronization. Finally the extracted data is stored in BLOCK-RAM according to the frame location depending upon the FSC. The future of the Telemetry is Network enhanced Telemetry that means point-to point and network telemetry combined.

REFERENCES:

[1] Jesse L.Maury, Jr, Frederick J.Styles(1964) " Development of Optimum Frame Synchronization codes for Goddard Space Flight Center PCM Telemetry Standards.

[2] U.Timor, Communications Systems Research Section, " Frame Synchronization in Time-Multiplexed PCM Telemetry with variable Frame Length

[3] JinpengXie, Yingqiang Ding, Shouyi Yang, Lin Qi, School of Information Engineering,Zhengzhou University, Zhengzhou, China(2010) "FPGA Implementation of FrameSynchronization and Symbol Timing Synchronization based on OFDM System for IEEE802.11a".

[4]M.Shahshahaniand L.Swanson, Communication systems Research Section (1986) " ANew Method for Frame Synchronization ".

[5] Cyprian Sajabi, Spread Spectrum Special Research Report. FPGA Implementation of aCorrelator and PN code generator. EE 737 –SPREAD SPECTRUM SYSTEMS, May 23,2004.

[6]. Haykin, S p.4790480 Communications systems -4th Edition. 2001 John Wiley & Sons Inc.

[7] Dixon, R. C. P.6 Spread Spectrum systems with commercial applications 1994 JohnWiley & Sons Inc.

[8] Rappaprt, S.S., Grieco, D.M. Spread spectrum Signal Acquisition: Methods andTechnology. IEEE Communications Magazine June 1985. Vol 22, No.6

[9]. ChristpheCunat and Emmanuel Boutillon,(2007) Member IEEE. "Simplified Hard wareBit Correlator".

[10].Usman ALI, Michel KIEFFER and Pierre DUHAMEL." Sliding Trellis based FrameSynchronization. 978-1-61284-233-2/11, 2011 IEEE.

[11].F.J.Lopex –Martinez, M. Garcia-Abril. E.Martos-Naya and J.T.Entrambasaguas, Dep.Ingenieria de Communicaciones, Universidad de Malaga, Malaga, Spain Year 2007, IEEE." Hardware Implementation of Corrleation –Based Synchronization Algortithm for wirelessOFDM.

[12].ShenSanmin and Liu Wenyi, Department of Electronic Science and Technology, NorthUniversity of China. " A Pulse Code Modulation decoding Method by Self-synchronizingwith VHDL 2010 International Conference on Computer Application and system, Modeling(ICCASM 2010).

[13]. En Zhou, XiaolinHou, Jianping Chen, Zhan Zhang and Hidetoshi Kayama, InnovativeRadio Transmission Lab, DoCoMo Beijing Communication Laboratories Co., Ltd. China.Proceedings of APCC2008 "FPGA Implementation and Experimental Performances of anovel Timing Synchronization Method in MIMO-OFDM Systems".

[14]. Cyprian Sajabi May 23, 2004. Spread Spectrum Special Research Report on " FPGAIMPLEMENTATION OF A CORRELATOR AND APN CODE GENERATOR". EE737-SPREAD SPECTRUM SYSTEMS.

[15].NazilaSalimi, John Nielsen and Gerard Lachapelle, University of Calgary. " CDMA Correlator Implementation in FPGA".

[16]. William K. Pratt, Image Processing Institute, University of Southern California, LosAngeles. IEEE Transactions on Erospace and Electronic Systems, VOL. AES-10, No.3 May1974. "CORRELATION Techniques of Image Registration".

[17] M. A. Majed and Prof. C.S. Khandelwal, "Efficient Dynamic System Implementation Of FPGA Based Pid Control Algorithm For Temperature Control System" International Journalof Electrical Engineering & Technology (IJEET), Volume 3, Issue 2, 2012, pp. 306 - 312, Published by IAEME.