© 2019 IJSRST | Volume 6 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology
DOI : https://doi.org/10.32628/IJSRST1962160

Delay Reduction in SDR Communication System

Sagar D. Barde[1], P. J. Suryawanshi[2]
[1]M. Tech, Department of Electronics Engineering, PCE, Maharashtra, India
[2]Assistant Professor, Department of Electronics Engineering, PCE, Maharashtra, India

ABSTRACT

Data communication plays an important role nowadays, used in many applications like financial transactions, social interactions, education, national security and commerce. As there is increase in the growth in the mediums, people used to communicate i.e. data communications, voice communications video communications, etc. So, there is requirement for modification in the conventional radio devices. The Software defined radio is a technology in which software modules running on a generic hardware platform consisting of DSP's and general purpose microprocessors are used to implement radio functions. This paper presents two methods which increase the efficiency of SDR: The Standard beacon based SDR and parallel processing based SDR. The proposed system will reduce the delay of SDR using beacon based technique.

Keywords : SDR, Radio Communication, Beacon, Transmitter, Receiver, Pipelining, Parallel Efficiency, Delay.

I. INTRODUCTION

Radio is now a necessary part of everyday life from broadcast radio to mobile communications and much more.

Issues with conventional radios

- These radio systems were designed to communicate using one or two waveforms.
- Different set of hardware for different type of radio communication. Example mobile phone having wifi, Bluetooth and GSM antenna.

SDR

- Radio in which some or all of the physical layer functions are software defined.
- A radio in which the RF operating parameters including, but not limited to, frequency range, modulation type , or output power can be set or altered by using software.
- A technique in which all the processing includes mixing, filtering, demodulation, etc.
- Used to implement different demodulation scheme and different standards can be implemented in the same device.
- Can be updated so the device becomes obsolete with time.

The objective of this paper is

- To design a SDR
- To reduce delay of SDR

Using beacon based technique and using Pipelined and parallel processing based SDR.

II. PROPOSED DESIGN

This section focuses on the design of SDR transceiver using ModelSim SE PLUS 6.3f. Design includes transmitter block, channel sources, channel sinks, channel and receiver block. Simulation is carried out

in the Simulink environment of MATLAB and at last, the implementation and simulation of SDR is done in ModelSim SE PLUS 6.3f.

ModelSim is a multi-language HDL simulation environment of hardware description languages such as VHDL, Verilog. Simulation is performed using graphical user interface (GUI). For building system models for any specific processing operation, performing simulations, as well as analyzing results, graphical user interface (GUI) is generally used.

Fig 1 below shows the SDR based transmitter and receiver.

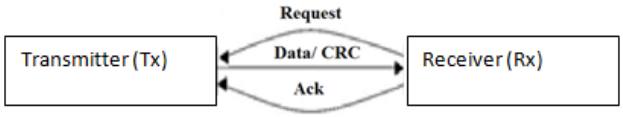


Figure 1 : Beacon based SDR

The request and acknowledgment policy is used by Beacon based SDR for transferring the data. As shown in the figure, there are two nodes namely transmitter and the receiver. The data acquired by transmitter is the data which the receiver requires. When the transmitter is free to accept the data, receiver will send a Beacon Packet to the transmitter. The Beacon Packet consists of the request from the receiver to transmit the data.

When the request reaches to the Transmitter, it will check the request first and the data required by the receiver. If all the constraints match, the transmitter will transmit the data. The error detection and removal technique is applied to the transmitted data with CRC so that at the receiver node if the data received is found to be faulty, CRC will reconstruct it. The Data and CRC are received at the receiver node and thus, from the newly received data, again CRC is calculated and it will be XORed with the received CRC and so it will be easy to detect the error bits. The error bits are then XORed with the received data and finally the original data can be reconstructed at the receiver node. When the original data is retrieved, the receiver will send an acknowledgment to the transmitter. The acknowledgment consists of the receiver's received (the desired data) and the transmitter will stop sending the data and will wait until the next Beacon request from the receiver.

The algorithm for the Beacon based SDR is as follows

Step 1: Checks if the transmitter is free to accept the data if yes then initialize the Beacon packet

Step 2: Send the Beacon packet to the transmitter,

Step 3: Calculate the CRC for the data.

Step 4: Send the transmitted data with CRC to the receiver.

Step 5: At the receiver node, calculate the CRC for the received data.

Step 6: XOR operation is done with the CRC received and CRC calculated.

Step 7: To get the Original Data, XOR the Error bits with the received data.

Step 8: Send acknowledgment to the transmitter.

Flowcharts for Beacon Based Transmitter and Receiver are as shown in figure 2 and figure 3

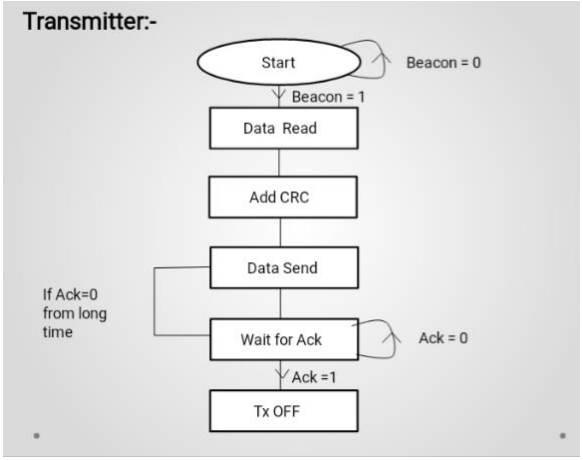


Figure 2 : Flowchart for transmitter used in Beacon based SDR

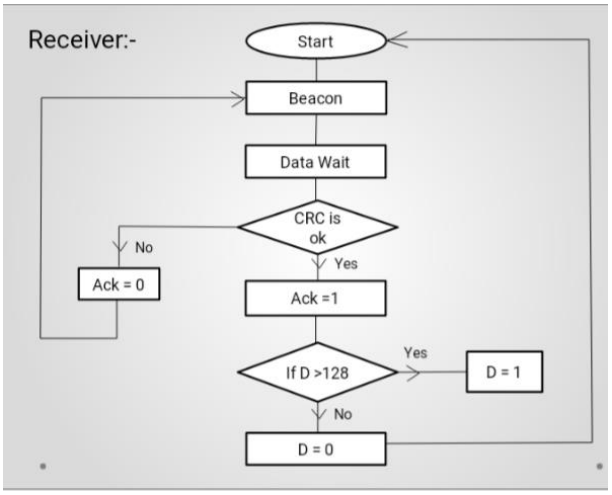


Figure 3 : Flowchart for transmitter used in Beacon based SDR

In Beacon based SDR , figure below shows the Main controller (Combination of Transmitter with Receiver) is designed to meet the objectives of this project i.e. to improve power efficiency and reduction in delay of SDR .

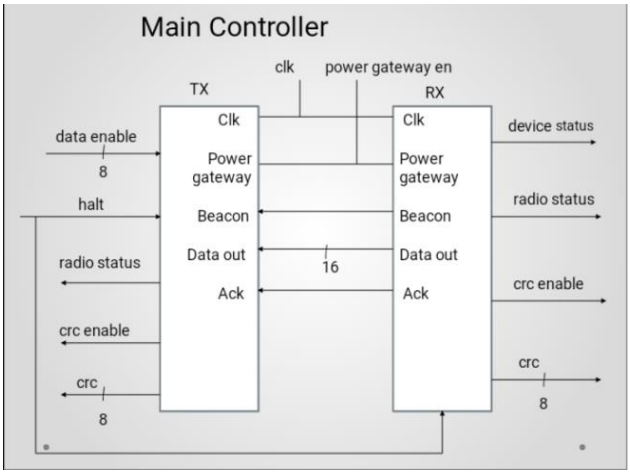


Figure 4 : Main Controller used in Beacon based SDR Pipelining and Parallel processing based SDR

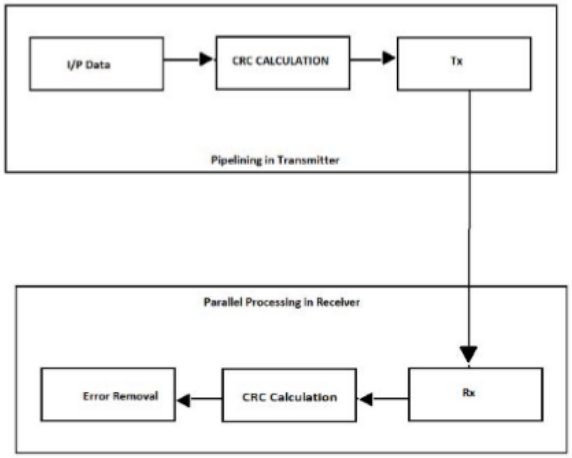


Figure 5 : Pipelined and Parallel process based SDR

In Pipelining and parallel processing based SDR, pipelining is applied in the transmitter side and parallel processing is in the receiver side.

Pipelining

This technique is generally used in advanced microprocessors where the microprocessor begins executing a second instruction before the first has been completed.

The pipelining stages are

- Instruction Fetch(IF)
- Instruction Decoder(ID)
- Execution
- Memory Access(MEM)
- Write back (ST)

Modules

The figure below shows the four pipelined instruction

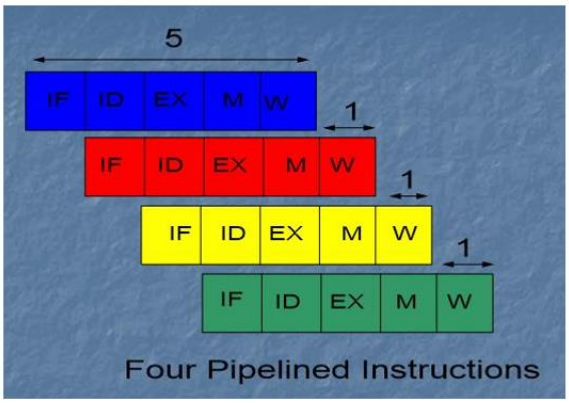
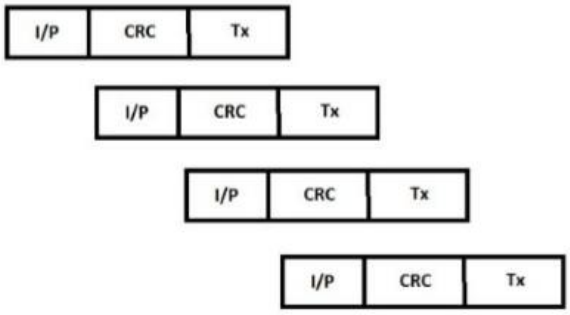


Figure 6 : Four Pipelined Instructions

Similar type of pipelining process used in the proposed system which is shown in the figure below



As shown in the above figure, at the transmitting side, the system has three operations the data input, calculating its CRC and transmitting the data. In first clock cycle, instruction pipeline reads the input data. In second clock cycle, an instruction pipeline reads the input data while CRC calculation is also being executed. Similarly, in the third clock cycle, transmitter receives the input, CRC calculation and also an instruction pipeline read the input data. Hence, with in the proposed method, with the help pipelining, multiple operations run without disturbing each other and also saves the time.

At the receiving side, parallel processing is used. In the old times, to run one operation requires one processor, which uses to create a long queue of processes to run, causing increase in the delay of the processes to run. To solve this problem, the Parallel Processing was introduced. In parallel processing, the processes are divided into multi-processes, where they can run individually without affecting the other operation.

There are three processes, also at the receiving side, which are executed parallel to each other. When the data reaches the receiver, the first step is to calculate the CRC and second step is the error removal. These two processes run, parallel to each other. So the time will be saved while processing these operations. Finally, the acknowledgement is sent to the transmitter.

In this process, a lot of time is saved without affecting the output but increasing the efficiency of the system.

III. PERFORMANCE PARAMETER

Arithmetic Logic Unit (ALU):

The ALU is a digital electronic circuit, responsible for performing all arithmetic and bitwise logical operations which takes place within the processor. These operations can have one operand or two, these values it takes either from the register file or from the immediate value from the instruction directly. All operations are done according to the control signal, coming from ALU control unit.

Cyclic Redundancy Check (CRC):

CRC is an error-detecting code which is commonly used in digital networks. CRCs are popular because they are simple to implement, easy to analyze and particularly good for detecting the common errors caused by the noise in the transmission channel.

Synthesis results of top module

For the purpose of implementation, the top module is the root of the design hierarchy. All the sub modules

are combined in the top-level module to form the final system and synthesized in ModelSim SE PLUS 6.3f. After synthesizing the design, the RTL schematic can be viewed as black box. It shows the inputs and outputs of the system as shown in figure 7 and figure 8 which shows the RTL Schematic of ALU and CRC respectively. On double-clicking the diagram of the RTL Schematics, Technology schematic can be viewed as shown in figure 9. It shows the Technology Schematic of Main controller.

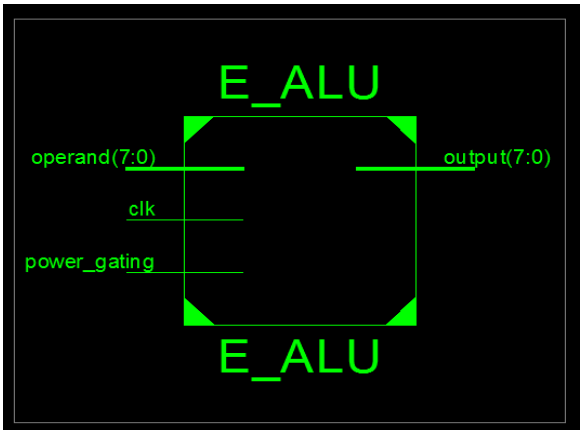


Figure 7 : RTL Schematic of ALU

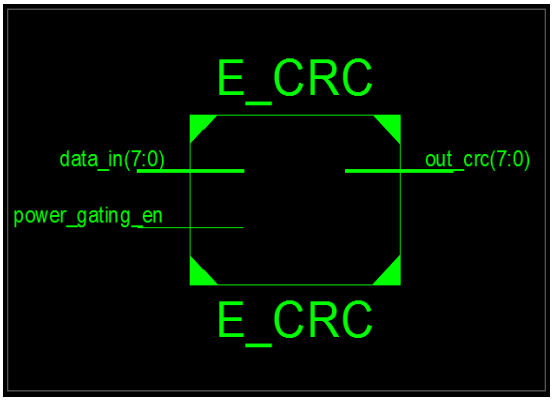


Figure 8 : RTL Schematic of CRC

IV. SIMULATION AND RESULTS

The simulation waveforms of Standard Beacon based SDR Simulation and Pipelined and Parallel processing based SDR Simulation are shown below. In the proposed work, both inputs and outputs will be binary streams of data as shown in figure.

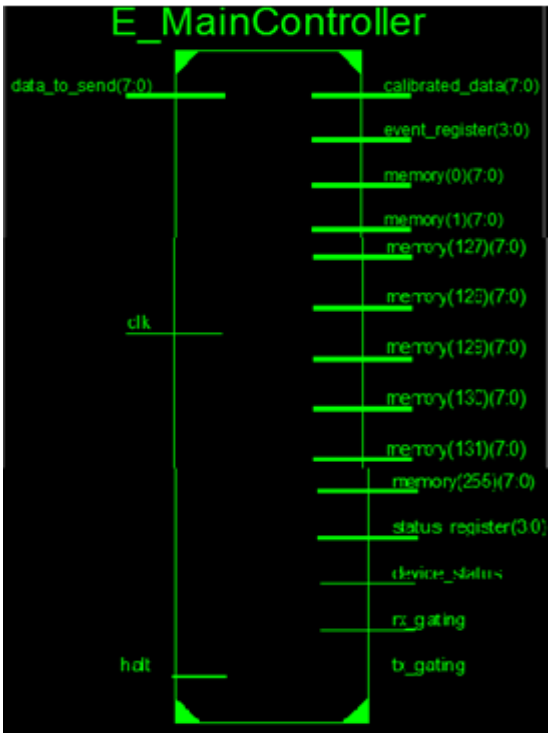
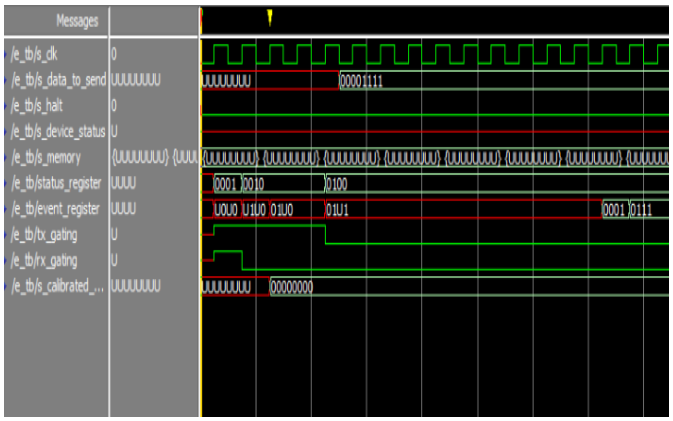
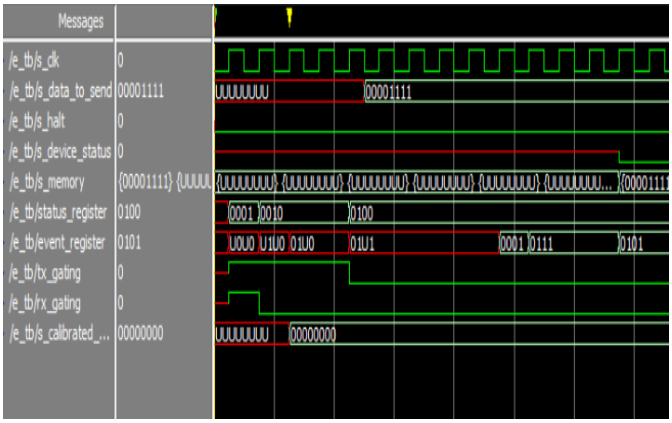


Figure 9 : Technology Schematic of Main controller

Standard Beacon based SDR Simulation



Pipelined and Parallel processing based SDR Simulation



V. OBSERVATIONS

Table 1 : Analysis

	Standard Beacon SDR	Pipelined and Parallel Processing SDR
Tx(Time)	500ns	500ns
Rx(Time)	1850ns	1350ns
Delay	1350ns	850ns

When the Standard Beacon based SDR is used, for transmitting the data, the time required is 500 nanoseconds and that for receiving the data is 1850 nanoseconds. So, the time required to process the data (delay) is 1350 nanoseconds. However when the pipelined and parallel processing SDR, for transmitting the data, the time required is 500 nanoseconds and that for receiving the data is 1350 nanoseconds. So, the time required to process the data (delay) is 850 nanoseconds.

So, when the pipeline and parallel processing SDR is used, it also transfers the same amount of data per unit time, the delay required is only 0.6 times the standard Beacon based SDR. The pipeline and parallel processing helps in saving the delay.

$$\textbf{Delay improvement} = \frac{1350 - 850}{1350} \times 100$$

Delay improvement = 37.07%

VI. CONCLUSION

The SDR is designed using beacon based technique and using Pipelined and parallel processing based SDR**.** As per the analysis taken, it can be concluded that the normal SDR which was studied in this paper takes N time to transfer the data. Whereas, it takes over a lesser amount of time in Pipelining and Parallel Processing SDR, making it the best technique.

VII. REFERENCES

[1]. Amiya Ranjan Panda, Debahuti Mishra, Hare Krishna Ratha, "FPGA Implementation of Software Defined Radio-Based Flight Termination System," February 2015.

[2]. T. Ulversoy, "Software defined radio: Challenges and opportunities," 2010.

[3]. R. Woods, J. McAllister, Y. Yi, and G. Light body, "FPGA-Based Implementation of Signal Processing Systems" Oct. 2008.

[4]. J. Meier, R. Kelley, B. M. Isom, M. Yeary, and R. D. Palmer, "Leveraging software-defined radio techniques in multichannel digital weather Radar receiver design," June 2012.

[5]. V. B. Alluri, J. R. Heath, and M. Lhamon, "new multichannel, coherent amplitude modulated, time-division multiplexed, software-defined radio receiver architecture, and field-programmable-gate-array technology implementation," Oct. 2010.

Cite this article as :
Sagar D. Barde, P. J. Suryawanshi , "Delay Reduction in SDR Communication System", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 6 Issue 2, pp. 761-766, March-April 2019. Available at doi : https://doi.org/10.32628/IJSRST1962160
Journal URL : http://ijsrst.com/IJSRST1962160