Baseband Data Handling System Using LEON3FT Processor

Arun. N
School of Electronics Engineering (SENSE)
VIT UNIVERSITY
Tamil Nadu, INDIA
arun.n@outlook.com

Dr. S Revathi
School of Electronics Engineering (SENSE)
VIT UNIVERSITY
Tamil Nadu, INDIA
srevathi@vit.ac.in

***Abstract*—** The data handling system is used to receive the data from the payloads of the satellite and format the data into suitable form so that it can be successfully received at the ground station. Data handling system consists of payload interface unit, preprocessor unit, data compression unit, data encryption unit, and channel coding and frame formatter. Till now, data handling systems were developed with the help of FPGAs. The current project involves the development of on board data handling system based on LEON3FT processor. Processor provides the advantages of reduced hardware complexity, programmability and computational performance.

Keywords- *leon3,data handling system, satellite, leon3 IP core, formatter*

*****

I. INTRODUCTION

Data Handling Systems are required to perform the capacity of preparing the Digital Video/Science information got from the optical/microwave/science payloads of IRS/SS satellites, utilizing a few uncommon strategies and advancements and configuration into the Digital Data Stream(s) suitable for transmission of the information from the satellite to ground at information rates going from many Kbps to several Gigabits for every second contingent on the payload and mission. Depending on the type of payload data types and requirement of the mission, a typical data handling system shall implement functions like data compression, Data Formatting, Data encryption and channel coding techniques.

Spacecraft-generated application data are formatted into end-to-end transportable data units and a transfer Frame is generated to reliably transport the data through the RF channel to the receiving ground station network. Transfer Frame begins with an attached frame synchronization marker followed by header information and payload data units. Formatted data is usually randomized except for the frame sync marker to ensure proper receiver operation and to maintain bit (or symbol) synchronization. On the receiving end, a frame synchronizer is used to detect beginning and end of the frame. Once the frame is identified based on the frame synchronization marker, header and payload data units are separated for further data processing.

At the ground formatted data is stored in the onboard in solid state recorders (SSR) for later playback. SSR plays a key role in remote sensing mission for the worldwide scope and is utilized for putting away the information as well as for executing rate supports when arranged information rate is higher/lower than the download transmission rate. BDH system receives data from payloads as well as from other subsystems on board spacecraft. Formatted/encrypted/compressed data is routed either to RF system for transmission to ground or to SSR for recording. Recorded data will be played back later during ground station visibility.

Major imaging payloads include PAN & multispectral Charge Coupled Device (CCD) cameras payloads which are usually used in different satellite missions. Normally the data rates of typical payloads are very high and are variable from few hundreds of megabits per second to Gigabits per second

II. DATA HANDLING SYSTEM : A SURVEY

The payload data from the different payload will enter into the data handling system through different payload interfaces such as low LVDS, ECL, TTL, LSTTL etc. Payload interface logic provides obligatory control signals to payloads, if required, and receives payload data. Data is transferred to BDH system with deference to the control signals or predefined protocols. Commonly used logic families for interfacing are TTL, LSTTL, LVDS, and ECL etc. Standard buses like 1553B, CAN, Space wire etc. are additionally utilized in categorical cases. Payloads have different interfaces and data rates.

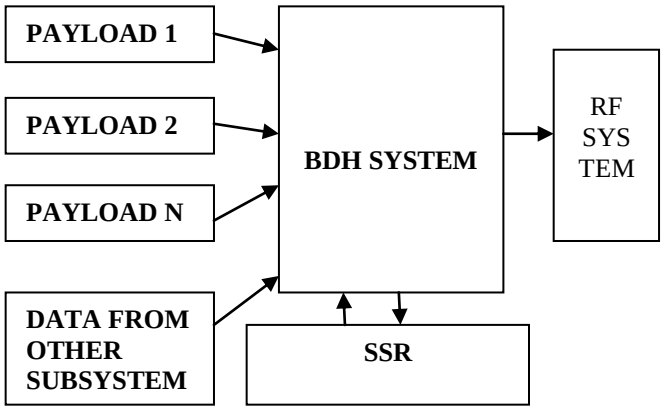

Figure 1. Block Diagram of typical data handling system

Pre-handling is done on the payload information to make the resulting processing (compression) more effective, and is extremely applicable on account of imaging payloads. Radiometric correction is carried out on the raw image data received from the electro-optical instrument to remove photo response non-uniformities in sensor output levels. There are many other types of onboard processing that could be applied in remote sensing to amend overall system efficiency. For example, it might be benign to analyze the images in order to detect the presence of clouds or other undesirable features. Information arrangement is consummated to make a physical picture as required by information compression calculation. In particular, cases preprocessing and arrangement may not be pertinent as on account of exploratory missions.

The main purport of compressing the data is to store as well as represent the data in fewer bits so that we can eschew the

redundancy of the data and make the system overall efficient. The compression is more germane when the abundance of data is to process and to be stored. The compression techniques can be mainly classified into two types, Lossy, and Lossless compression technique. In the lossless compression technique, information content is not lost but the compression ratio will be much lesser when compared to the Lossy compression technique.

One of the Encryption scheme utilized in the satellite is stream ciphering. A stream cipher engenders a sequence of bits called key stream. An LFSR (Linear Feedback Shift Register) can engender such stream. Encryption is accomplished by coalescing the key stream with payload data bit sagacious XOR operation. Now AES (Advanced Encryption Standards) is becoming more popular and is recommended by CCSDS.

Multiple individual payloads in the satellite system will be having different data rates and the packet length will be also different. So the necessity of making this asynchronous payload data into a synchronous data is by converting it into the standardized format. The asynchronous payload data is multiplexed together into a single synchronous data stream of fixed length coded frames for the reliable data reception by the ground base station.

Channel coding is a strategy by which information can be sent from a source to a destination by preparing information so that distinct messages are effortlessly recognizable from each other. This permits recreation of the information with low error probability

III. SYSTEM DESIGN

- In the design diagram, FPGA provides the interface for various payloads, stores the data into memory whenever necessary.
- FPGA interacts with LEON3 processor to receive the data from payloads that are interfaced with processor (CAN, SpaceWire).
- FPGA also interacts with processor during mathematical calculations, reception of commands, transmission of telemetry parameters etc.
- FPGA further processes all the payload data into desirable format and transmits it to the further subsystems

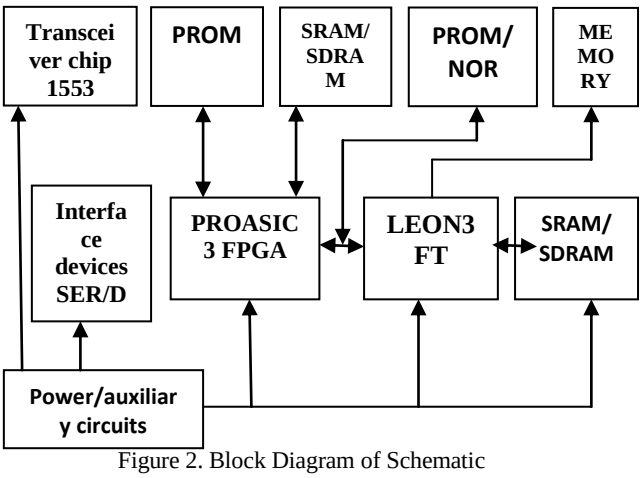

Figure 2. Block Diagram of Schematic

The heart of the system is the LEON3 processor which will be able to do more complex mathematical calculations so that overall hardware complexity of the system can be reduced. LEON3 processor will accept the payload data from the different interfaces such as CAN, SpaceWire and this data can be processed inside the processor or it can pass the data into the FPGA so that it can handle the data through some dedicated modules inside the FPGA which includes formatting, compression, channel coding or preprocessing such as Non Uniformity correction (NUC).

The FPGA also have its own interfaces such as Serializer and Deserializer, Transceiver 1553, LVDS which will also able to accept the payload data directly through the FPGA and will be able to process those data with the help of the LEON3 processor. The memory units connected to the FPGA are SRAM, SDRAM and programmable ROM.

A. SCHEMATIC DESIGN

The main interfaces provided for the LEON3 are Debug Support Unit (DSU), JTAG, CAN, and SpaceWire. The storage connected to the LEON3 are 2 Gb SDRAM. An LVTH is also connected to the processor so that to interface with the 5v devices.

Interface connected to the FPGA Prosaic3 are LVDS, LVTH and memory is connected as SRAM and NOR flash memory. EEPROM is connected to PROASIC3 and is accessed by the LEON3. SpaceWire control is also connected to the PROASIC. The FPGA will acts as a memory mapping devices for the LEON3.

The FPGA and Processor is connected through the general purpose input output device (GPIO). It will also act as a programmable interrupt. The signals shared between the two devices are Chip select signals, SRAM control signals, PROM and read write signals

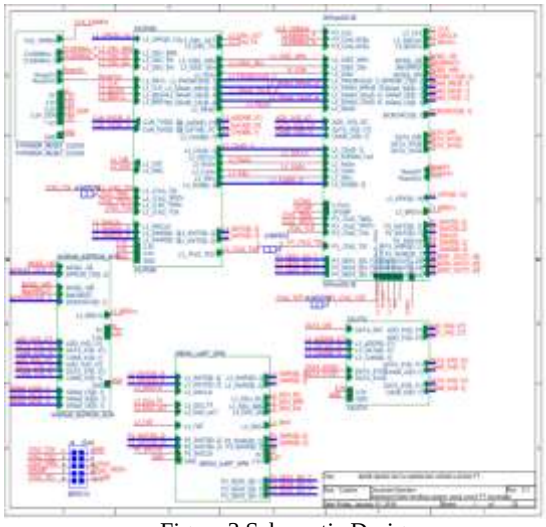
Figure 3 Schematic Design

B. COMPONENTS

1) Power, Reset & Clock

Clock signal is of 40 Mhz for the other interfacing devices the voltage generated by the system are 3.3v, 5v, 2.5v and 1.5v. 3v is also generated for the RESET purpose for the LEON3. However the input signal voltage given to the system is of ONLY 5v.

2) LEON3FT

The LEON3FT is fault-tolerant SPARC V8 32 bit 66 MHz processor. It is fabricated in monolithic technology. The processor is having pipelined design and hold the ability to

execute various high performance tasks.LEON3FT is mainly settled for space backgrounds. It is having the ability to correct the errors present in the on-chip RAM. The interface present in the LEON3FT includes SpaceWire, PCI, CAN bus, Ethernet, memory controller, programmable interrupt and contain an AMBA interface which is integrated to the core of LEON3FT
LEON3FT was fundamentally intended for the space exploration application and is funded by the European Space Agency (ESA). It is designed to have high performance with low complexity and with low power consumption.
The important features of LEON3FT are it is IEEE-1754 SPARC V8 compliant, 32-bit processor. It comes with a 5 stages pipeline. The processor also support the multiprocessing ability in which multiple core can used for processing. On-chip debugging ability with trace buffer support is also available for this processor.

3) LEON3 Core

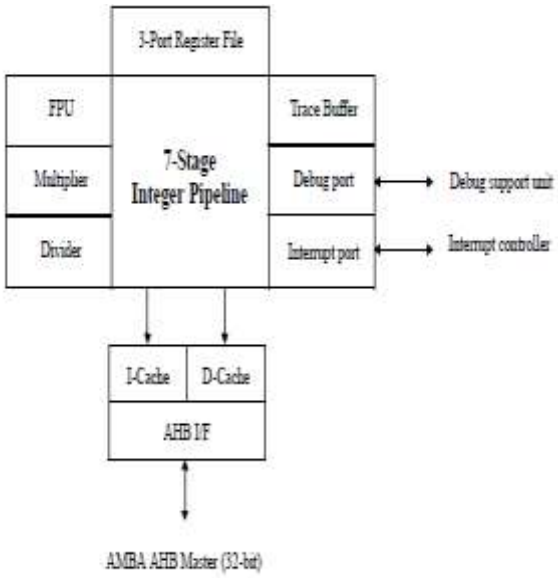

Figure 4. LEON 3FT Core

The main features of LEON3FT is the 7 stage pipelining with Harvard Architecture, which is having a distinct instruction and data cache, hardware divider and multiplier, floating point unit, hardware debug support.
The main features of the SPARC architecture is it will achieve a 60MIPS throughput when a clock frequency of 75 MHz is used. The size of the cache parity is about 4bit and the size of instruction cache memory and data cache is 32 byte and 16 byte respectively. The pipeline available is seven stage which will help in handling the multiple tasks simultaneously. The floating point unit will work as parallel operation.
The number of interrupts available are 15 hardware interrupts which has been generated by the general purpose input output (GPIO). This processor also support the SPARC trap models. The sensitivity of interrupt can be either level sensitive or edge sensitive. The interrupts are also programmable.
Debug Support Unit (DSU) will provide an platform for debugging without the disturbing the whole system. Debug unit can be connected with the interfaces such as PCI, SpaceWire, UART or through JTAG. DSU unit will access all the registers of the processor and cache memory.
4) ProASIC3E
ProASIC3E is the third-generation flash FPGA family from the Actel which deals performance, density, and features further than the ProASICPLUS family. The main advantage of this FPGA is it provide high security, low power and is Live at power-up (LPA). This also provides a 1 Kbit of on-chip, non-volatile, programmable Flash ROM memory.

IV. IP CORE INSTANTIATION

The GIRLAB IP core is a reusable set of integrated core. This is developed mainly for system on chip development. This core is dependent on vendor and will support multiple CAD tools and different mark technologies. The configuration will be using the paly and play method which result in unaltered global resources. The LEON3 processor is also a synthesizable VHDL models with the architecture of SPARC V8 32 bit.

A. Minimal System Requirement

IP cores required for the minimal LEON3 is
- CLKGEN: Clock Generator
- MEMCNTRL: Controller to access ROM and RAM
- RSTGEN: Reset Generator
- GPTIMER: General purpose Timer
- AHBCNTRL: Controller for AHB bus
- APBCNTRL: Controller for APB bus

B. Typical Memory Map

The most important memory map area are RAM and PROM area, the RAM area will starts from 0x40000000 and PROM area is from 0x00000000. The location of APB/AHB is little bit less important and the location is from 0x80000000

C. Installation

GRLIB is available as a zip file and can be installed in any location on the system.
gunzip the tar file using the gunzip command in bash shell
The distribution includes the following files in the hierarchy:
- Doc: will contain the documentation about the process
- lib: various VHDL libraries which is useful
- software: various software utilities available
- verification: contain the test benches
- board: files for different FPGA prototyping design boards
- designs: template designs for various board
- bin: contain different scripts and tool support files

D. Implementation

Implementation is typically done in three steps
1. CONFIGURATION USING XCONFIG
2. SIMULATION OF DESIGN
3. SYNTHESIS, PLACE&ROUTE

The template design is based on three VHDL files

1) config.vhd: This package contain design configuration parameters. This is automatically generated by using the xconfig GUI tools
2) leon3mp.vhd: Top level entities of the on chip IP-cores, It will also use the config.vhd files to instantiate the cores
3) testbench.vhd: test bench

E. Configuration

Using the command 'make xconfig' in bash shell (Unix machine) or Cygwin (windows machine). It will start the GUI for configuration, from that window the LEON3 is modified according to the design

F. Design flow

- Download LEON3FT source code from the web
- Modifying the design of the IP core and making it to compactable in the ACTEL ProAsic3 FPGA and also by adding design blocks for the target FPGA
- Simulate using a ModelSim simulator
- Synthesize using Synplify
- Place and route FPGA using vendor specific tools Actel Libero Designer
- Target design to one of many development boards

G. Instantiation Result

Result section will show the transcript window of the Instantiated LEON3 IP core using the Libero IDE 10.2. The Instantiated number of master and slaves can be seen in the output of the transcript. This IP care can be directly fused into the FPGA to make the Processor into an FPGA. From the output it is obvious that the Design contain 3 masters and 8 slaves. The masters are Leon3 SPARC v8 processor AHB UART debug UART and JTAG Debug link.

```
# LEON3 Actel PROASIC3-1000 Demonstration design
# GRLIB Version 1.0.20, build 3403
# Target technology: proasic3  ,  memory library: inferred
# ahbctrl: AHB arbiter/multiplexer rev 1
# ahbctrl: Common I/O area disabled
# ahbctrl: AHB masters: 3, AHB slaves: 8
# ahbctrl: Configuration area at 0xfffff000, 4 kbyte
# ahbctrl: mst0: Gaisler Research        Leon3 SPARC V8 Processor
# ahbctrl: mst1: Gaisler Research        AHB Debug UART
# ahbctrl: mst2: Gaisler Research        JTAG Debug Link
# ahbctrl: slv0: European Space Agency   Leon2 Memory Controller
# ahbctrl:       memory at 0x00000000, size 512 Mbyte, cacheable, prefetch
# ahbctrl:       memory at 0x20000000, size 512 Mbyte
# ahbctrl:       memory at 0x40000000, size 1024 Mbyte, cacheable, prefetch
# ahbctrl: slv1: Gaisler Research        AHB/APB Bridge
# ahbctrl:       memory at 0x80000000, size 1 Mbyte
# ahbctrl: slv2: Gaisler Research        Leon3 Debug Support Unit
# ahbctrl:       memory at 0x90000000, size 256 Mbyte
# ahbctrl: slv7: Gaisler Research        Single-port AHB SRAM module
# ahbctrl:       memory at 0xa0000000, size 1 Mbyte, cacheable, prefetch
# apbctrl: APB Bridge at 0x80000000 rev 1
# apbctrl: slv0: European Space Agency   Leon2 Memory Controller
# apbctrl:       I/O ports at 0x80000000, size 256 byte
# apbctrl: slv1: Gaisler Research        Generic UART
# apbctrl:       I/O ports at 0x80000100, size 256 byte
# apbctrl: slv2: Gaisler Research        Multi-processor Interrupt Ctrl.
# apbctrl:       I/O ports at 0x80000200, size 256 byte
# apbctrl: slv3: Gaisler Research        Modular Timer Unit
# apbctrl:       I/O ports at 0x80000300, size 256 byte
# apbctrl: slv7: Gaisler Research        AHB Debug UART
# apbctrl:       I/O ports at 0x80000700, size 256 byte
# apbctrl: slv11: Gaisler Research        General Purpose I/O port
# apbctrl:       I/O ports at 0x80000b00, size 256 byte
# ahbram7: AHB SRAM Module rev 1, 4 kbytes
# grgpio11: 7-bit GPIO Unit rev 0
# gptimer3: GR Timer Unit rev 0, 8-bit scaler, 4 32-bit timers, irq 8
# irqmp: Multi-processor Interrupt Controller rev 3, #cpu 1, eirq 0
# apbuart1: Generic UART rev 1, fifo 1, irq 2
# ahbrom8: 32-bit AHB ROM Module,  64 words, 7 address bits
# ahbjtag AHB Debug JTAG rev 0
```

Figure 5. Transcript of the Instantiation.

V. DATA HANDLING CHAIN FORMATTER

The purpose of the formatting is to specify a standard protocol. This formatter design is inspired from the Consultative Committee for Space Data Systems (CCSDS). This protocol is designed to meet the requirement of space mission to efficiently transfer the data of various length, type and characteristics over the network that involves a ground-to-space or space-to-space link of communication.

The formatter provides a unidirectional transfer of data from a single or multiple source user application to one or more destination user application through one or more subnetworks. The definition of Logical Data Path (LDP) is the path from the source user application to the destination.

A. Block Diagram

The only interface with the FPGA is the interface with the LEON3FT. The data from the LEON3FT through its CAN/SPACEWIRE/UART interface is collected from the LEON3FT by the formatter FPGA.

The data will be either in the form of serial data or in parallel form, the proposed formatter design is flexible for both the data through separate enable signal to switch between the two options. The data are typical payload data which are the output of the Charge Coupled Device (CCD) or PAN multispectral imaging devices.

The compressed data is converted into packet in this module. The packet is generated by adding the packet header into the payload data. The packet header contains the Application process ID (APID), Packet count, number of bytes in the packet.

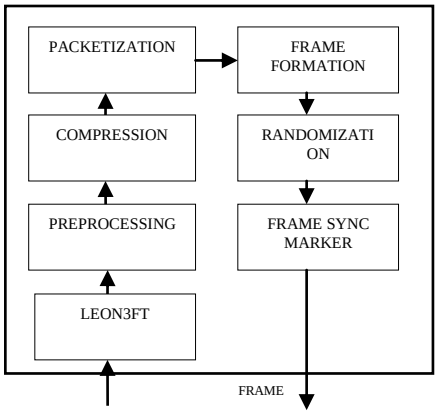

Figure 6. Block Diagram of formatter

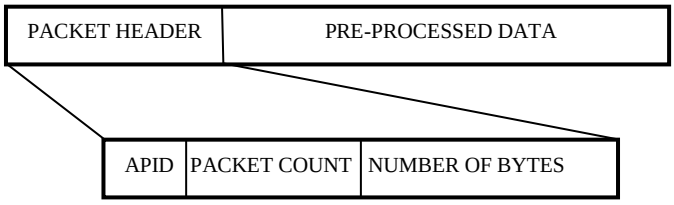

Figure 7. Packet Format

Here FRAME is formed by taking multiple packets and adding the frame header into the constant or fixed length.

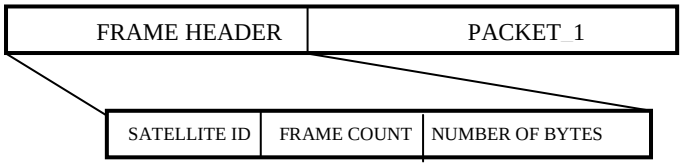

Figure 8. Frame Format

Pseudo-randomizer is used to improve symbol transition density as an aid to bit synchronization at the receiving end. The method for ensuring sufficient transitions is to exclusive-OR each bit of the Transfer Frame with a standard pseudo-random sequence.

Frame synchronization marker is used to identify the Frame from the ground station. The frame synchronization marker include the code as "1A CF FC 1D".

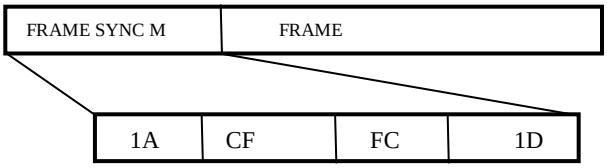

Figure 9. Frame Synchronization marking

B. *Flow Chart*

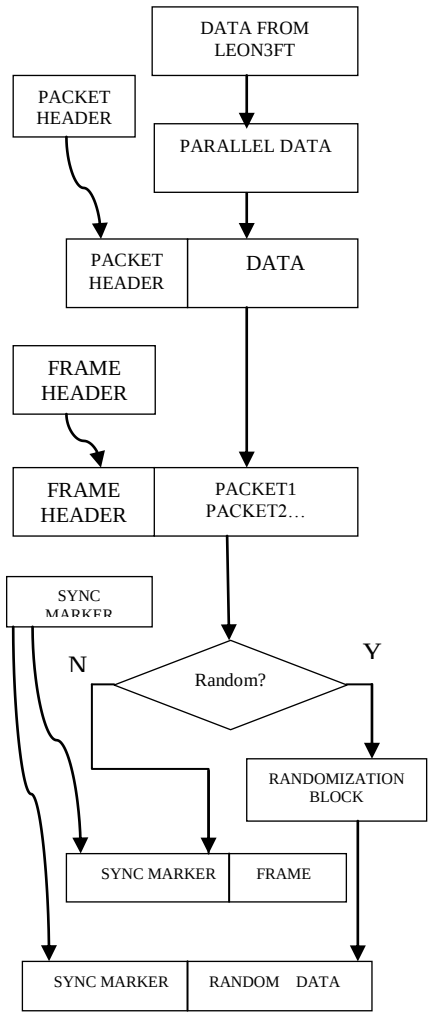

Figure 10. Flowchart of Formatter

C. *Output of the Formatter*

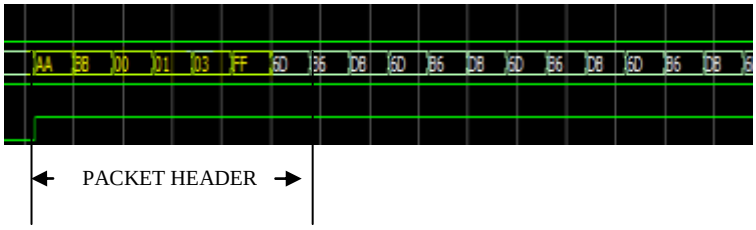

Figure 11. Packet with Packet header

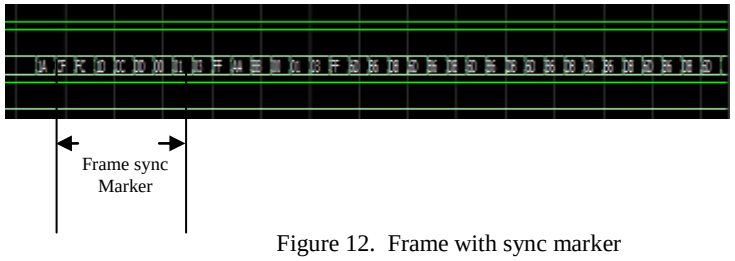

Figure 12. Frame with sync marker

VI. SOFTWARE FOR READ AND WRITE OPERATION FOR LEON3FT

A. *Flow Chart For The Application*

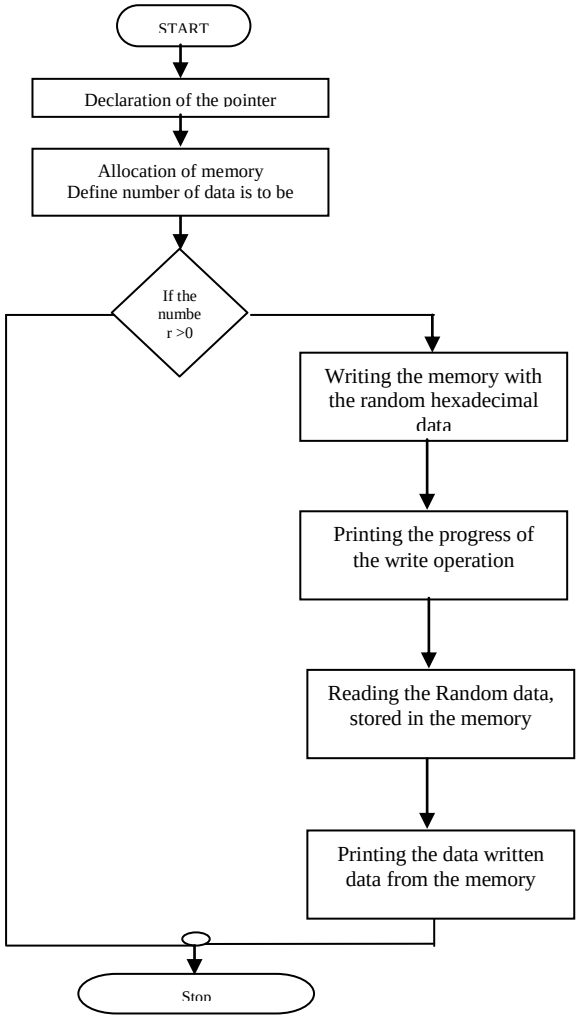

Figure 13. Flow chart of the program

LEON3FT Aeroflex Gaisler development tools are installed. With the help of Eclipse environment an application

is coded which will write some random data directly into the internal registers of LEO3FT and reading the data from the LEON3FT processor.

B. Execution Result

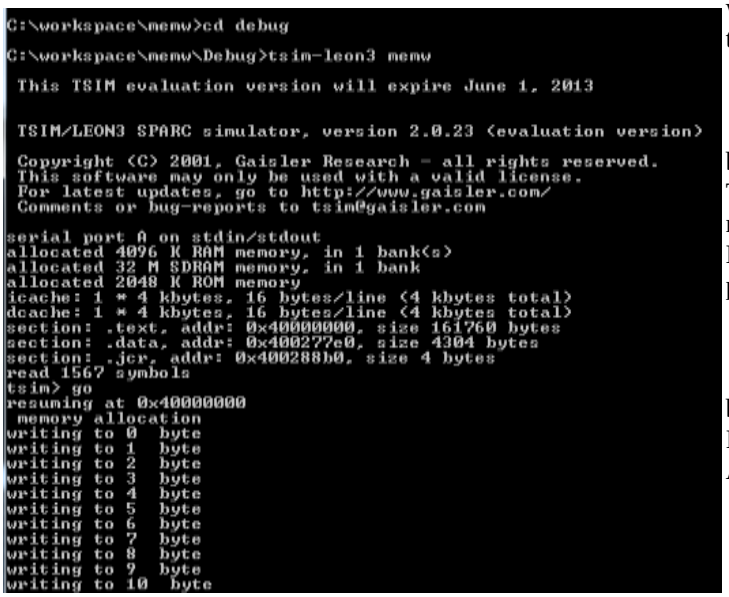

Figure 14. Output: Memory writing

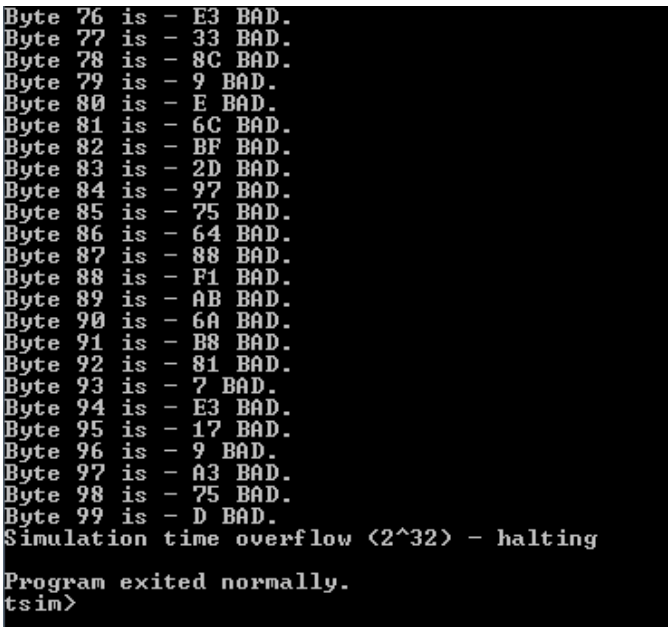

Figure 15. Output: Memory reading

VII. CONCLUSION

Data handling system using the LEON3FT processor is designed successfully, the outcome of this project is based on the result obtained from the four design modules.

In the first module schematic for the PCB fabrication for the proposed system is designed.

The second module IP core of the LEON3FT has been instantiated successfully which allows the flexibility for the future design simulation for the modified LEON3FT core.

In third module the formatter chain for a typical data handling system is designed inside a FPGA, which will successfully convert the payload data which has been acquired with the help of LEON3 processor interface into the frame format.

In the last module a software has been designed which will link the module three and module four in such a way that the data from the payload will be acquired through LEON3 is written into the LEON3 and it read the written data back from the LEON3 so that it will be available to the formatter

VIII. FUTURE SCOPE

Channel coding, compression, Encryption module can also be incorporated into the system to make the design robust.

The IP Core can be further modified to make the processor more dedicated for handling the data

More applications can be developed to utilize the programming capability of the LEON3.

IX. APPLICATION

Since the schematic is flexible, this can be used for any on board processing mission such as Space exploration satellites, Imaging satellites, Reconnaissance or spy satellites, Astronomy satellite.

REFERENCES

[1] Aeroflex Colorado Springs Inc., UT699 LEON 3FT/SPARC V8 MicroProcessor Advanced User Manual, Aug. 2010
[2] Aeroflex Colorado Springs Inc., UT699 32-bit Fault-Tolerant SPARC V8/LEON 3FT Processor Data Sheet, Feb.2011
[3] AMBA Specification (Rev. 2.0), http://www.arm.com/.
[4] SPARC assembly language reference manual", http://docs.sun.com/
[5] Cadence Design System, http://www.cadence.com/.
[6] Microsemi Axcelerator Family FPGAs ProASIC3 Revision-18 March 2012
[7] "GRLIB IP Library User's Manual (V1.1.0)", http://gaisler.com/products/grlib/grlib.pdf