"8086 IP Core based Data Acquisition System"

Mr. S. S. Rathod[1], Dr. S. A. Pardeshi[2]
[1]PG Scholar, Dept of E & TC, RIT College, Sangli, India
[2]Doctor, Dept of E & TC, RIT College, Sangli, India

***Abstract*—An Electric Propulsion System (EPS) uses electrical energy to change the velocity of a spacecraft.EPS configuration consists of a thruster module, feed system and power processing and control system. In this paper designing the Data Acquisition system (DAS) and various Interface interfaces for Telecommand and Telemetry systems for controlling different element of EPS. The necessary control signals are designed and implemented using IP CORE 86 in Libero software.**

***Keywords*—EPS, DAS, IP CORE 8086, Libero.**

I. INTRODUCTION

Spacecraft propulsion is a method of accelerating spacecraft's and artificial satellites into the atmosphere. Artificial satellites must be launched into orbit and subsequently must be placed in their nominal orbit. Once into desired orbit, they often need some form of attitude control so that they are correctly pointed with respect to the earth, son and possibly some astronomical object of interest. They are also subjected to drag from the thin atmosphere, so that to stay in orbit for long period of time some form of propulsion is occasionally necessary to make small corrections .Many satellites need to move from one orbit to another from time to time and this also requires propulsion. A satellite's useful life is over once it has exhausted its ability to adjust its orbit. We mainly designed the Telemetry, Telecommand, and Data Acquisition systems for controlling the various parameters of Electric Propulsion system. The Telecommand, Telemetry and Data Acquisition systems depend on requirement of the spacecraft. The Telemetry system is a health monitoring system. The telemetry downlink must provide the ground station team with information about the functioning of the elements and subsystem of the on-board spacecraft. The Telecommand system sends a command to control an on-board spacecraft system to from the ground station.

II. SYSTEM DESIGN

A. Clock Division

A clock divider or a frequency divider is a simple component with an objective of reducing the input frequency in order to cater to the timing constraints of various applications.

B. Code Logic

The clock divider is implemented through the use of a scaling factor and counter.

The scaling factor is the relationship between the input frequency and desired output frequency. Scaling factor: fin/fout
The counters run through half of the scaling factor in order to generate one complete cycle for the duration of the scaling factor.

Input Frequency= 12 MHz Desired
output Frequency = 1 KHz

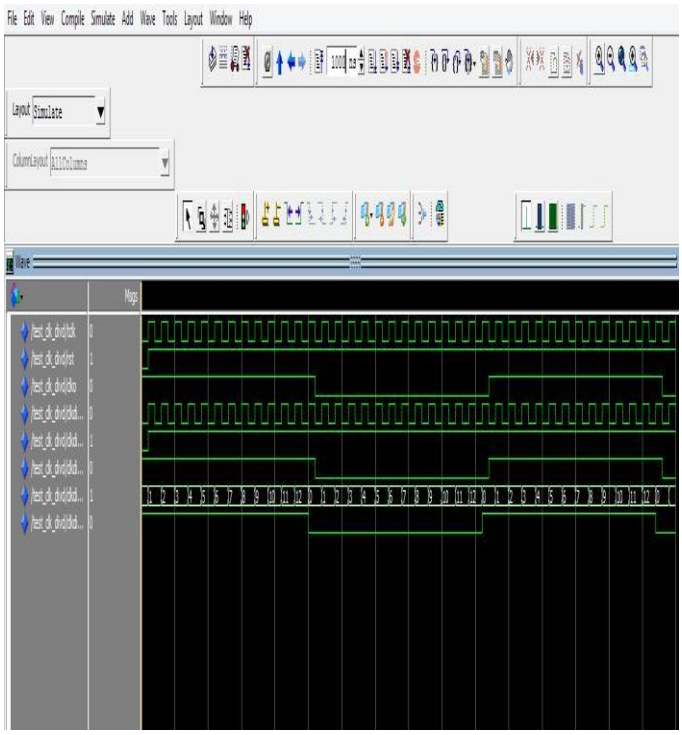

Fig 1. Simulation of clock division

III. MEMORY READ AND WRITE OPERATION

In IP core 8086 has a 20 bit address bus width and 8 bit data bus width. Therefore address memory size is 1Megabyte. It is divided into the parts ROM memory and RAM memory.

The ROM contains simple Jump instruction to jump to RAM memory address after reset the program.

3.2.1 Code Logic

When por (power on reset) is 0 then read the ROM memory. In Rom Memory giving the jump instruction to

jump on Ram address (0000:0400 h) memory location. Afterward address bus contains RAM memory address and fetching, decode the data on this particular memory location.

When por (power on reset) is 0 then read the ROM memory. In Rom Memory giving the jump instruction to jump on Ram address (0000:0400 h) memory location. Afterward address bus contains RAM memory address and fetching, decode the data on this particular memory location.

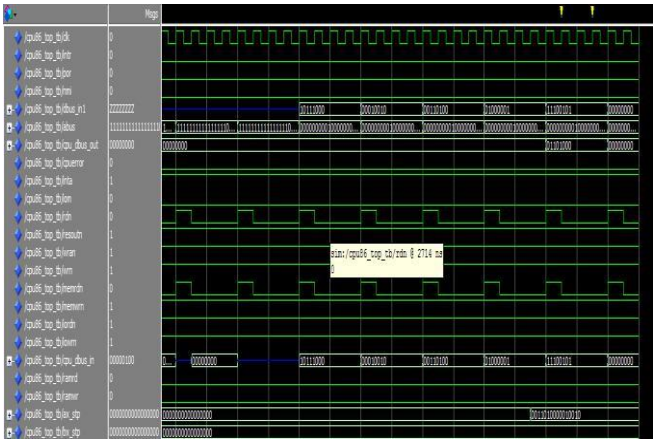

Fig 2 Simulation of memory read and write operation

IV. DATA ACQUISITION SYSTEM

Data Acquisition System (DAS) provides data acquisition, control, data analysis and comprehensive data reporting. The DAS provides high speed, multi channel system data acquisition to record pressure, propellant flows, temperatures, exhaust velocity, drive current and various signals required .The DAS is a user friendly method of collecting data in the Aerospace industry. In general DAS uses multiplexers, ADC and a register. The DAS is enabled by a start acquisition signal. We have used AD571 which is a successive approximation ADC

.The AD571 gets the start of conversion (SOC) signal from the register .The SOC signal is a 1-bit signal.

The part to be implemented in FPGA is the generation and reception of signals in the register.

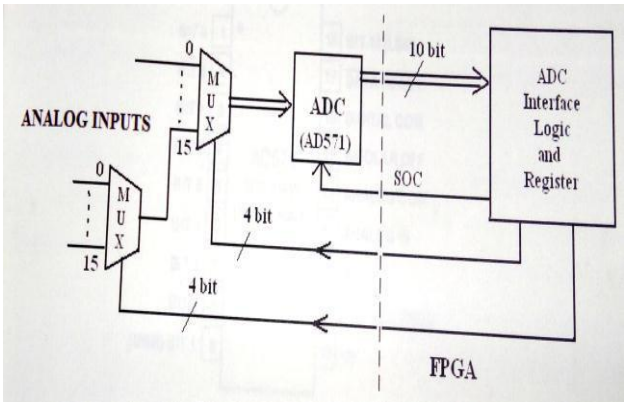

Fig 3 Design of data acquisition system

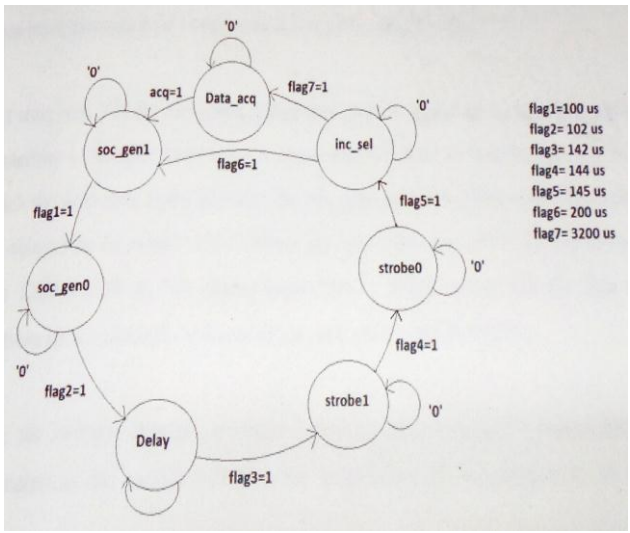

Fig 4 Finite state machine implementation of DAS

These operations are demonstrated using a finite state machine (FSM) implementation. The various states involved in the FSM implementation are data acquisition state, SOC generation1 state, SOC generation () state, delay state, strobe1 state, strobe () state and increment select line state. Each of these states is represented as enumerated state types. For example, the data_acq state is represented as enumerated state 000 and the corresponding states follow the respective order. Thus, all

the seven states are represented as 000,001,010,011,100,101,110,111 respectively.

The system remains in the data_acq state where the select lines and all other parameters are reset until the rise of start is enabled in which a delay of 100µs is generated after which the SOC signal is made high. This is followed by the soc_gen () state where a further delay of 2µs pulse width has been generated so that it can be fed to the ADC to start its conversion cycle.

Next, the delay state includes a further delay of 40µs which is the conversion time of the ADC. This is followed by the strobe1 state which makes the strobe signal high (at 142µs). The strobe () state, after a delay of 2µs, makes the strobe signal low. Hence a 2µs strobe signal has been generated (at 144µs) which is used to latch the data. The next state which is the increment_select line state is incremented by 1. Another 58µs delay is induced in this state so that the data can be latched after which the next state again becomes the soc_gen1 state (at 200µs) and the cycle repeats until the select line becomes "1111". When the select line is "1111" the commenced time would be 3200µs (200µs * 16 analog input lines), which means that the data has been acquired from all the channels and hence data_acq state is enabled again.

Therefore, for 16 input channels, the state transition cycle runs for 16 times (200µs each). After 16

iterations the output indicates the completion of the conversion of all the 16 channels.

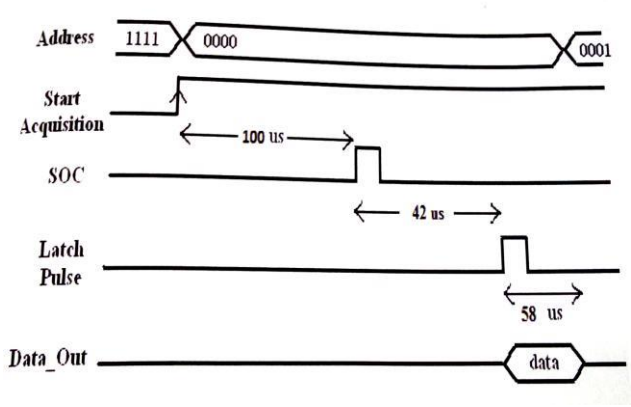

Fig 5 Timing Diagram of Single Channel

V. PULSE COMMAND GENERATOR

A mono-pulse generator is present to which a trigger pulse and a clock are given as input. The trigger pulse is essentially a bit that toggles and when toggle condition is detected, a pulse is generated. This pulse is called as toggle pulse. This pulse is used in the Telecommand decoder for initiating latch operations.

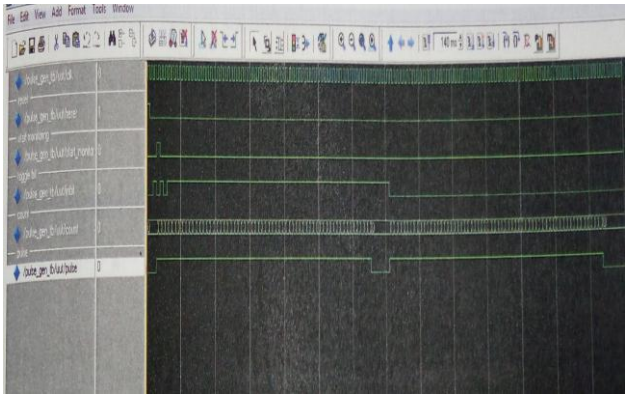

Fig 6 Simulation of pulse code generator

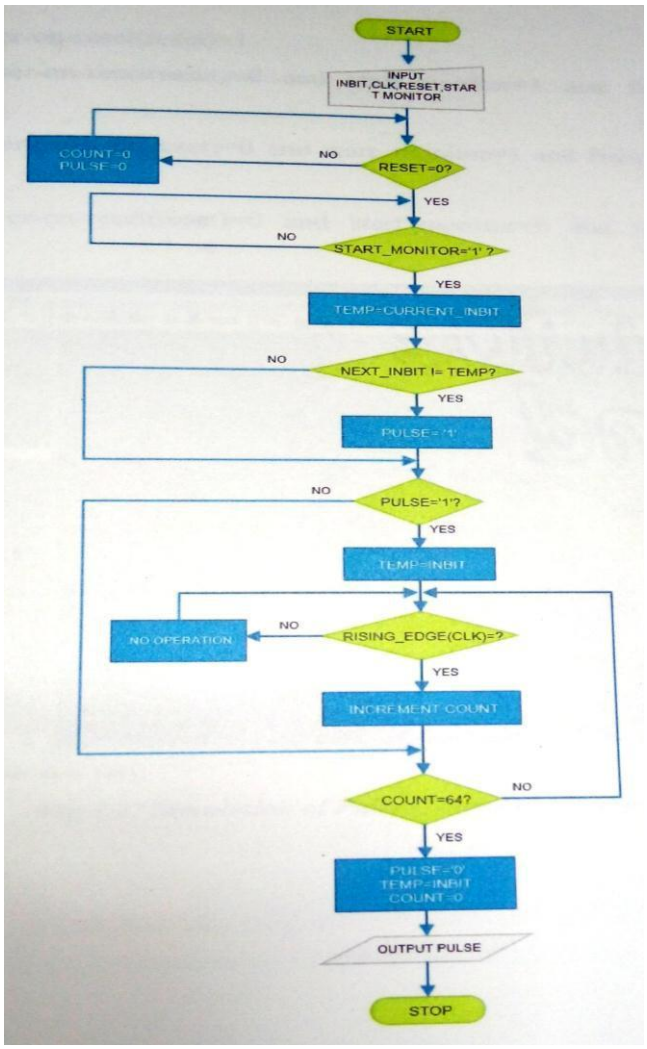

Fig 7 Flowchart for pulse code generator

VI. TELEMETRY AND TELECOMMAND INTERFACE

We have designed the telemetry interface for the telemetry system. The telemetry signals that contain information about the various health parameters are encoded into a specific frame format by the telemetry encoder onboard. This encoded data has to be decoded by the telemetry decoding unit in the ground station for analysis. Faults in any of the onboard equipment's are found and the cause is diagnosed. After the faults are analyzed, necessary steps to be taken are carried out to overcome these faults or other impediments.

The parallel address is given as the select input to the multiplexer, Based on this select input, one of the address lines from the decoder is selected and the multiplexed data is the output of the multiplexer block. The output from the multiplexer is 8-bit data. Initially, asynchronous loading is carried to synchronize all the functions; a clock synchronization block is used to integrate all the

functions. The multiplexed data is selected based on the select input from the SPC. The selected data from the MUX is written to the parallel to serial converter (PSC) based on the load signal from the clock synchronization block .thus ,the clock synchronization unit integrates and synchronizes the functions of all the blocks finally generating a serial output signal from the entire telemetry module. This signal is analyzed for carrying out further operations. Based on the analysis of this signal, further command signals are generated for the required operations of the on board equipment.

Code logic:

We have designed the interface in order to enable Serial to parallel conversion and vice versa required for telemetry as well as Telecommand interface.

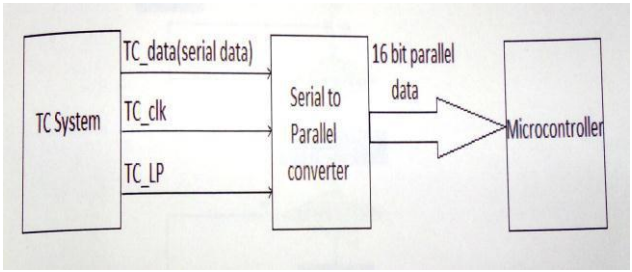

Fig 8. Schematic diagram of Telecommand system interface

8-bit serial to parallel conversion:

The input to serial to parallel converter are a 1kHz clock ,8-bit serial address and a latch pulse .the 8 bit serial address is shifted serially with every rising edge of the clock pulse. After every 8 bits, the latch pulse is generated to latch the 8 bit parallel address.

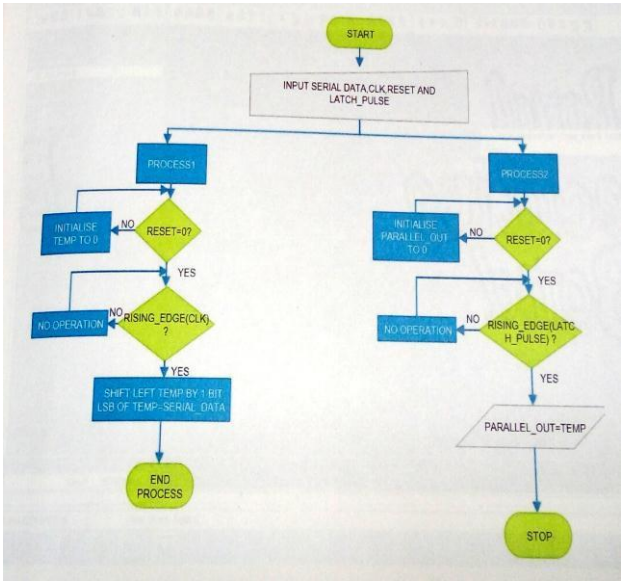

Fig 9 Flowchart for 8-bit serial to parallel conversion

VII. SOFTWARE REQUIREMENT
LIBERO IDE v8.4

After the codes are validated and simulated in Modelsim, the codes have to be synthesized before implementing in FPGA. In the Libero IDE, the main tools that are used for synthesis are Project manager; simplify Pro Actel and designer 8.4. in project manager ,all the HDL source files to be synthesized are added .the required device and required pin configuration are chosen .they are synthesized using Simplify Pro Actel .after synthesis, the detailed report of the code is obtained like the number of system gates used ,number of gated clocks, slew rate etc.

Finally, a*edn files is created .this files is used for post-layout simulation and for generating the programming file .then, designer v8.4 tool is used. In the designer tool, the *edn file is compiled .This step indicates if the design can be implemented in the configured FPGA. It shows the number of I/O pins used and other similar details. Next, the layout of the design is necessary. The required inputs and outputs are assigned to the pins of the device selected. All the inputs are placed and routed .then a delay file is generated using the Back-Annotate command. This delay file (*sdf) along with the *edn file is used for the post-layout simulations .Only, when the post-layout simulations are accurate, the required output can be obtained in the FPGA. Finally, a programming file is generated which can be downloaded into the FPGA for testing.

Modelsim actel :

In this project, Modelsim ACTEL is chosen as the software to edit, compile, and simulate the VHDL code for a single cycles MIPS processor project .only Windows and Linux OS are supported by ACTEL. Modelsim is used because it is a user –friendly software tool. The code is written using VHDL. It is complied and the simulation results can be verified easily. The code is validated by generating test benches in Modelsim. Modelsim supports both VHDL and Verilog languages.

VIII. FUTURE WORK

The behavior of the Data Acquisition System (DAS), Telemetry, and Telecommand model's VHDL code will be implemented on ACTEL ProASIC3E FPGA board. The required Hardware is shown in figure 10.

Fig 10. ACTEL ProASIC3E FPGA board

IX. CONCLUSION

The EPS needs the telemetry system, Telecommand system and data acquisition system to control the various elements involved in it. The design of these systems depends on various factors .So, the design varies with the spacecraft requirements. One such requirement is considered and the systems are designed accordingly to control the various parameters of the electric propulsion system .The design blocks are coded using VHDL language. They are validated using Modelsim ACTEL software tool by generating test benches and simulating the design codes.

REFERENCES

[1]. Pierpaolo Pergola,"Semianalytic approach for optimal configuration of electric propulsion spacecraft", IEEE transactions on plasma science, vol. 43, no.1, January 2015.
[2]. Savithaa , Rajiv R. Chetwani, M. Ravindra, K.M.Baradwaj, "Onboard processor validation for space applications", 978-1-4799-8792-4/15/$31.00_c 2015 IEEE.
[3]. Xiaohua Zhou, Yu Guo, Hao Li," FPGA design and realization of ground testing equipment bus interface for microwave remote sensor on satellite",978-1-4799-1114-1/13/$31.00 ©2013 IEEE.
[4]. Satish Sharma, P.N.Ravichandran, Sunil, P.Lakshmi Narsimhan, R.Seshaiah, "Wireless telecommand and telemetry system for satellite", 1-4244-1057-6/07/$25.00 ©) 2007 IEEE.
[5]. N.M.Desai, Rinku Agrawal, J.G.Vachhani, V.R.Gujraty and S.S.Rana," High speed data acquisition systems for ISRO's airborne and spaceborneradars",msdd/msg/sac,Space Applications Centre (sac), Ambawadi Vistar p.0 ISRO, Ahmedabad-380 015. India, proceedings of incemic – 2003.