 doi:10.20944/preprints201807.0590.v1

Article

Hardware Design of a Flight Control Computer System based on Multi-core Digital Signal Processor and Field Programmable Gate Array

Francisco Bilendo [1]; *francisco-bilendo@hotmail.com

Sheng Shouzhao [2] ; shengz@nuaa.edu.cn

*francisco-bilendo@hotmail.com; Tel.: +86 18551847950

Abstract: Flight Control System is an integrated avionics system equipped with the minimum required components for an autonomous flight. This paper focuses on the Hardware Design of the Flight Control System and presents specific details of the components and its interface. The system architecture is based on Field Programmable Gate Array and Digital Signal Processor. Employing these two processors in the flight control system would improve the Flight Control System performance in terms of fast sequential processing of high-level control algorithms. In addition to Field Programmable Gate Array and Digital Signal Processor, the flight control computer system will also make use of Global Positioning System and Micro Electro Mechanical System sensors. The project will be implemented using Altera's System On Programmable Chip builder, currently known as Qsys – Platform Designer implemented in Quartus-II. The system employs Nios-II processor which is 32-bit soft-core embedded-processor architecture designed especially for the Altera's family of Field Programmable Gate Array. From conceptualization to final design, this paper presents the functionality of the different modulus and complex interfaces employed in this Flight Control System.

Keywords: Field Programmable Gate Array, Flight Control System, and Hardware Design.

1. Introduction

The Unmanned Aircraft Vehicles (UAV's) have surely made a huge impact in today's world technology and currently, the UAV's are widely being used in real world applications in different sectors of technology, military as well as civil. It's right to define the UAV as an aircraft which doesn't require a pilot on board, but capable of autonomous flight because of it's on board equipped flight control system. The FCS is an embedded system with specialized functions, in which its complexity depends on the required tasks to be performed during the mission. The basic functions of the FCS is to collect data from its surroundings and following, process the data into useful information to be

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

used in order to stabilize the aircraft during the flight.

The size-to-weight ratio is a very important aspect to be considered in the world of UAV's and because of this aspect, various small size Unmanned Aircraft Vehicles , also known as (UAV) are facing a huge challenge concerning to limited processing power [2]. The challenge is to maintain high-end performance and high capabilities in small size UAV's, seen that it's a challenge to maintain this features into smaller vehicles. However, this features are in demand, note that the UAV's are widely being used in real world applications and different sectors of technology. Currently, there are multiple projects ongoing in the industry in order to improve numerous aspects concerning the unmanned aircraft vehicles. This project, aims to develop a hardware system design for UAV's based on FPGA and DSP. The implementation of the two processors in the system is expected to improve the processing power of the system. Also, the interface between the system components would be improved, resulting in a better FCS performance.

2. Materials and Methods:

2.1 Hardware Architecture

The main objective of this hardware design is to improve the processing power and optimize the performance to size ratio for a widely applicable flight control system (satisfactory for larger size vehicles as well as smaller size) a system enabled by Micro Electro Mechanical System (MEMS) sensor and embedded processing technologies.

A. Field Programmable Gate Array and Digital Signal Processor.

Field Programmable Gate Array (FPGA) plays a huge role in the system, seen that the Soft-Core CPU is build inside the FPGA and all system components are connected to the Soft-Core CPU through the FPGA. Digital System Processor (DSP) plays also an important role by providing the board its sequential processing power, while the FPGA perform low-level and parallel interface functions for external components such as GPS and the IMU. The DSP is multi-core based TMS320C6678 with many advantages for processing abilities, and it communicates with the FPGA through UART port to Avalon Data Bus.

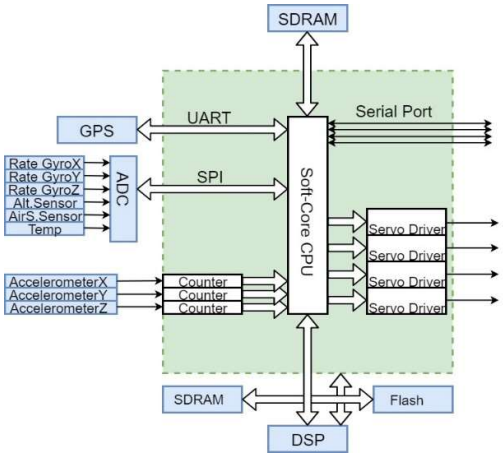

Fig. 1. Embedded System Architecture

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

B. Sensors and Actuators

The sensor board is represented by ADIS16350, which is a Tri Axis Gyroscope and Accelerometer, which is basically the complete IMU set. This is augmented with absolute and differential pressure sensors and a μBlox GPS OEM module.

C. System Design with Nios-II

The embedded system in the FPGA employs Nios-II which is a processor system equivalent to a microcontroller or "computer on a chip" that includes a processor and a combination of peripherals and memory on a single chip [4, 6]. The idea is to combine a microcontroller based system with the FPGA in order to overcome the limitations of most FCS currently in the market, and improve the performance.

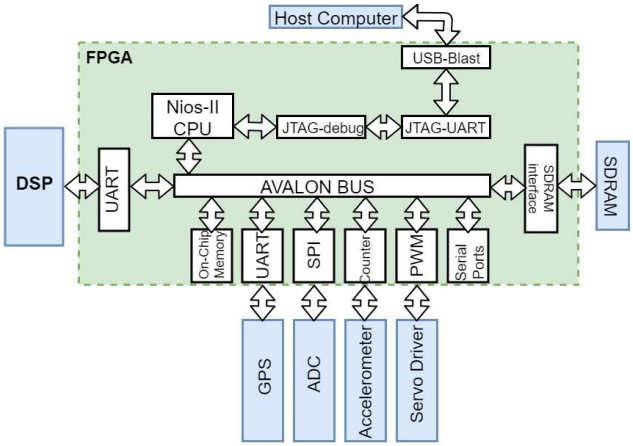

Fig. 2 System On Programmable Chip (SOPC) Hardware Design Architecture

2.2 System Interface

System components communicate within each other through a data bus called Avalon Interface. The Avalon interface family defines interfaces appropriate for streaming high-speed data, reading and writing registers and memory, and controlling off-chip devices.

D. Avalon Data Bus and Memory

The Avalon interface is a synchronous interface defined by a set of signal types with roles for supporting data transfer. There are two types of Avalon interface port, Avalon Master Port and Avalon Slave Port. The Avalon Master Port initiates transfer and the Avalon Slave Port responds to transfer requests. The Avalon Bus communicates with the system memory (On-Chip Memory and SDRAM) in order to store data for the system [5, 6].

E. Universal Asynchronous Receiver/Transmitter

The Universal Asynchronous Receiver/Transmitter (UART) core with Avalon interface implements a method to communicate serial character streams between an embedded system on an Intel FPGA and an external device (in this case, Multi-Core DSP and GPS). The core implements the RS-232 protocol timing, and provides adjustable baud rate, parity, stop, and data bits. The core

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

provides an Avalon Memory-Mapped (Avalon-MM) slave interface that allows Avalon-MM master peripherals (such as a Nios-II processor) to communicate with the core simply by reading and writing control and data registers [6, 7, 8, 9,].

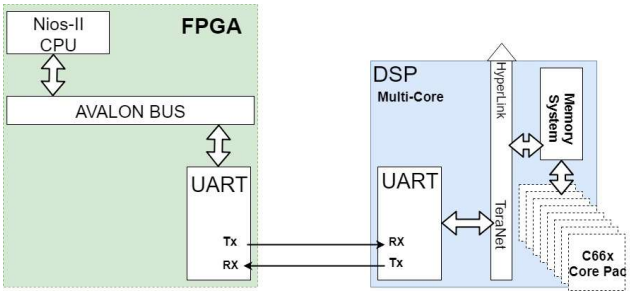

Fig. 3. UART Communication Protocol between FPGA and DSP

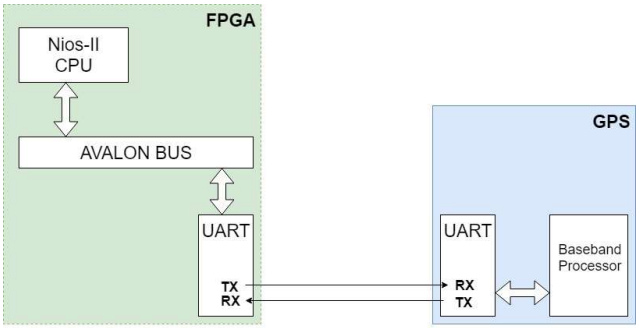

Fig. 4. UART Communication Protocol between FPGA and GPS

As for the DSP/FPGA, the UART performs Serial-to-Parallel conversion on data received from the FPGA and parallel-to-serial conversion on data received from the DSP. The DSP can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communication link [9].

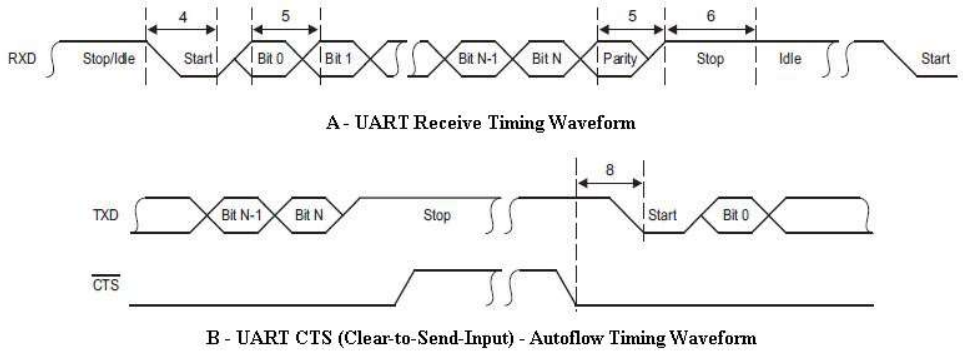

Fig. 5. DSP-UART Receive Timing Waveform and CTS (Clear-to-Send-Input) Autoflow Timing Waveform [9]

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

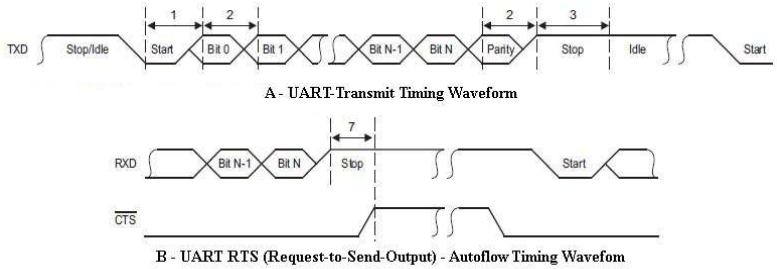

Fig. 6. DSP-UART Transmit Timing Waveform and RTS (Request-to-Send-Output) Autoflow Timing Waveform [9]

F. Serial Peripheral Interface

The Serial Peripheral Interface (SPI) core with Avalon interface implements the SPI protocol and provides an Avalon Memory-Mapped (Avalon-MM) interface on the back end. The SPI core can implement either the master or slave protocol. The SPI is the communication protocol between the FPGA and the IMU.

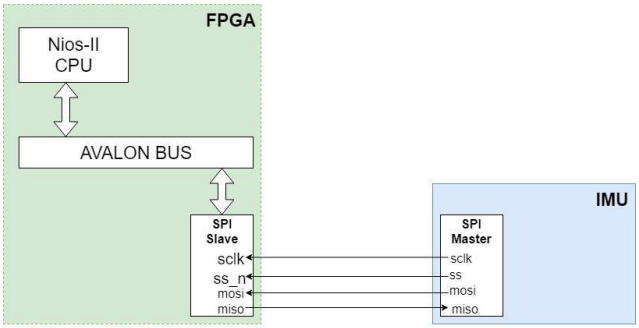

Fig. 7. SPI Interface Protocol between FPGA and IMU

Fig. 8 shows ADIS16350 DIN Bit Sequence and SPI sequence for Read Commands. ADIS16350 requires only 5.0 V power supply and a four-wire industry standard Serial Peripheral Interface (SPI). A simple register structure is in charge of handling the outputs and user programmable functions. Each of this registers is 16-bits in length with its own unique bit map. The SPI interface of ADIS16350 includes four different signals: Chip Select $\left(\overline{CS}\right)$, Serial Clock (SCLK), Data Input (DIN), and Data Output (DOUT) [10]. Figure 3.5 shows the SPI interface between the FPGA and ADIS16350 device. The signal symbol might be different than in figure 3.6, but they have the same functions. Master Out Slave In (mosi) - Output data from the master to the inputs of the Slaves. Master In Slave Out (miso) - Output data from a slave to the input of the master. Serial Clock (sclk) - Clock driven by the master to slaves, used to synchronize the data bits. Slave Select (ss_n) - Select signal (active low) driven by the master to individual slaves, used to select the target slave.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

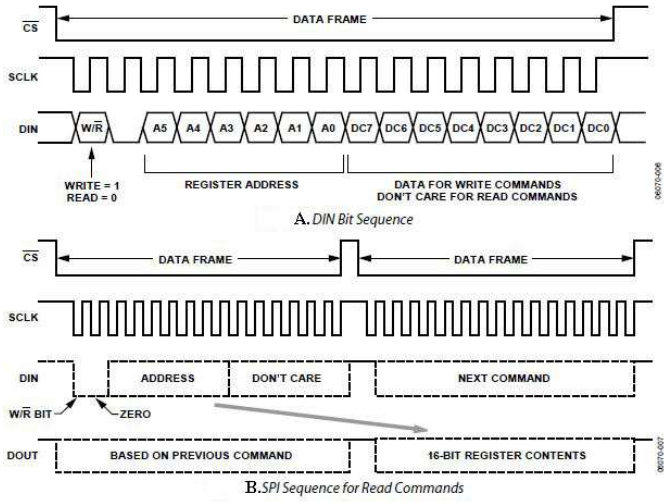

Fig. 8. ADIS16350 DIN Bit Sequence and SPI sequence for Read Commands

G. *Counter and Pulse-Width Modulation*

A Pulse-Width Modulation interface to control Servo Drivers is required. A Servo Driver is controlled by sending a pulse every certain amount of ms (example: 20ms), or a certain amount of Hz (example: 50Hz), to the servo. The duration or width of the pulse determines the angle output by the servo. The specifications of servos may vary between manufacturers.

H. *JTAG- UART*

The JTAG UART core with Avalon interface implements a method to communicate serial character streams between a host PC and a Platform Designer system on an Intel FPGA. The user-visible interface to the JTAG UART core consists of two 32-bit registers, data and control, which are accessed through an Avalon slave port. An Avalon master, such as a Nios-II processor, accesses the registers to control the core and transfer data over the JTAG connection. The core operates on 8-bit units of data at a time; eight bits of the data register serve as a one-character payload [6].

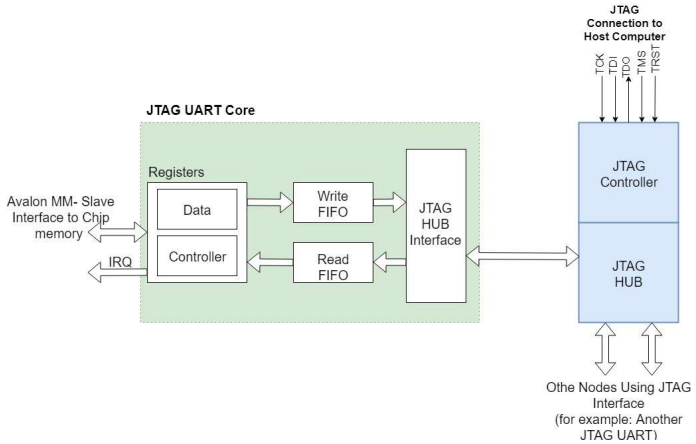

Fig. 9. JTAG UART interface with the Avalon Bus

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

2.3 System Design Flow

For research purpose, an embedded system is built on Altera's Platform Designer also known as Qsys. Qsys-Platform Designer is a system-level integration tool included as part of the Intel Quartus Prime software.

I. Nios-II System Design Flow

The Nios-II Development Flow consists of Hardware and Software system design. After selecting and configuration of the system components, Qsys-Platform Designer automatically generates interconnect logic to integrate component into hardware system. Following is the design flow of Nios-II system.

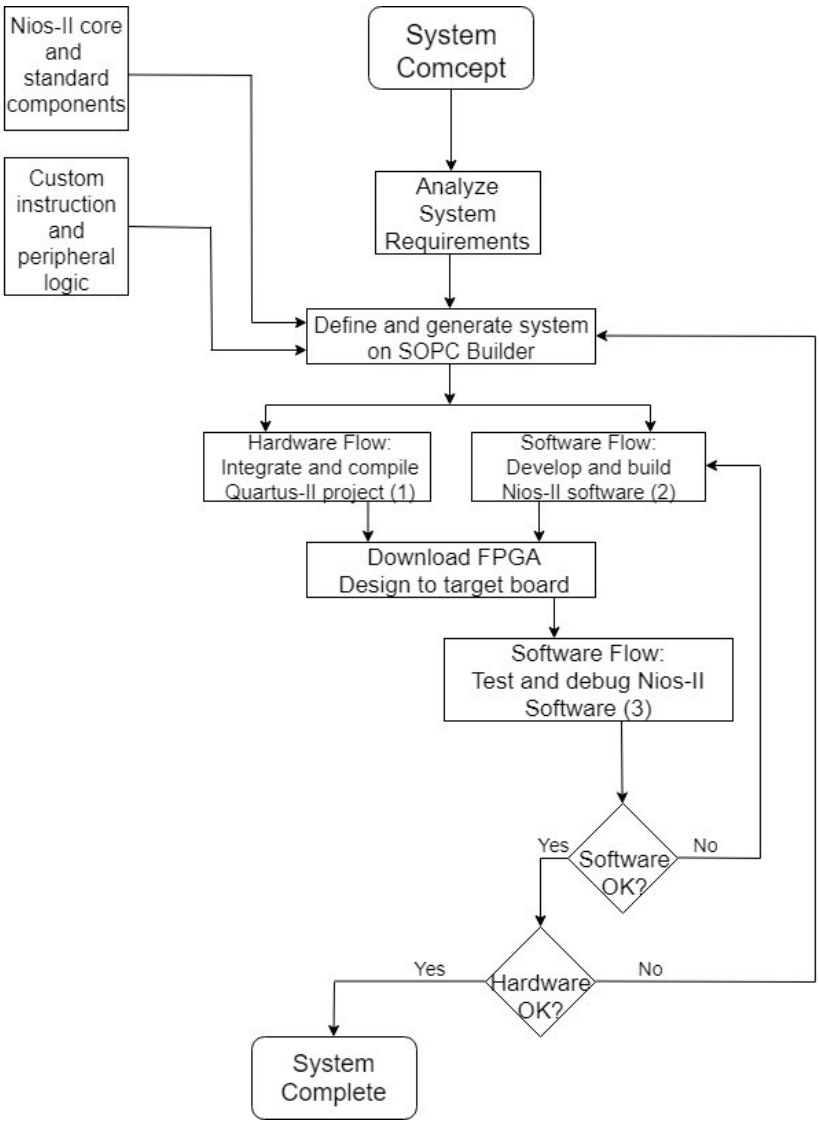

Fig. 10. Nios-II System Design Flow

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

3. Results

Fig. 11 shows Nios-II system components and its respective interface. Nios-II is configured to 'e/economy' core version. Nios-II Reset Vector Memory and Exception Vector Memory are assigned to SDRAM. The Avalon Bus is connected to all system components. External components interfaces have been established according to its type of communication protocol.

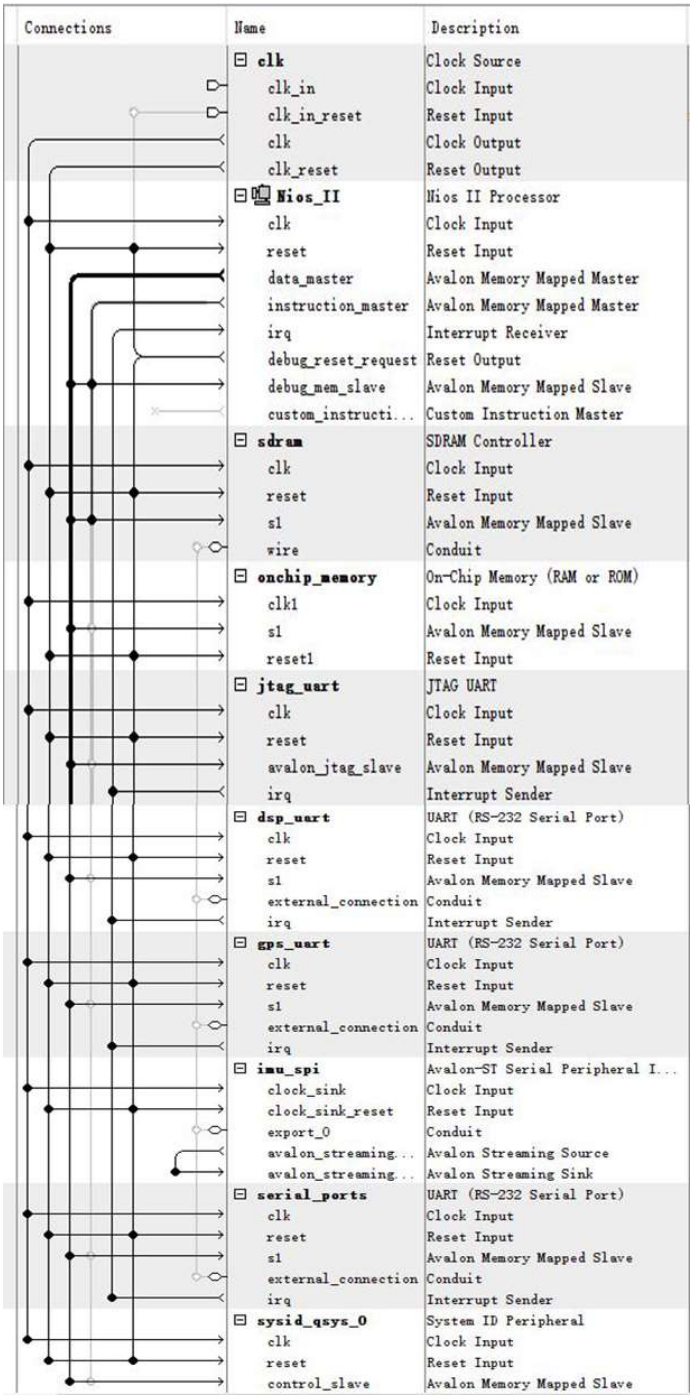

Fig. 11. Embedded System Design on Qsys-Platform Designer

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 July 2018 doi:10.20944/preprints201807.0590.v1

After selecting all the components, the user has to assign priority of the components which is IRQ no. SOPC builder automatically assign base address to all components. When everything is settled, the system is generated and Quartus-II is ready to provide the schematic view of the generated system, as shown in fig. 12. Based on the target board, the pins need to be assigned and the system is ready to be compiled.

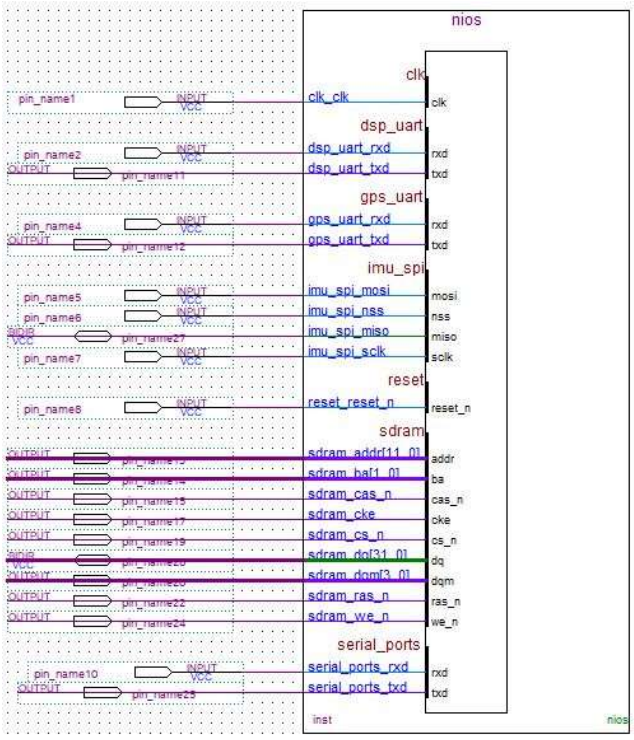

Fig. 12. Embedded System Schematic View

4. Discussion

This research results proves that it is possible to build a Flight Control Computer System based on Field Programmable Gate Array, with a Soft-Core CPU feature, and combined with a Digital Signal Processing, for the improvement of processing power. After the system generates the schematic view in figure.12, the user can assign the pins, based on the user's board. The system can be then, loaded in the board and further experimentations can be done.

5. Conclusions

Various small size Unmanned Aircraft Vehicles are facing a huge challenge concerning to limited processing power. The challenge is to maintain high-end performance and high capabilities in small size UAV's, seen that it's a challenge to maintain this features into smaller vehicles. In this paper, a new method to overcome this challenge has been shown and performed. The method is a combination of FPGA/Soft-Core CPU and a Multi-Core DSP. Employing these two processors in the flight control system improves the FCS performance in terms of fast sequential processing of high-level control algorithms. A demonstration has been made on how to implement a soft-core CPU in the FPGA and a schematic view of the generated system has been show. There is no doubt that this method is perfectly adaptable into real world application.

Author Contributions:

Francisco Bilendo: Project execution

Sheng Shouzhao: Accademic Supervisor of Mr. Francisco Bilendo.

Funding: This research received no external funding.

Acknowledgments:

Francisco Bilendo: I would like to thank God the Almighty for the grace. Secondly, I would like to thank my parents for the love and support. Last but not least, I would like to thank my academic supervisor "Sheng Shouzhao" for the support and guidance.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guowei Cai; Ben M.Chen; Tong Heng Lee. "Unmanned Rotorcraft Systems;" Springer: New York, NY, USA, 2011.
2. Henrik B. Christophersen; Wayne J.Pitckell; Adrian A. Koller; Suresh K. Kannan; and Eric N.Johson. "Small Adaptive Flight Control Systems for UAVs using FPGA/DSP Technology;" Atlanta GA 30332-0150.
3. Robin Van Wyk. "Development of an Integrated Avionics Hardware System for Unmanned Aerial Vehicle Research Purposes," 2011.
4. Altera, "Nios-II Processor Reference Handbook," 2014.
5. Altera, "Avalon Interface Specifications, "2018.
6. Altera, "Embedded Peripherals IP user Guide," 2017.
7. Thamer Salman Ali. "Handheld Navigation System Implementation on FPGA Board," 2011.
8. Ublox, " LEA-6/NEO-6 Hardware Integration Manual," 2010
9. Texas Instruments, "Multicore Fixed and Floating-point Digital Signal Processor," 2014.
10. Analog Devices, "Tri Axis Gyroscope & Accelerometer, Preliminary Technical Data," USA, 2007.
11. Altera, "Introduction to the Quartus-II Software," California, USA, 2010.