Effective FPGA implementation of algorithms for [redacted]

Insitute of Electronic Systems
Warsaw University of Technology
Warsaw, Poland
Email: [redacted]@stud.elka.pw.edu.pl

***Abstract*—This paper describes the implementation of [redacted] The main goal of this work is to investigate the properties of proposed implementation such as efficiency, speed of the system, energy consumption and use of embedded logic resources. Final result should confirm that FPGA implementation of this algorithm can be much more effective in many ways, than created for microcontrollers, so they could be substituted by FPGA chips in a few branches of industry.**

I. Introduction

Nowadays, there is a large need for real-time determining [redacted]

There are two basic solutions, which can provide values [redacted]

$$\varphi(t) = \int_0^t \omega(\tau) d\tau \tag{1}$$

[redacted] this and it's used to calculate roll and pitch.

$$\theta = \arctan\left(\frac{A_y}{A_z}\right) \tag{2}$$

[redacted]

Fig. 1. [redacted] [2]

$$\phi = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \tag{3}$$

[redacted] to local electromagnetic interferences, which can also increase a measurement error. [2]

II. Sensor Fusion

To reduce limitations related to each sensor, many engineers are using special signal processing algorithms, which combine data from all three sensors [3]. This operation is called sensor [redacted]

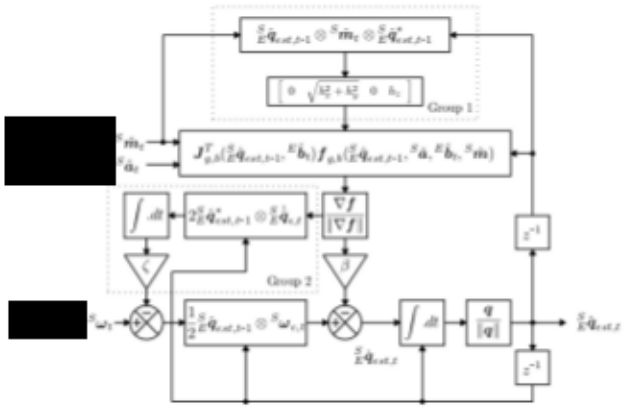

Fig. 2. [redacted] [5]

[redacted]

III. [redacted] Filter

[redacted] uses quaternion arithmetic. A quaternion [6] is a four-dimensional extension of complex number that can be used to represent the orientation of a ridged body or coordinate frame in three-dimensional space. In this work quaternion $\hat{q}$ is represented as four element vector:

$$\hat{q} = [q_1 \ q_2 \ q_3 \ q_4] \tag{5}$$

where q_1 is a real part and q_2, q_3, q_4 are imaginary parts. In this work quaternions describe rotation of one frame relative to the second frame and some vector described in that frame. Denoting first frame by *B*, second by *A* and vector by ${}^{A}\vec{n}$ this relation can be described as:

$$q_B^A = [\cos\frac{\alpha}{2} \ -n_x \sin\frac{\alpha}{2} \ -n_y \sin\frac{\alpha}{2} \ -n_z \sin\frac{\alpha}{2}] \tag{6}$$

where α is a rotation angle.

[redacted]

IV. FPGA implementation

[redacted]

square roots) [5]. This algorithm is sequential, but it can be splitted to few subsequent rounds. As was said before this algorithm comes down to transforming and multiplying of matrices. With FPGA chips many operations can be done in parallel for each round, so each filter update on FPGA should be much more faster than on CPU or MCU. Moreover, splitting algorithm to rounds in hardware opens possibility to use pipeline-processing, which can also increase speed of computations.

[redacted]

by 30. The divisions and square roots (which are very computational expensive) are used to normalise quaternions, in other words to calculate an inverted square root. This operations can be completely removed, by replacing them with Newton-Raphson Method [7], which is the best method for fixed-point arithmetic, typically used in FPGA chips.

Last problem which must be solved is an effective implementation of inverse trigonometric functions *arctan* and

[redacted]

The main goal of this work is to investigate how efficient would be implementation in medium-class FPGA using proposed solutions, compared to the MCU version.

V. Device Architecture

A. Component Selection

In this work it is assumed that FPGA chips would be medium-class. Searching of a proper circuit was constrained to products offered by three biggest FPGA companies: Altera, Xilinx and Lattice Semiconductor. Finally it was decided to

[redacted]

[redacted]

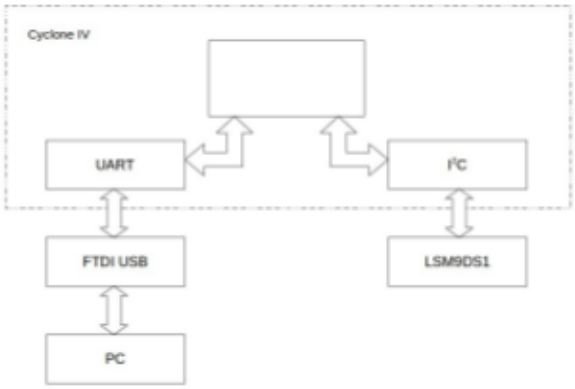

Fig. 3. Device architecture block scheme

B. Device Architecture

Figure 3 shows the architure of the whole device, built

[redacted]

VI. Results

At this moment implementations of an UART and I^2C in FPGA are done. Implementation of the first version of [redacted] which don't use pipeline processing is still in progress. There is also a reference version implemented on STM32F303VCT6 MCU, to compare with FPGA version.

References

[1] [redacted]
[2]
[3]
[4]
[5]
[6] K. Shoemake, "Quaternions," University on Pensylvania, Tech. Rep., .
[7] P. C. K. Piromsopa, C. Apomtevan, "Fpga implementation of fixed-point square operation," Chulalongkorn University, Tech. Rep., .
[8] R. Andraka, "A survey of cordic algorithms for fpga based computers," Andraka Consulting Group Inc., Tech. Rep., .