Implementation of FPGA based star tracker for spacecraft attitude determination

Oscar Björkgren

Supervisors, Åbo Akademi: Johan Lilius, Annamari Soini
Supervisors, Aboa Space Research Oy: Tero Säntti, Tuomo Komulainen
Master's thesis in Computer Engineering
Åbo Akademi
2021

Abstract

Keywords

Abbrevations

Contents

List of Figures

1. Introduction

A hardware based solution for a software algorithm is sometimes needed within application areas where performance matters the most. One example of this is the aerospace industry where the most optimized solution is sometimes the only choice. In this thesis I work together with Aboa Space Research Oy to create a hardware implementation of a star tracker system using a software component as a basis. A star tracker is commonly used as part of a larger system where the role of the tracker is to provide orientational data to other components. This information might be used for navigation and control or in combination with scientific instruments to help with further analysis. Hardware design can be inspired by software algorithms when designed for a specific purpose as in this case. The software defines the functional part of the system, which leaves a big part of the supporting architecture to be designed. In this thesis I will explain the process of implementing functionality to a digital circuit, by example.

To complete a set of tasks in any environment, a scientific project needs a well defined tools to reach its goals. In a space environment the tools often needs to be multi purpose for optimal use. An image sensor is in this case used for providing reference data for a star tracker. This image sensor can provide data to multiple systems by reuse of data or capturing new images with other settings. The purpose of the system in this project is to refine data from pre defined sensors to provide additional and reinforcing information about the environment for the use of other scientific instruments. The goal of this thesis is to create a roadmap of the system design process and include theory of relevant areas.

2. System Requirements Analysis

3. Aerospace concepts

4. Custom hardware design

4.1 Single and general purpose processors

A general purpose processor refers to a hardware computing platform which is designed for universal use with broad benefits across different problem solving domains. Computing platforms such as microprocessors and CPUs are examples of this type of processor, their hardware logic is implemented in such a way that it enables a large variety of computations to be performed. The characteristics of a processor for general problem solving are well suited to common tasks where there might be many hard- and software abstractions between the application interface and hardware logic. This is a necessity in for example PCs.

When hardware is required for a single purpose with a finite set of tasks to be performed there will be drawbacks with using hardware designed with flexibility in mind. This could be compared to the use of a multitool for driving a screw into a piece of wood when in fact only a screwdriver is needed. It is not the wrong way to do it but there is a more optimized way of achieving the result. Any CPU could be used for computer graphics calculations but since GPUs are designed for the single purpose of this type of processing it is a better tool for the job.

Single purpose processors have a rich history with roots in the early ages of computers. Vector processors were for example used in supercomputers. This was before computers were seen as general purpose technology and high performance computing was about as common as ordinary PCs. Since recent years the utilization of specialized hardware has been rising, with new application areas evolving. Single purpose processors such as GPUs and FPGAs have found themselves into areas such as machine learning, cryptocurrency mining, and other high performance computing applications. As the hardware used in aerospace applications generally is specialized due to strict requirements, single purpose computing platforms are heavily used. For example, the NASA perseverance Mars rover

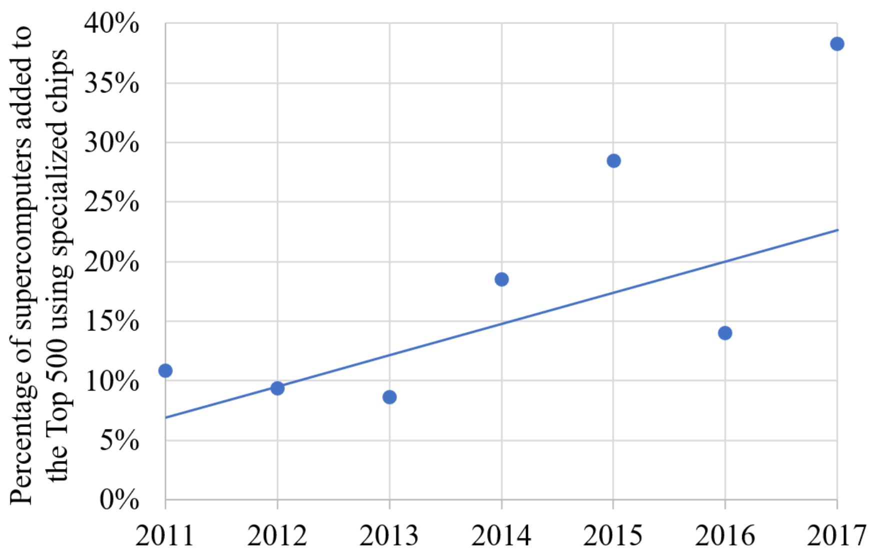

Figure 4.1: Graph showing increasing popularity of specialized hardware in high performance computing [1]

relies heavily on Xilinx manufactured FPGAs for different tasks, such as image processing pipelines [5].

4.2 FPGA Overview

A field programmable gate array, or FPGA, is a digital integrated circuit that consists of millions of logic blocks that can be configured to perform different operations. The process of configuring the logic blocks is called FPGA design and can be compared to the design of integrated circuits. Reconfigurability and ease of design are main advantages when using an FPGA as computing platform [6]. When comparing an FPGA design to a solution that is developed using software, the FPGA stands out with process pipelining and high level of parallel processing. When comparing FPGAs among each other the main performance metrics are power consumption and number of logic cells, which describes and complexity of the design that the FPGA can fit.

Xilinx is one of the key companies involved in FPGAs both currently and historically. They introduced the world to FPGAs in 1985 and has since then led the programmable logic technology industry with a 51% market share of programmable logic device suppliers in 2017. Major end market categories for Xilinx are aerospace, defense and communications.The main competitor of Xilinx is Intel with a market share of 37% [7][8].

The basic internal elements of an FPGA consists of look-up tables, flip-flops, wires and input/ output pads. Using these elements the FPGA design is implemented in the hardware. The architecture is explained by figure 4.2 which shows a matrix arrangement of configurable logic blocks which are connected to each other with wires running between them. The wires also run to the input/ output pads which enables interfacing with off-chip components such as SDRAM or sensors [2]. Configurable logic blocks contains the look-up tables and flip flops. An overview of a CLB is shown in figure 4.3.

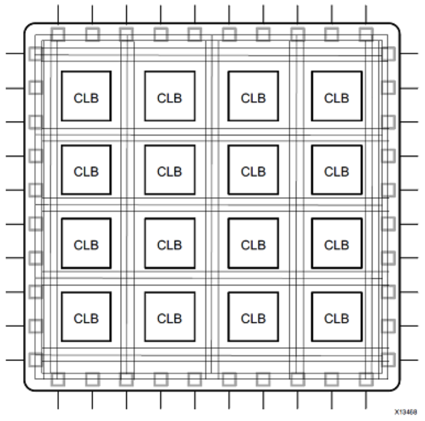

Figure 4.2: FPGA architecture. An FPGA consists of configurable logic blocks, wires and input/ output pads [2].

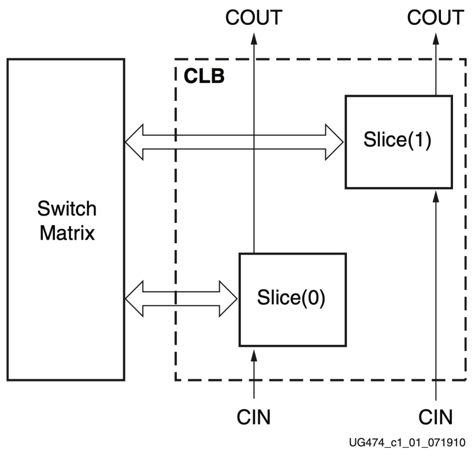

Figure 4.3: Diagram of a configurable logic block. A CLB consists of logic element containers called slices, inputs and outputs, and a switch matrix which connects the CLB to other CLBs [3].

The slices in figure 4.3 represents the core element of any FPGA. This is an instance of an element where a significant amount of an FPGA design will be implemented. The slice consists of look-up tables, storage elements, multiplexers and carry logic [3].

Look-up tables or LUTs, are mainly used as logic elements in FPGAs. They can implement any boolean logic operation by combining memory cells with multiplexers. The look-up table is essentially a truth table that can be used for reading the result of boolean operations. Memory cells and multiplexer routings are initialized by the FPGA design, which is what makes the look-up table element configurable [4]. An example of a 4 memory cell wide LUT is shown in figure 4.4.

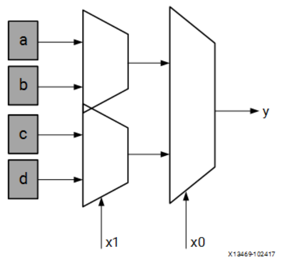

Figure 4.4: Look-up table with 4 memory cells and two multiplexers. The FPGA design initializes memory cells **a-b** and uses inputs **x1** and **x0** to select value **y** [4]

The circuit elements discussed in this chapter forms the basic blocks of FPGA technology. A larger set of elements are often included in an FPGA to enable some optimizations and further functionality in the design. For example there are different types of slices a CLB can contain which makes it more suitable for storage purposes rather than logic. Signal processing pipelines are a popular use case for FPGAs and they often contain slices specialized for operations in this problem domain. Digital signal processing can be accelerated by parallelizing operations in a SIMD fashion, thus to maximize the advantage, there can often be found a large number of these slices. Figure 4.5 shows an example of a FPGA configuration with some of these specialized blocks mapped out. The green CLBs are designated to memory usage and are closely coupled to the red DSP blocks to enable fast processing of the stored data. The purple external memory controllers uses the input/ output pads to interface with external memory such as SDRAM.

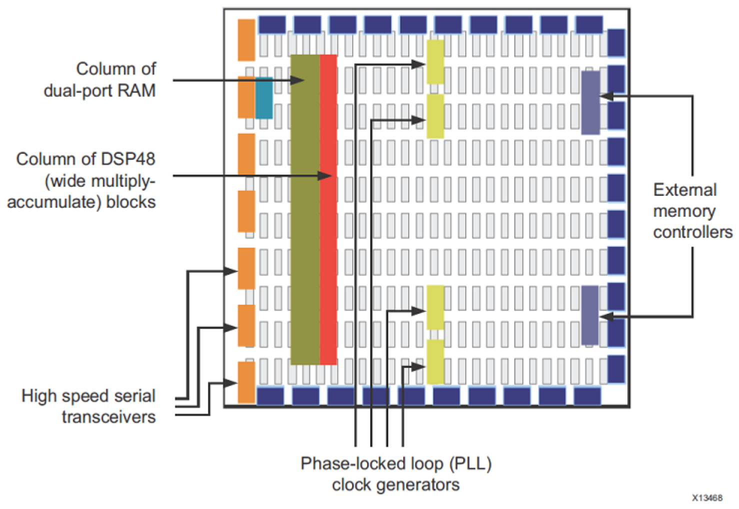
Column of dual-port RAM
Column of DSP48 (wide multiply-accumulate) blocks
High speed serial transceivers
Phase-locked loop (PLL) clock generators
External memory controllers
X13468

Figure 4.5: [2]

4.3 Xilinx 7-Series

5. Discussion

Bibliography

[1] Svenja Spanuth Neil C. Thompson. The Decline of Computers as a General Purpose Technology. Visited 20.1.2021, 2018.

[2] Understanding FPGA Architecture. https://www.xilinx.com/html_docs/xilinx2017_4/sdaccel_doc/odz1504034293215.html. Visited 19.4.2021.

[3] Xilinx Inc. *7 Series FPGAs Configurable Logic Block*, 2016.

[4] LUT. https://www.xilinx.com/html_docs/xilinx2017_4/sdaccel_doc/yeo1504034293627.html. Visited 21.4.2021.

[5] Xilinx Employee. Touchdown! NASA's Perseverance Rover Lands on Mars with Xilinx FPGAs On Board. Visited 19.4.2021, 2021.

[6] What is an FPGA? https://www.xilinx.com/html_docs/xilinx2017_4/sdaccel_doc/gac1504034293050.html. Visited 19.4.2021.

[7] Wim Roelandts. 15 Years of Innovation. *XCell*, 32:2, 1999.

[8] Investor Overview. https://investor.xilinx.com/static-files/2958145c-e3a2-456f-a461-bc9cba375af3. Visited 19.4.2021.