CHALMERS

Silent Integrated Electronics Car Sensor

Claes Thorén, Linus Klemets

2011-06-15

Abstract

A prototype of an integrated electronics car sensor has been developed. It is a device that locks a vehicle seat belt when it is tilted or horizontally accelerated more than normally expected for a vehicle, keeping the occupant safely in her seat. It uses an accelerometer as sensor, an FPGA for signal processing and a truck belt retractor containing an electromechanical actuator. A comparison between a mechanical car sensor and the prototype car sensor show that it is possible to get comparable performance and a more silent operation using the prototype. Further it is possible to automatically calibrate the prototype car sensor so that the sensor might be installed with an inclination relative to the horizontal plane, enabling development cost cutting being able to use a belt retractor model for several car models. A drawback is that this car sensor demands a power supply.

Acknowledgements

We would like to express our appreciation for some of the people who greatly contributed to this master thesis report. Yogen Patel started out with a vision about a new generation of car sensors that would raise the bar of seat belt safety equipment. He got in contact with us, asking if we could help him realize his vision. Without Mr. Patel, there would be no project to begin with. Peter Enoksson, examiner of the master thesis and great inspiration in technical matters, thanks for your dedication in our work. Inga-Lill Agardsson and the staff in Autolivs test laboratory, you put up with us despite or countless questions. Alexander Dymér was always helpful in matters concerning Bosch sensors. Lars Kollberg was very helpful in acquiring hardware and letting us use the electronics prototype lab. We also direct our sincere thanks to Gunnar Elgered, who made the electronics measurement lab available to us. Sven Knutsson took his time to answer our questions about digital hardware design. These are the reasons why we are indebted to you all.

Contents

List of Figures

List of Tables

Nomenclature

Actuator It is the moving part of the car sensor that locks the seat belt.

ADC Analog to Digital Converter

ASCII American Standard Code for Information Interchange

ASIC Application Specific Integrated Circuit

Belt retractor The spool where the seat belt is collected. The mechanical car sensor is mounted in the belt retractor.

Car sensor The part of the seat belt that senses movements in the car and locks the belt. The car sensor consist of both the sensor and the actuator.

CSB Chip Select Bit

DSP Digital Signal Processing. This unit processes the incoming data from the sensor and external signals. Depending on this information the DSP makes a decision if the actuator should be activated or not.

EMC ElectroMagnetic Compatibility

FPAA Field Programmable Analog Array

FPGA Field Programmable Gate Array

GUI Graphical User Interface is the graphical interface between the user and the prototype that realize the host interface.

Host interface An interface between the prototype and a user such as a technician that allows for readout of data and setting of prototype parameters.

I2C Inter Integrated Circuit is an interface standard for communication between integrated circuits. It is not used within this project.

LSB Least Significant Bit

Matlab Computer program used for mathematical calculus.

MSB Most Significant Bit

NRE Non-Recurring Engineering expense

ODR Output Data Rate

Onset Acceleration pattern for acceleration testbench [measured in g/s]

Payout Length of extracted belt after locking

PCB Printed Circuit Board

SPI Serial Peripheral Interface is an interface standard for accessing the accelerometer registers.

UART Universal Asynchronous Receiver/Transmitter is an communication standard used for communication between the FPGA and a host interface computer.

VHDL VLSI HDL - Where VLSI is short for Very Large Scale Integration and HDL for Hardware Description Language.

Chapter 1

Introduction

This master thesis project report explains the development of a novel car sensor. The project was issued by Autoliv, who saw an increasing demand for silent car sensors from their customers. Being the first in a series of thesis projects covering the novel car sensor, this project focus on the electrical utilization of parts handling sensing and signal processing.

1.1 Background

The automotive industry is subjected to safety regulations. A seat belt has to, among other regulations, lock the belt if the vehicle is accelerated in any horizontal direction or tilted over a certain threshold value. The purpose is to lock the belt in case of a collision or if the car rolls over. This is most often solved by the use of a mechanical car sensor. In such a device, an iron ball moves when it is exposed to acceleration in any form. This could be due to the vehicle accelerating in any direction by motor power, applied brakes or that the vehicle is exposed to an external force such as a collision. The car sensor can also sense if the vehicle is tilting by detecting the change of gravities pull on the sensor, making the iron ball move. The ball affects a mechanical arm as it moves to either side, resulting in locking the belt. Nowadays electric vehicles are getting more common and the quality demands are steadily increasing. The driving environment in an electric vehicle is fairly silent due to the electric motors reduced sound level compared to vehicles with combustion engines. This creates a problem when using mechanical car sensors. When the vehicle is in motion vibrations are generated, due to those vibrations, audible noise is created by the iron ball. A noiseless car sensor is therefore desirable.

Another drawback with the mechanical car sensors is that a new production tool for the car sensor must be made for each car model for calibration purposes. An integrated electronics car sensor may be able to be calibrated to fit into any car model, thus suitable for mass manufacturing.

1.2 Purpose

The purpose of this master thesis project is to evaluate the possibility of using integrated electronics to achieve a silent car sensor.

An second purpose is to constitute the cornerstone in a larger context of future thesis projects, all investigating different aspects of the car sensor discussed in this project.

1.3 Method

In order to meet the purpose, the answer to the following question had to be answered; Is it possible to make a silent car sensor by using integrated electronics? To investigate the answer to the question, a prototype of a car sensor with integrated electronics was made. The prototype car sensor was developed with focus on the intergated electronics parts. There were no restrictions on choice of solutions for the integrated electronics in the prototype. The development of the prototype started with a brainstorming meeting with people from different technical background, aiming at generating ideas. These ideas were further investigated in a litterature study. The possible choices were compared to each other and a solution was choosen. The different parts of the prototype were found by conducting a second survey aiming specifically at finding suitable components. The prototype were built and the digital hardware was designed. Tests were made on the prototype and analyzed.

1.4 Today's car sensors described

Today's mechanical car sensor is purely mechanical and it is depending on the gravity of the earth and acceleration of the vehicle. One example of such a car sensor can be seen in figure 1.1. The iron ball is set in motion during acceleration of the vehicle or when the vehicle is tilting. The shape of the bowl in which the iron ball sits determine the properties of the sensor. When the acceleration is high enough, the iron ball rolls out of the bowl and pushes the plastic arm up, locking a cog. The result will be the same if the vehicle is tilting enough for the gravity of the earth to move the iron ball. The cog is connected to the mechanism that is locking the seat belt.

Figure 1.1: Mechanical car sensor

1.5 Outline

The first chapter of the report is introducing the topic of car sensors and describes what the purpose of the project is. In the second chapter, the requirements of the car sensor is presented and explained. In the section “Choosing components for the prototype car sensor”, the design alternatives such as components are discussed and chosen. “The prototype” is a results section about how the prototype is designed and how it performs under testing. Closely connected to the prototype is the signal processing section. This section describe the vital signal processing parts of “The prototype”, that is the filtering and locking algorithms. An explanatory section on binary calculus is provided for better understanding of the challenges of VHDL design. In the conclusion, results are discussed and concluded and fields that needs further research is pointed out. Lastly, the project is summarized.

Chapter 2

Requirements

Presented in this section is a condensed version of the extensive mechanical car sensor requirement list used by Autoliv for commercial car sensors. All requirements that are not interesting to consider for an early stage prototype is omitted. Such requirements include, among others, corrosion and dust resilience. The result is a non application specific framework of requirements, specific enough to provide the necessary boundaries for an early stage prototype.

2.1 Requirements explained

The measure of how quick the belt is locked after applied stimuli, is commonly given in a derived metric. That is the length of extracted belt, called payout.The definition of applied stimuli is that, either the threshold value for tilt or acceleration is reached. When designing the digital hardware for the car sensor it is possible to get an estimate of how quick the design responds to input, that is the response time. The estimated response time is of course given in seconds and not in millimeters payout. That makes the derived metric somewhat impractical for comparative purposes. Therefor it is useful to be able to alternate between the two representations of response time.

Within this project the following two assumptions is used to simplify the models. Assume that a car in constant linear motion suddenly is caused to accelerate with linearly increasing acceleration. Assume that this change in speed will cause the seat belt to spool out the distance corresponding to the difference in distance the car has traveled with the added acceleration compared to the distance it would have traveled if the speed had been kept constant, that is the payout. These are the same assumptions used by Autoliv for testing car sensors. The lines in figure 2.1 show different ways of accelerating a test fixture in the car sensor test. The acceleration pattern of the fixture is called onset. Figure 2.1 contains a list of the different onsets that Autoliv use when testing car sensors. Onset 150 g/s reaches the target acceleration $A_{Thr} = 7\frac{m}{s^2} \approx 0.7g$ at 4 ms which hereafter will be called T_{onset}, thus $T_{onset} = 4ms$. It is a requirement

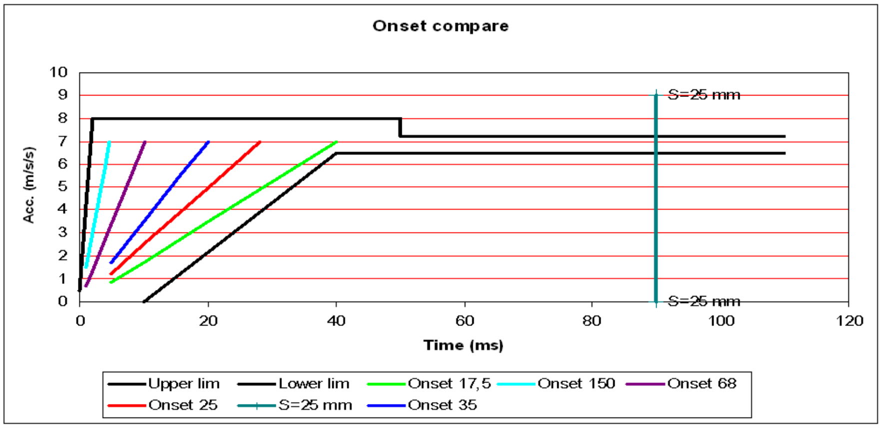

Figure 2.1: Lock-up with different onset

that the test is passed with this onset as well as an onset of 17.5 g/s. Figure 2.2 show how payout relates to time. At t = 0, the acceleration has just reached 0.7 g. On the y axis, the payout starts at approximately 11 mm and 14 mm for onset 17.5 g/s and 150 g/s respectively. That is due to the belt spooled out because of the distance travelled to reach 0.7 g added to the minimum payout for the belt retractor. The minimum payout of the belt retractor is caused by a distance between the locking teeth and housing in the locking mechanism. For 17.5 g/s onset, the payout is larger until approximately 26 ms, where the 150 g/s onset catch up and overtake the payout.

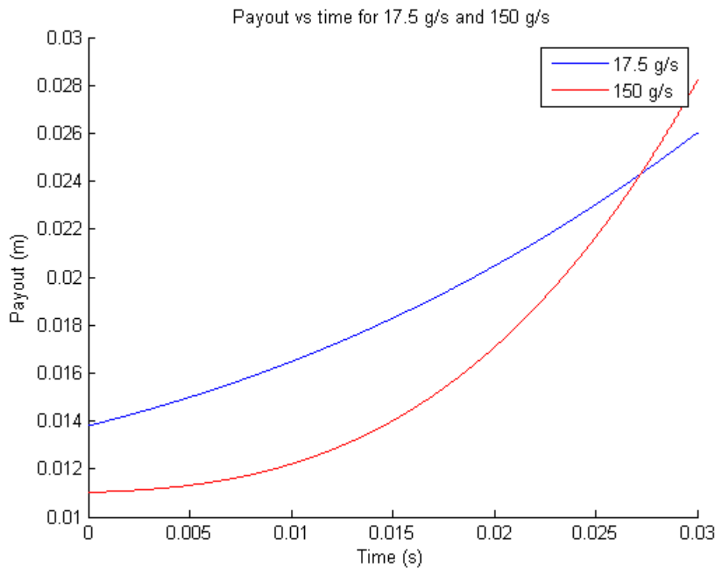

Figure 2.2: Payout vs time for 17.5 g/s and 150 g/s onset

An equation to translate between response time and belt payout is derived from equation 2.1, where s is distance, a is acceleration and t is time . From this, a simple translation scheme was derived, seen in equation 2.2.

$$s = \frac{1}{2}at^2 + v_0 t \tag{2.1}$$

$$D_{Payout} = \frac{1}{4}\left(\frac{A_{Thr}}{T_{onset}}\right) * (T_{onset} + T_D)^3 + D_{Payout,Min} \tag{2.2}$$

D_{Payout} is the length of belt that is extracted from the belt retractor at an acceleration that is linearly increasing from zero through the point A_{Thr} that is the threshold acceleration at which the belt is supposed to lock. T_{onset} is the onset time, that is the time it takes to reach A_{Thr} with an onset of 150 g/s. T_D is the total delay time for the car sensor, which can be divided into $T_{D,DHW}$ and $T_{D,Actuator}$. $T_{D,DHW}$ is the delay that is caused by the digital hardware, including the accelerometer data readout time. $T_{D,Actuator}$ is the time the actuator takes to lock when given a signal. $D_{Payout,Min} = 11mm$ is the minimum payout of the belt retractor.

$$T_D = \sqrt[3]{\frac{T_{onset} * D_{Payout} * 4}{A_{Thr}}} - T_{onset} \tag{2.3}$$

There is a requirement of acceleration and tilt at which the belt should lock and maximum allowed belt payout legislated by European and USA authorities that may be differently interpreted in different countries. The strictest requirement, and therefore most interesting, is the requirement in USA. There, the belt should lock at an acceleration above 0,7g and a tilt between 15º and 27º and unlock not later than at 12º tilt. The tilt is tested by tilting the car sensor in 8 directions. The maximum allowed belt payout at the acceleration test is 25 mm. Using equation 2.3 to calculate delay from these figures yields $T_D = 27.7ms$. This is the maximum allowed delay for the car sensor system. However, this project does not include development of a new actuator and the actuator used for the prototype of this project is not optimized for speed. Therefore Autoliv's set a target requirement for $T_{D,DHW}$ to maximum 1 ms and by doing so, assuming that an actuator delay of less than $T_D - T_{D,DHW} = 26.7ms$ is achievable.

For sensing the vehicles accelerations the sensor samples data. Ideally, the data would be sampled continuously from a sensor with infinitely high bandwidth. In that case, all changes in accelerations, no matter how quick, could be registered. Such a system is not realizable with digital electronics since discrete time is used which demands a sampling in some frequency. In order for the car sensor to provide a sufficiently quick response to stimuli, the car sensor systems must have a high enough bandwidth. Knowing the requirements for the systems response time, the bandwidth of the system can be calculated. A relation between bandwidth and system response time may be derived from the Nyquist theorem [2] (equation 2.4) and equation 2.5, giving equation 2.6.

$$F_{BW} = \frac{F_s}{2} \tag{2.4}$$

$$T_R = \frac{1}{F_{BW}} \tag{2.5}$$

$$F_S = \frac{2}{T_R} \tag{2.6}$$

$T_{D,DHW} = 1ms$ is the requirement of the accelerometer and digital hardware delay. Instead of seeing the time it takes to receive data from the sensor as a delay, it may be seen as a a response time. The sensors response time is the time it takes for the sensor to respond to a mechanical stimulus. $T_{R,acc}$, the sensors response time must be less than 1 ms, since $T_{D,DHW}$ also include the digital hardware delay, which gives a sampling frequency $F_{S,acc} > 2\,kHz$. Exactly how high $F_{S,acc}$ should be is determined by the delay of the digital hardware. While this is not a strict requirement for the sensors sampling frequency, it gives a ballpark figure, something to start with. Assuming that an sensor with sampling frequency well over 2 kHz is used, when reviewing the requirements during the digital hardware design, the clock frequency may be chosen to meet the actual requirement. The sampling frequency will hereafter be referred to as ODR (Output Data Rate) since that is commonly used and a more descriptive nomenclature ($F_s = ODR$). Given that the threshold acceleration is 0.7 g this must be within the measuring range of the sensor. According to [1], a measuring range of $\pm 2g$ is adequate for a sensor for automotive purposes such as measuring a cars movements in traffic.

As mentioned, the requirements for this project is basically a boiled down version of the requirements for the mechanical car sensor, meaning that all requirements not immediately crucial for an early prototype is removed. Since the integrated electronics car sensor contain electrical semiconductors that might be temperature sensitive, the requirement for temperature robustness was kept as is. The car sensor should be fully functional in a temperature range of -30ºC to +85ºC and be fully functional after storage in -40ºC to +100ºC.

2.2 Requirements summarized

Requirement	Value
Maximum belt payout at acceleration test	25mm
A_{Thr}(Acceleration threshold)	0.7g
T_{onset}(Time to reach A_{Thr})	40ms
$T_{D,DHW}$(Delay for digital hardware including sensor)	1ms
$F_{S,acc}$Accelerometer sampling frequency (ODR)	>2kHz
Minimum locking angle	15º
Maximum locking angle	27º
Minimum unlock angle	12º
Temperature (full function)	-30ºC to +85ºC
Temperature (storage)	-40ºC to +100ºC

Table 2.1: Requirements summarized

Chapter 3

Choosing components for the prototype car sensor

In this section different possible solutions for the prototype are presented. The section is divided according to the different parts of the prototype; Sensors, digital platform and actuators. Lastly, design choices are discussed and a solution is selected.

3.1 Sensors

The sensor used within this project is supposed to measure two different forms of movement, acceleration and tilt. Browsing the market for different solutions for sensing movements, accelerometers and gyroscope sensors were found. The gyroscope sensor (angular rate sensor) is detecting angular accelerations, that is rotational movements. Since tilt is a rotational movement, the gyroscope sensor would seemingly suffice for tilt sensing. But due to drift it needs to be recalibrated. That is, the signal processor needs to keep track of the rotations so that it know the actual tilt angle. There is no way for the gyroscope sensor to measure it's actual orientation. This is a problem since the slightest drift in the sensing would eventually lead to malfunction of the car sensor. The belt is also supposed to lock at horizontal accelerations of a certain magnitude, which is not a rotational movement. This would make a gyroscope sensor unsuitable for acceleration sensing. Accelerometers however, is most suitable for sensing accelerations. The use of both an accelerometer and a gyroscope sensor would possibly enable for self calibration of the gyroscope sensor. Then, reliable tilt measurements would be possible. In fact, accelerometers may also be used for sensing tilt. This is explained in section 5.2.2. Both accelerometers and gyroscope sensors come in both analog and digital readout versions. The choice of which alternative to use is above all depending on the device the sensor is interfacing with. If there is an analog to digital converter available or if the interfacing circuit is entirely analog, then a sensor with analog output is the natural choice. Such a sensor provide a continuous analog output. That however, does not imply an infinite bandwidth. The bandwidth of an analog sensor is limited by the mechanical properties which is, in the case of a commercial sensor, specified in the data sheet of the accelerometer [24]. A digital accelerometer is essentially an analog accelerometer with a built in (Analog to Digital Converter) that sample the analog voltage with a certain rate [25].

Model	Manufacturer	Range	LSB/g	Resolution	ODR
Min	requirements	±2g	10	5 bits	2kHz
SMB380	Bosch	±8g	64	10 bits	3kHz
BMA150	Bosch	±8g	64	10 bits	3kHz
BMA180	Bosch	±16g	512	14 bits	2.4kHz
MMA8451Q	Freescale	±8g	1024	14 bits	800Hz
MMA8452Q	Freescale	±8g	256	12 bits	800Hz
MMA8453Q	Freescale	±8g	64	10 bits	800Hz
AIS326DQ	ST Microelectronics	±6g	364	12 bits	2.56kHz
AIS328DQ	ST Microelectronics	±8g	256	16 bits	1kHz
LIS302DL	ST Microelectronics	±9g	14	16 bits	400Hz

Table 3.1: Comparison of digital accelerometers

Shown in table 3.1 is an excerpt from three different accelerometer manufacturers product portfolios. The row marked “Min requirements” contain the accelerometer requirements for this project. It is obvious that many of the accelerometers are unable to meet the ODR requirement. Looking at the measure range, all listed accelerometer seem to have a large enough range to meet the specification. Though it is not desirable to have a larger range than necessary, all listed accelerometers have the possibility to electronically choose a measuring range of ±2g, which is the required range. All listed accelerometers meet the requirements for temperature range, resolution and sensitivity (LSB/g). AIS326DQ from ST Microelectronics have a high ODR, 2.56 kHz, but investigation of the data sheet show that the bandwidth is merely 640 Hz which is less than the required 1 kHz that was shown in section 2.1.

3.2 Digital platform

Even though it is possible to implement an analog signal processor, that solution was never considered. Such a signal processor would not meet Autolivs needs for reconfigurability. To adjust the behavior of such a signal processor, the values of the components would need to be changed [3], provided that not a bulky and expensive FPAA (Field Programmable Analog Array) is used.

Looking at digital electronic alternatives for handling the signal processing and and connecting the diffrent electrical components together for the car sensor, three different alternatives stand out; ASIC, FPGA and microprocessor.

Microprocessors are quickly programmed to interface with e.g. an accelerometer. They are reconfigurable and fairly cheap. Microprocessors run sequential programs where one instruction is processed at a time [4]. This means that it is not possible to process data in parallel for parallel algorithms. This would rule out an architecture where both tilt and acceleration are computed simultaneously without interruptions. ASIC's offer high architectural flexibility and low price per unit but high NRE (Non-Recurring Engineering expense) [5]. ASIC's are designed using VHDL. FPGA's are configurable digital electronic devices that are flexible and easy to implement. Their reconfigurability makes them suitable for prototyping devices. FPGA's are configured using VHDL which enable large architectural flexibility in terms of processing, which may be done in serial, parallel and even asynchronous processes. FPGA's are rather expensive [6].

3.3 Actuators

Since actuator design was not the focus for this project, different actuator topologies has not been thoroughly investigated. However, an example of two possibly considerable actuation principles are presented briefly to contrast the conventionally used electromagnetic actuator.

3.3.1 Piezoelectric actuators

A Piezoelectric material produce a voltage when subjected to stress or reversed, deflect when a voltage is applied [7]. This property make it useful for actuator applications. Three basic piezoelectric actuators are stacks, benders and motors. The stack actuator is a device built up by piezoelectric discs stacked in a sandwich construction. A voltage source is applied to each layer of piezoelectric material causing the multilayered stack to deflect about 0.1-0.2% of the total stack height [8].

Piezoelectric benders may be constructed in different ways. A straight, thin sheet beam attached in one side is able to move freely in the other side. A voltage applied between the upper and lower side of the beam cause one side to expand and the other side to contract. This results in a deflection of the bender. If the voltage is alternated the beam will oscillate.

A piezoelectric motor is either a linear or a rotary motor. There are several different types of linear and rotary motors, ultrasonic motors is one of them. In the ultrasonic motor, piezoelectric (ceramic) material is attached to a stator. A combination of longitudinal and flexural ultrasonic vibration modes is used to create an elliptic micro-motion of surface points. This elliptic micro-motion is transferred to a slider that begins to move [9].

Since these kinds of motors are working close to the stators resonance frequency, the resonance frequency needs to be well known. This results in high production accuracy needs and sensitivity to parameters such as temperature [10]. Piezoelectric devices needs a high supply voltage relative to the voltage of a car battery. This means that some kind of voltage transformation circuitry is needed [11].

3.3.2 SMA actuators

Shape memory alloy (SMA) is a group of materials that can change shape when temperature is changed [12]. That is because SMA's have different crystal structures for high and low temperature. The movement in the material from changes of temperature may be used for actuation. In SMA actuators triggered by an electric signal, a current passing through the SMA cause the temperature to rise. For this application a high resistive SMA is preferable, e.g. Ni-Ti alloys [12]. The Ni-Ti alloy has a maximum elongation of about 40-50% [13]. Since heating is done actively by passing an electrical current through the SMA, heating the SMA is quicker than cooling it of. This may cause problems when using it as a fast actuator since the actuator may recover slow from actuation [14]. Fast acting Ni-Ti SMA's also have a low transformation temperature, working in an approximate temperature span of -30 to maximum +120 degrees centigrade. The car sensor need to be able to operate in a temperature span from -30ºC to +85ºC.

3.4 Design choices

Developing an ASIC within the scope of a master thesis project is unrealistic since it would require a much larger budget of both time and money than what is feasible. An ASIC would probably still be a good choice for large scale production since it enable parallel computing architectures and the price per unit would be low compared to the FPGA [5]. The FPGA is, in this case very useful for prototyping, since it is possible to implement parallel computing architectures and the development of an FPGA somewhat reflects the development of an ASIC. That is because VHDL is used in both cases, thus giving a rough estimate about how quickly an ASIC would make the computations needed. The choice of using an FPGA also affected the choice of sensor. The FPGA does not (normally) have an ADC which rules out the use of a analog output sensor. Striving for a simple and compact solution there seemed to be no point in using both an accelerometer and a gyroscope sensor. It would greatly add to the complexity of the sensing algorithms but has no obvious accuracy advantage, considering that acceleration and tilt is tested separately (see section 4.5). The choice fell onto the accelerometer with digital output from Bosch called BMA180. The reasons for choosing this particular sensor were; The output data rate is high, 2.4 kHz, the data sheet was readily available on the manufacturers web page and there was affordable pre-soldered break out boards available.

The choice for actuator principle was based on availability and how it would fit the prototype. Piezoelectric or SMA actuators might, or might not, be good choices for fast and reliable actuation. Using any of these actuator principles would take considerable time to make an actuator fit for the prototype. When

Autoliv presented the electromechanical truck belt retractor, this was chosen because it is easily incorporated into the prototype. The choice of an electromechanical actuator was not meant to reflect on what would be the best choice for the final product since that needs more investigation and will be dealt with in detail in the next master thesis concerning this car sensor actuator system.

Chapter 4

The Prototype

Within the scope of this project a prototype was made to verify that an integrated electronics car sensor is possible to create. However, the prototype was not made small enough for containment within a belt retractor. That would require considerably larger developing efforts not possible within the scope of this project. The prototype is designed for testing applicable car sensor test cases within the standard Autoliv test routine for belt retractors. For this, available test equipment at Autoliv site Vårgårda was used. The use of these test rigs enable benchmarking of the prototype against the specification and comparisons to other car sensors.

The prototype consist essentially of four hardware modules, an accelerometer, a digital platform an actuator and a computer host interface. The modules are chosen so that the potential of the integrated electronics car sensor might be investigated. Some components that are used have higher performance than required. This cause the DSP (Digital Signal Processing) algorithms to determine the performance to a larger extent than one would get aiming for a low cost prototype.

4.1 Accelerometer

The sensor is a three axis accelerometer with a 14 bit resolution manufactured by Bosch (model: BMA 180). The prototype is using only 8 of 14 bits in the calculations, this gives a resolution of 0.0157 g at the ± 2 g range. However, using the calculations explained in section 5.2 the error and uncertantiy of the resolution will be magnified [16] and have a greater impact in the algorithm. The sensor is a high end accelerometer with performance exceeding the requirements. The accelerometer has built in axis calibration algorithms. This supposedly makes it possible to install the accelerometer with an angle and calibrate it to seem horizontal. There are also a number of built in digital low pass filters to choose from. These filters are useful when developing the prototype. The accelerometer break out board is mounted on the fixture holding the belt retractor for better stability and is connected to the FPGA via a flat cable.

4.2 Digital platform

The digital hardware is realized on an FPGA mounted on a development board. The FPGA is a Spartan 6 from Xilinx and the development board is an LX9 Microboard from Avnet. For cost effective production of a car sensor, an ASIC and not an FPGA would be the choice of topology. The reason for using an FPGA for the prototype is that it is easy to implement and VHDL is used for the FPGA design as well as for ASIC design. This makes it possible to get delay estimates from the VHDL implementation, presented in section 4.2.4.

The architecture of the digital hardware is divided into a number of blocks, each taking care of their specific task. These blocks are shown in figure 4.1. The names within brackets in the headings of the following subsections correspond to the blocks in figure 4.1. The filters are not implemented and just bi-passed to simplify for future implementation. There was not enough time to implement the filters within the scope of this project. Instead, the built in filters in the accelerometer are used. The block within the DSP called COMB is there to show that additional algorithms, e.g. sensing combinations of tilt and acceleration, might be added. ACC is the block representing the accelerometer sensor.

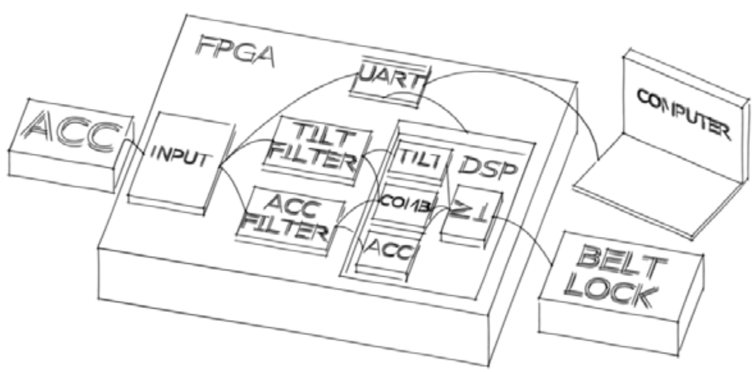

Figure 4.1: Prototype architecture

4.2.1 Accelerometer interface (INPUT)

This section will explain more in detail how the accelerometer interface is built up and how the communication with the accelerometer is done. Also some explanation on certain design decisions will be mentioned. The INPUT block is utilizing an SPI (Serial Peripheral Interface) that communicates with the accelerometer. The interface may read an arbitrary number of addresses from the accelerometer register and write to a single address in the accelerometer register. The interface block have different outputs for accelerometer data and other data. This architecture minimize the risk of malicious data entering the DSP which may cause unreliable signal processing. However, this should not be a problem since the system, when in operating mode, will only read accelerometer data. The only time anything else will be read from the accelerometer is when the car sensor is being configured.

There are two ways to communicate with the accelerometer, either with the use of I^2C protocol or the SPI protocol. In this project the SPI protocol was chosen as this fulfilled the needs and the authers were familiar with the protocol. To start the SPI communication with the accelerometer, the chip select bit (CSB) on the device needs to be set to low. It is not allowed to change any data on the SPI before this, since this could unintentionally trigger a start condition for the I^2C protocol [25]. A timing diagram explaining how communication should work is seen in figure 4.2.

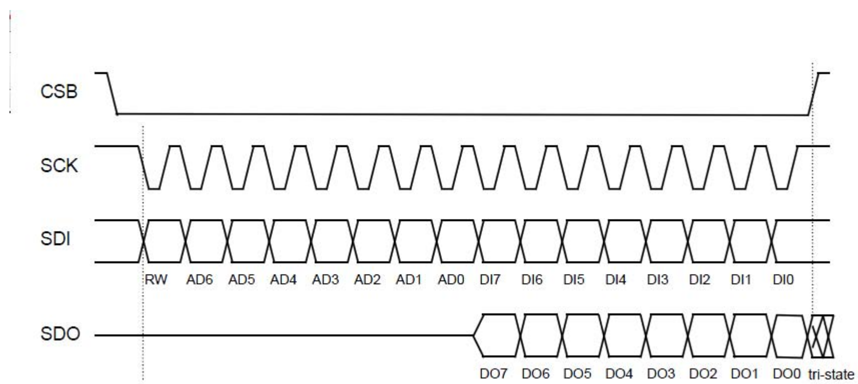

Figure 4.2: SPI timing diagram

The first bit sent to the accelerometer is the write/read bit. If this bit is set to '0' it will write data to the specified memory and if it is set to '1' this will indicate a read operation. The next 7 bits are the memory address. During a write operation the next 8 bits will be the data to be written to the specified memory. If the CSB continue to be low after those 8 bits it will start a new read/write operation. If it instead is a read operation it will only read the first 8 bits, the read/write bit and the 7 for the memory address, after this it will ignore any incoming data. Contrary to the write operation the read operation will continue to read as long as CSB is low. The address sent indicates only where it should start reading. For example if it starts reading at memory 02h it will continue with 03h, 04h and so on until CSB is set to high. This function is used in the design since the data for x,y and z axes are located after each other in the register and also consist of two bytes each. It should be noted that the accelerometer is sending the data in a register with MSB first down to the LSB. Since the data for each axis consists of two bytes and the byte containing the least significant bits will be read first, the data received need to be rearranged.

The accelerometer interface has both the serial system clock and the slower SPI clock as input. It uses an input bus from the UART that has a length of 16 bits which is used for deciding which actions to take. The first 8 bits consist of the write/read bit and the memory address and the remaining 8 bits contain the data that should be written (if it is just a read command those bits will be ignored).

Since the idea is to always read data from the bus and sending the corresponding data to the accelerometer it was necessary to add a bus operation that prevents the INPUT from doing any operations. If the INPUT block receives the address 00h it will stop doing any operations until the address has changed. The reason for this addition is to avoid certain bugs and errors during configuration of the accelerometer.

Some improvements that could be done in the accelerometer interface is to use the interrupt functionality of the accelerometer. Every time the accelerometer values has been updated the interrupt pin goes high indicating that it is ready for a new readout. For the moment this has been internally hard wired to '1' and a new read operation starts immediately after the last one has finished. This is not optimal in resource use nor in speed as unnecessary duplicate readouts are done but is very easy to implement. Another improvement would be to allow to read from all addresses. At the moment it is not possible to read out the LSB address of the x axis since this is used as an "activation code" to start reading all the accelerometer values. The reason for this is that x,y and z values are contiguous in the address memory of the accelerometer, which was mentioned earlier and that it is possible to read out all the accelerometer values in a row without having to restart the read operation for each new memory address.

4.2.2 Signal processor (DSP)

The signal processor implements two parallel processes that each realize an algorithm for locking the belt. The algorithms are fed with accelerometer data for all three axes. One algorithm handle the case of the car tilting, the other algorithm handle the cars accelerations. A decision from one of the processes to lock the belt will cause the belt to lock, independent of the other process. For further development of the car sensor prototype, more parallel processes might be added to realize more algorithms for increased safety.

There are six parameters that determine when the belt is locked and unlocked. These parameters are set via the Host interface GUI that is explained in section 4.4. One of these parameter sets the tilt angle at which the car sensor should lock and one parameter sets at which tilt angle the seat belt should unlock. A pair of parameters determine the distinction between tilt and acceleration sensing by setting a range of the absolute value of the resultant vector of acceleration data in which the accelerometer is considered tilting. The principles of function for the algorithms are further explained in section 5.2.

4.2.3 Host interface - in FPGA (UART)

The UART contain a state machine that governs the functions of all digital hardware seen in figure 4.1. It is in that sense the controller unit for priority decisions of the entire car sensor. The UART block also handles all communication with the host interface computer. It listens to the computer for commands and send requested data to the computer. The computer may request to get the data from any address in the accelerometer register with one exception. The register byte containing the LSB of the x axis accelerometer value is reserved for accelerometer data readout, meaning that trying to read from just that address will automatically result in the readout of all accelerometer data as mention in section 4.2.1. This data will also not be available on the bus that the UART will try to read from. The computer may also request to write any data to any address in the accelerometer register. These functions are used to monitor and manipulate the contents of the accelerometer data registers which control the accelerometers behavior. The host interface computer may also change the parameters in the DSP.

Since multiple bytes are sent and received over the UART, both input and output buffers are necessary. The output buffer stores a number of bytes that should be sent to the computer. Then, the UART may send one byte at a time picking them from the buffer. For the input buffer, each received byte is placed in the buffer until all bytes are received. Then all of them are made available to the recipient block on the FPGA.

Accelerometer data is sent from the FPGA to the computer using one hexadecimal coded digit at a time. This means that the FPGA needs to send two bytes for each byte of data the FPGA wants to send, as shown in figure 4.3. The reason for this solution is that data gets a reserved set of ASCII characters, making it possible to neatly print the data to a text file. If the data would contain all possible ASCII characters, there would eventually appear unwanted ASCII characters in the text file, such as line feeds, making the text file hard to read for the user as well as for a computer program e.g. Matlab.

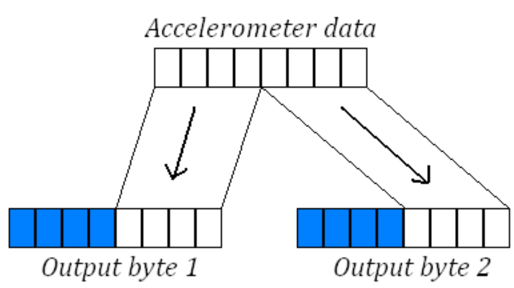

Figure 4.3: Hexadecimal coded output data

4.2.4 Estimations

Studying the architecture of the digital platform, it is possible to extract the response time. The SPI clock frequency for the accelerometer is 10 MHz, that is a clock period of 0.1 µs. This is the pace at which it is possible to read and write data from and to the accelerometer. It should not be mistaken with the updating rate of the accelerometer data which is much slower (2.4 kHz $\rightarrow$ 417 µs). The interfacing INPUT block will start the readout of the accelerometer data by sending the address for the x-axis LSB data as the starting point. This will take 9 clock cycles to complete, one for the initiation where the interrupt bit is detected and another one for the "read/write" bit and finally 7 for the address bits. The accelerometer will at this point start sending the x-axis data. There will be 6 addresses that need to be read since each axis consist of two 8 bits addresses. However since the addresses are consecutive, no more address data needs to be sent to the accelerometer, reducing the number of clock cycles needed. The total number of clock cycles for the INPUT interface is 6*8 for all the acceleration data and another 9 for the x-axis starting point address giving a total of 57 clock cycles or 5.7 µs. The DSP add a delay of one 100 MHz clock cycle, that is an extra 10 ns which is negligible. That gives a total delay $T_{D,DHW} = T_{D,acc} + T_{D,DHW} =$ 417µs+5.7µs=422.7µs.

4.3 Actuator

The belt retractor used for the prototype is a truck belt retractor with electronically controllable locking, shown in figure 4.4. This type of belt retractor is used in trucks with suspension seats. In such vehicles, the belt retractor and the car sensor is separated from each other. The belt retractor is mounted on the seat and the car sensor is mounted on the chassis of the vehicle. Even though the truck belt retractor does not provide as quick locking function as more sophisticated belt retractors, it is still useful for the prototype since the possibility of controlling the locking mechanism electronically makes it easy to use as it is. The belt locking output signal from the FPGA is fed to a power amplifier driving the belt retractor lock mechanism, shown in figure 4.5. Since the FPGA output signal is active low, the belt will be locked when the output signal is low and unlocked when the output is high.

Figure 4.4: Truck belt retractor used for the prototype

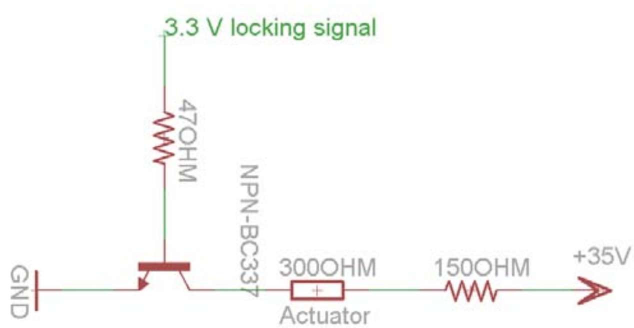

Figure 4.5: Actuator power amplifier schematic

4.4 Host interface - in computer

A graphical user interface (hereafter referred to as GUI) was developed for the prototype system both for help with the development and testing but also as an example of how the human interaction with the finalized product could look like. The platform was done in Matlab because of its mathematical strength and the inbuilt support for graphical display. The GUI can display the readouts of the accelerometer raw data in a form that show the different axes values as can be seen in figure 4.6. This is only for development purpose to show that the prototype system is operational and working as intended.

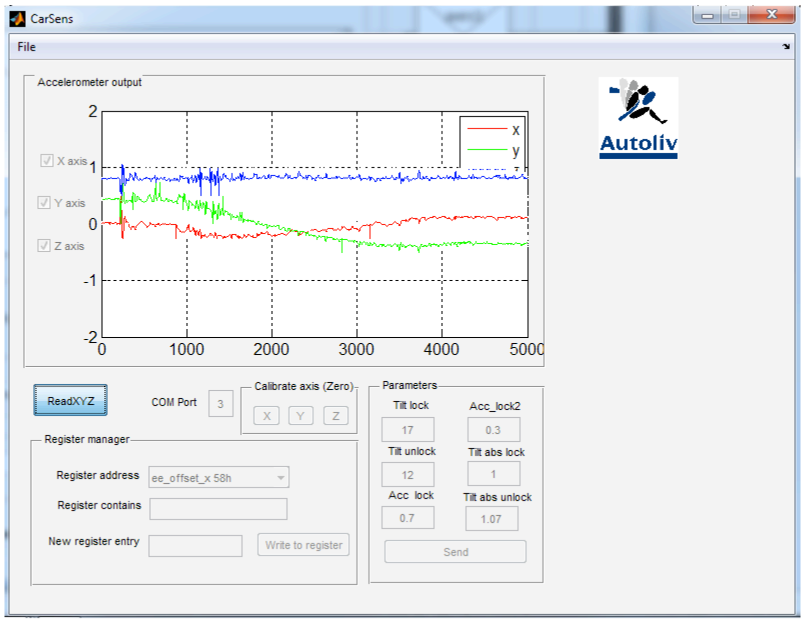

Figure 4.6: Graphical user interface for the prototype system

It is also possible write or read a specific register address from the accelerometer to manually reconfigure certain attributes. This could be for example to activate built-in filters, make certain some attributes are correct or recalibrate an axis. Since the calibration take a number of steps to complete, buttons have been added to run those scripts. The parameter fields in figure 4.6 is used for tuning the DSP's behavior. Those parameters are used in the algorithms in section 5.2. The GUI is required for the prototype system to work since it does not have any predefined parameters built in and the FPGA memory used is volatile. Also, to synchronize the data stream with the GUI, the prototype system requires an activation command from the GUI to start reading the x, y and z values. This is done when the ReadXYZ button is toggled.

4.5 Testing of prototype

Applicable parts of the test procedures that Autoliv normally use to test their car sensors was used to verify the functionality of the prototype. Only the test steps for the tilt and acceleration algorithms are used, omitting e.g. endurance tests. This makes it possible to compare test result of the prototype with test results from other car sensors. The test method is proven to be reliable, thus eliminating the need for developing a specific test method for the prototype, thus saving a lot of time.

Figure 4.7 shows spike shaped noise observed in the GUI's display of outputs from the prototype car sensor.

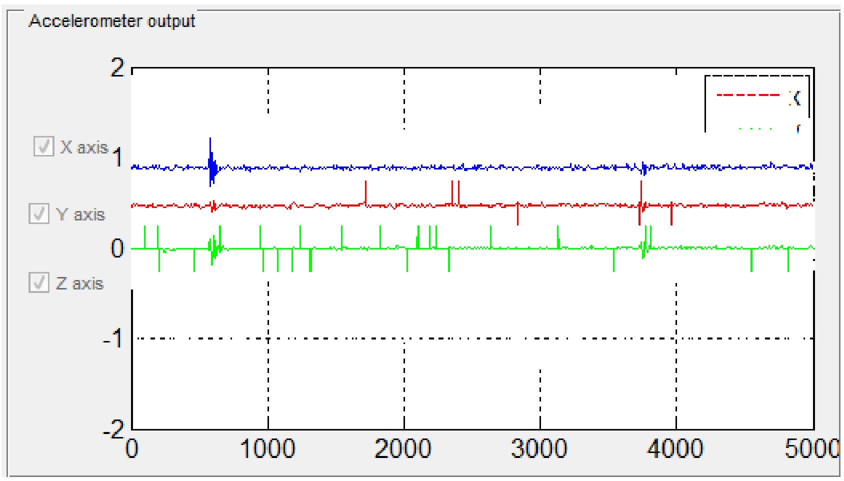

Figure 4.7: Accelerometer readouts with spikes

It was observed that the noise seemed to be of the same amplitude each time and after some test this was confirmed. The amplitude of the noise has been observed to about 0.24 g. Another observation was that the noise only seemed to occur at certain values. Due to the strange nature of the noise and the considered magnitude it was concluded that the noise was unnatural and somehow generated by either the accelerometer or the digital platform. Several tests and reconfigurations were done to try to detect or remove the noise resulting in some observations. The noise was of the same magnitude percentage (12%) regardless of the range of the accelerometer, that is 0.24 g at the range of ± 2 g and 1.9 g at ± 16 g. The noise only occurred at certain values, for example at 0 g, ± 1 g. There were considerable less noise when the accelerometer had a lower cutoff frequency but this also results in a lower bandwidth [25].

4.5.1 Using Autolivs test bench for tilt

This section describes the test method for verifying the tilt algorithm. The belt retractor with the car sensor is placed in a fixture on a table that is possible to rotate with three degrees of freedom, seen in figure 4.8.

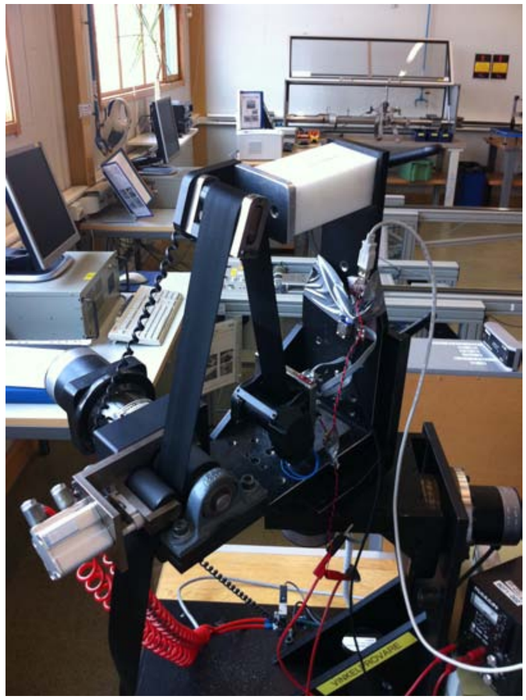

Figure 4.8: Tilt test rig

The seat belt is passed over a pulley with a sensor that senses when the belt is locked, down to a set of pulleys that may spool belt in or out of the retractor. A computer is used for initiating the test and monitoring the results.

Here the steps of the tilt test are listed:

1. The test is started, the belt is continuously spooled out in a slow pace ($\approx \frac{10cm}{second}$). At the same time, the fixture with the belt retractor and pulleys is slowly tilted ($\frac{2^{\circ}}{second}$).

2. When the belt is locked, the computer takes a note of the angle.

3. The fixture is then tilted back up with a rate of tilt of $\frac{2^{\circ}}{second}$ while a small amount of belt ($\approx$ 5 cm) is spooled back into the retractor and then back out to test if the belt still is locked by the car sensor. This step is repeated until the belt is unlocked by the car sensor and the computer notes the angle at which the belt was unlocked.

4. The belt is spooled back into the retractor so that it contains the same amount of belt as it did at the beginning of the test sequence. Then the fixture is pivoted according to the pivot angle column in the test sequence description in table 4.1. This is to test at which tilt angles the car sensor locks and unlocks when tilted in different directions.

5. Step 1-4 is repeated until all of the eight pivot angles are tested. Then the test sequence is finished.

Q: What is the reason for spooling some belt back into the retractor in test step 3?
A: Once the belt is locked, even if the car sensor is not locking the belt, it will not unlock unless the tension on the belt is released. This is something that most people have experienced when trying to put the seat belt on to quickly so that it locks, thus having to release the tension on the belt in order to be able to pull out on the belt again. Locking the belt by pulling to quickly on it is however the effect of the bandacc sensor. This sensor locks the belt if it is extracted with an acceleration over a certain threshold. Both the car sensor and the bandacc sensor use the same locking mechanism for locking the belt.

Figure 4.9: Tilt adjustment for tilt test bench

Presented in table 4.1 are the results of an actual test sequence from the prototype car sensor. The columns "Lockmin" and "Lockmax" denotes the lowest and highest allowed angle respectively, at which the car sensor may lock in order to pass the test. These are the angle requirements mentioned in the requirements section. The columns "Lockangle" and "Unlockangle" list the angles at which the car sensor locked and un-locked the belt respectively.

Pivot angle	Lockmin	Lockmax	Unlockmin	Lockangle	Unlockangle
0º	15º	27º	12º	18.1º	15.0º
45º	15º	27º	12º	17.2º	13.5º
90º	15º	27º	12º	17.8º	15.3º
135º	15º	27º	12º	17.6º	13.9º
180º	15º	27º	12º	19.1º	15.5º
225º	15º	27º	12º	19.9º	16.3º
270º	15º	27º	12º	19.3º	15.7º
315º	15º	27º	12º	19.1º	15.4º

Table 4.1: Results from a tilt test of the prototype car sensor with locking angle set to 19º and unlock angle set to 15º

In order for the test results to be reliable and repeatable the test setup must be carefully performed. Therefore the test setup was done according to figure 4.10 that is further explained in the following numbered list. The accelerometer output is monitored via the GUI. The accelerometers built in 10 Hz low pass filter is used for filtering. A digital inclinometer instrument (Mitutoyo, Digital Protractor Pro 360; calibrated twice annualy, resolution 0.1º) is used to measure the angle of inclination of the fixture.

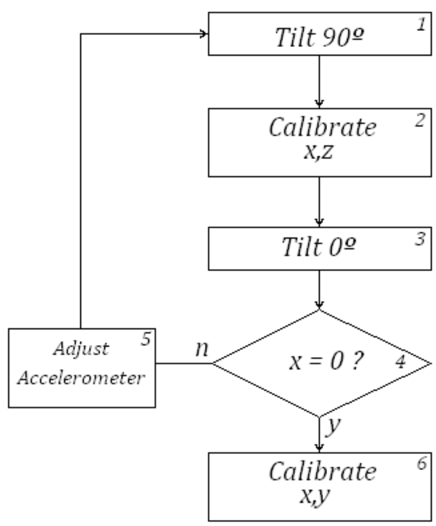

Figure 4.10: Calibration sequence

Calibration sequence in figure 4.10 described in six steps with numbers corresponding to the numbers in the figure:

1. Tilt the fixture 90º using the tilt adjustment, seen in figure 4.9, so that the x and z axes are in zero g orientation while y is in -1 g orientation.

2. Calibrate both the x and z axis and check that the output from both x and z is zero g.

3. Tilt the fixture to $0^{\underline{o}}$ using the tilt adjustment, so that x and y axis is in zero g orientation while z is in -1 g orientation. Since the fixture is tilted around the x axis, x is still oriented in zero g orientation. Therefor, the output from x still should be zero.

4. If x is zero, calibrate both the x and y axis and check that the output from both x and y is zero g.

5. If x is not zero, the accelerometers placement is adjusted so that the x axis is more aligned with the axis of tilt. Then start over on step 1.

6. Start the test sequence

The test sequence was run three times to get an indication about the systematic and stochastic deviations from the expected values. A collection of lock angle data from three tilt tests with eight different pivot angles in each test gave a variance of 1.19. When looking at a certain pivot angle, the lock angle had the same tilt angle at all three tests, suggesting that the stochastic variations were small. Systematic variations for different pivot angles was the source of the variance.

4.5.2 Using Autolivs test bench for acceleration

Autoliv have a test bench for controlling that the car sensors meet the requirements for locking at accelerations. In the test bench, seen in figure 4.11, the belt retractor is mounted on a movable fixture. The belt is directed trough a set of pulleys that sense if the belt is locked, and then fastened. The movable fixture may pivot in the horizontal plane in order to test the car sensors acceleration sensitivity in different directions. A computer is used for controlling the acceleration of the moving fixture as well as noting when the belt locks. The characteristic acceleration of the fixture, called onset, may be chosen according to figure 2.1. The requirement dictates that the threshold acceleration 0.7 g should be reached in 40 ms, that is an onset of 17.5 g/s.

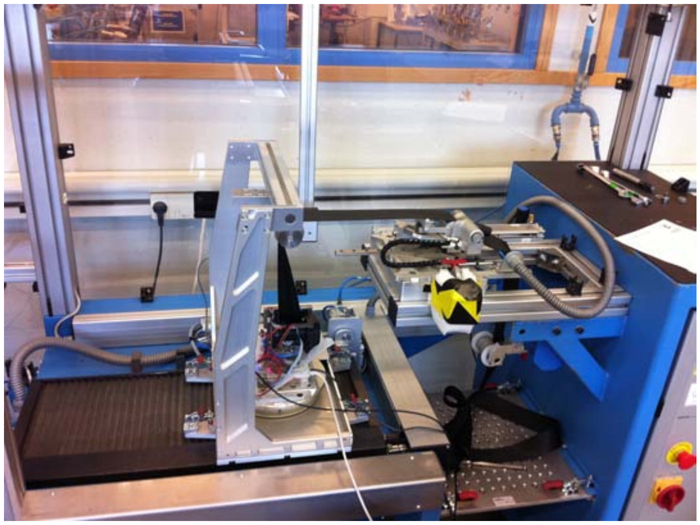

Figure 4.11: Acceleration test bench

The aim for this test was to see how the payout varies with filter cutoff frequency and onset. The aim was also to see if the sensitive tilt algorithm would erroneously lock the belt at horizontal accelerations. Since separate filters for the tilt and acceleration algorithm are not implemented in the prototype, the test was designed to investigate how the prototype would work if these filters had been present. This was done by alternating between the different built in filters of the accelerometer and by letting only the acceleration algorithm or acceleration and tilt together decide when to lock the belt. At first, the accelerometer was mounted on the belt retractor. The surface on which it was mounted did not provide a stable enough base, as the accelerometer tended to move slightly during tests. Therefore, the accelerometer seen as the small chip marked with a ring in figure 4.12, was mounted on the fixture instead. Since the fixture in the acceleration test bench is not tiltable, it was difficult to get an exact calibration of the z axis. The accelerometer had to be dismounted from the horizontal top of the fixture and placed on the vertical side of the fixture, where the x and z axes where calibrated. The accelerometer was mounted back onto the horizontal top of the fixture where x still should be in 0 g orientation. Then, both the x and y axes were calibrated.

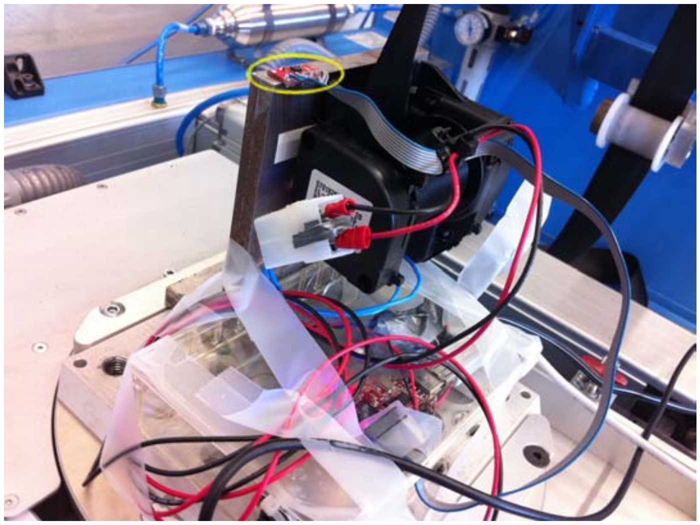

Figure 4.12: Acceleration test setup

A series of tests were made with both 17.5 g/s and 150 g/s onset. The acceleration was incrementally increased with 0.1 g at a time. Starting at 0.5 g and ending at 0.8 g. Each step was ran four times for each angle. Five different angles were used in the tests; 0º, 15º, 45º, 90º, 120º. According to Autolivs test system, a belt lock with a payout larger than 50 mm is not considered to be a belt lock. Figures 4.13 and 4.14 are excerpts from such test, they contain only the 0.7 g acceleration step. These test cases contain no erroneous locking payouts at other accelerations except payouts above the 50 mm threshold.

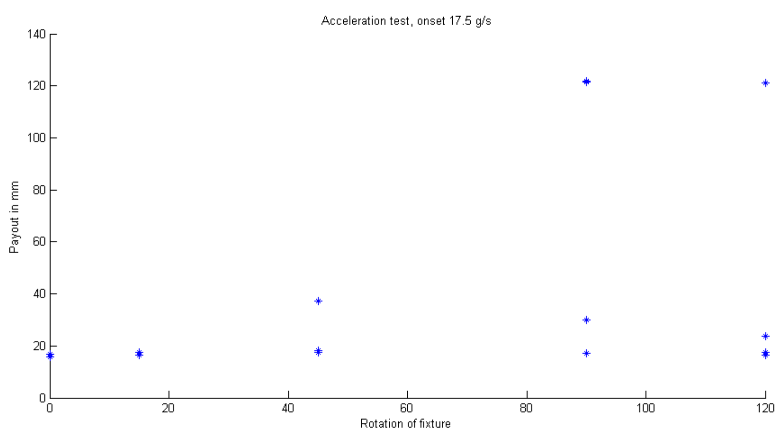

Figure 4.13: Lock-up with 0.7 g, onset 17.5 g/s and 75 Hz cutoff frequency

Figure 4.13 shows a test with onset 17.5 g/s, filtered with a bandpass filter with 75 Hz cutoff frequency. In the figure, payout varies with rotation of the test fixture. The most interesting aspect of these results is perhaps that the prototype car sensor performs rather well despite the low cutoff frequency. When the fixture is rotated 90º the prototype display the worst performance within the test sequence. The prototype should theoretically perform identical to 0º rotation since the accelerometers x and y axis are equally sensitive. The accelerometer PCB (Printed Circuit Board) was attached to the fixture using a double-coated adhesive tape. The accelerometer wiring is rather stiff and heavy compared to the accelerometer PCB. This combination could possibly and quite probably cause the PCB tape to flex when accelerated so that the weight from the wiring bends it. This would cause bad readouts from the accelerometer and possibly impulses which would only partially be filtered out with the 75 Hz filter. This theory is further supported by a test sequence where an even lower cutoff frequency is used, 10 Hz. In that test, the largest payout is also found at 90º rotation but with no payouts larger than 30.9 mm. That is still more than the required 25 mm though. Due to the possible presence of the noise discussed in section 4.5, that would affect even the z axis, the algorithm for acceleration was changed to having z fixed to one. Theoretically, z should be one during the entire test. There were no obvious difference to the test results after this change.

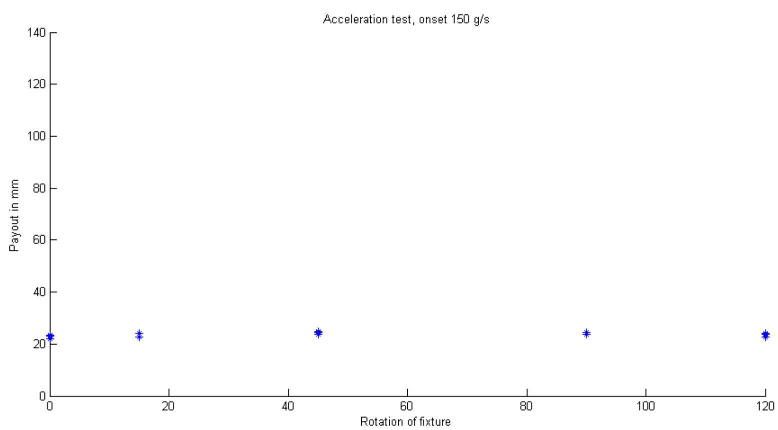

Figure 4.14: Lock-up with 0.7 g, onset 150 g/s and 10 Hz cutoff frequency

Figure 4.14 show a test sequence that was within the payout requirement. 10 Hz cutoff frequency was used and 150 g/s onset. This show that even when a cutoff frequency used for the tilt algorithm and the largest onset used by Autoliv, it is possible to get reasonable results in an acceleration test. The payout has small variation but is very close to the requirement of 25 mm. The closest value is 24.9 mm and the value with largest margin to the requirement is 22.1 mm.

To look specifically at interesting observations, other test were also conducted. Such a test could be seen in table 4.2 where the prototype car sensor seems to respond more assertive to the smaller onset of 17.5 g/s, having much smaller variance compared to the 150 g/s onset. The payout however, is smaller in the 150 g/s onset case which contradict the behavior predicted by equation 2.2. For the presented 150 g/s test sequence, the values are grouped in five payouts ranging from 10.8 mm to 11 mm and five payouts ranging from 17.1 mm to 17.3 mm, this is the reason why the variance is so large. The explaination to why the car sensor displays such a behavior might be found in the mechanical properties of the belt retractor. The actuator locks a cog with a number of teeth. Comparing the payout from the case when the actuator tip just miss a cog wheel tooth with the case were the actuator tip hits the cog wheel tooth imediatly at actuation, the difference might be up to 6 mm. This might be a reason for the large variance.

Onset	Mean payout	Median payout	Min	Max	Payout variance
150 g/s	14.06 mm	14.05 mm	10.8 mm	17.3 mm	11.24
17.5 g/s	17.45 mm	17.5 mm	17.2 mm	17.5 mm	0.01

Table 4.2: Comparing performance of 1200 Hz cutoff frequency filtered signal with onset 17.5 g/s and 150 g/s for 0^{o} rotation of fixture based on ten repetitions

At 0.6 g acceleration, the belt should not lock. In the tests, a higher amount of these unwanted belt locks were observed when a high cutoff frequency was used compared to when a low cutoff frequency was used. In fact no unwanted belt locks with payout smaller than 50 mm was observed with the 75 Hz low pass filter in contrast to almost 50% with the 1200 Hz low pass filter. This is most likely due to vibrations in the acceleration test bench causing the belt to lock, when the vibrations are not filtered out with a the low pass filter. The lower the filters cutoff frequency is, the more vibrations will be filtered out.

A comparison between a mechanical car sensor and the prototype car sensor is shown in table 4.3. Even though the prototype car sensor have a larger variance, the results are comparable.

Sensor	Mean payout	Median payout	Min	Max	Payout variance
Prototype	14.06	14.05	10.8 mm	17.3 mm	11.24
Mechanical	14.39	14.5	12.7 mm	16.5 mm	1.7

Table 4.3: Comparing performance of the prototype car sensor and a mechanical car sensor at 17.5 g/s onset. The prototype car sensor use the 1200 Hz cutoff frequency, the rotation of the fixture is 0^{o}. The values are based on ten and 25 repetitions for the prototype and the mechanical car sensor respectively.

Chapter 5

Signal processing

The car sensor needs an algorithm to decide when to lock the seat belt, or actually at least two algorithms. One algorithm is used for determining when the car is tilted over a certain angle, the other algorithm evaluates accelerations.

5.1 Filtering

A car in motion picks up vibrations from the surface on which it is traveling. This will be picked up by the accelerometer as noise that will impair the signal processors ability to accurately interpret the acceleration data. The influence of this noise source might be suppressed with the use of filters. That is, removing the high frequency components that the noise consist of, by letting the accelerometer data pass through a low-pass filter. However, the characteristics of the filter must be chosen carefully so that as much of the noise as possible is removed while the useful signal remains unspoiled. That is because the low-pass filter sets the limitations of how high frequency components that might be registered from the acceleration data. Therefore, there are different requirements for filtering of signals for the inputs to the tilt and acceleration algorithms. This suggest that two filters might be desirable, one for each algorithm. But what are the differences in frequency components in an acceleration and a tilt signal? That is explained in the two following sections.

5.1.1 Filtering for acceleration sensing

In the case of a car crash the impact will send a high frequency impulse through the car. The lower the filters cut-off frequency, the longer time it will take for the impulse to be registered. The following question needs to be answered; What is the highest frequency component that needs to be recognized? This frequency will set the bandwidth. According to the Nyquist sampling theorem [2], seen in figure 2.4, the answer to that question is $F_{BW} = \frac{F_s}{2} = \frac{2000}{2} = 1kHz$. Thus the low pass filters cut-off frequency should be $F_{cut-off,acc} = 1kHz$ for the filter used for the acceleration algorithm.

5.1.2 Filtering for tilt sensing

In the test sequence used at Autoliv to verify the tilt functionality of the car sensor (described in section 4.5.1), the tilt is slow ($\frac{2^{\circ}}{second}$). To get a rough estimate on the highest frequency component in such a signal, an example is useful. If the car sensor should lock the belt at an angle θ with an accuracy of $\pm 1^{\circ}$ (2° interval) and the rate of tilt is $\frac{2^{\circ}}{second}$, the value of the accelerometer needs to be sampled at least $\frac{2}{2} = 1$ time each second (1 Hz) not to miss the specified accuracy interval. That is a very slow tilt and it is probably more realistic to say that it is a requirement to recognize a $\frac{20^{\circ}}{second}$ tilt, giving a frequency component of $\frac{20}{2} = 10Hz$. This is not strictly specified in the requirements other than that the car sensor should pass Autolivs tilt test where the rate of tilt is $\frac{2^{\circ}}{second}$. Therefore $F_{cut-off,tilt} = 10Hz$ is a suitable cut-off frequency for the tilt filter.

5.2 Algorithms

To be able to appreciate the challenges of VHDL construction, some background knowledge in binary calculus is necessary. This is presented before the actual algorithms are described. But first an explaination on how tilt and acceleration is distinguishable.

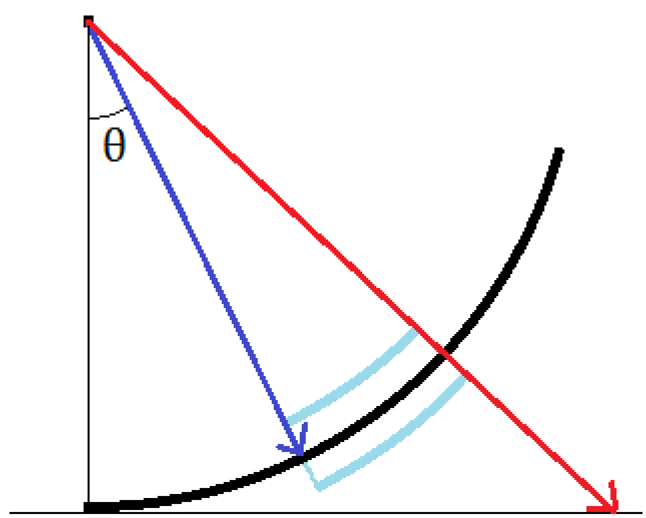

Figure 5.1: Distinguish between tilt and acceleration

Figure 5.1 show the acceleration excerted on an accelerometer with constant tilt compared to when the accelerometer is accelerated in the horizontal plane.

For static tilt, the resulting acceleration from all axes will be 1 g, coming from the gravitaional pull. In the case where the accelerometer is accelerated in the horizontal plane the acceleration will add to the magnitude of the resulting acceleration, on top of the gravitaional pull. This property is used within the algorithms to distinguish between tilt and acceleration. The accelerometer is considered tilting if the resulting acceleration from all three axes are within a choosable span around one and the tilt algorithm can lock the belt if the threshold tilt is reached. Otherwise, the signal will be considered being an acceleration and the acceleration algotithm can lock the belt if the threshold acceleration is reached.

5.2.1 Binary calculus

The binary system is based on representing numeric values using only two symbols, 0 and 1. Within computer systems each binary logic is built up using transistors, those can be seen as very small switches with two possible states. If the transistor is leading current this is interpret as a logic one and if the switch is blocking the current it is read as 0. Based on this simple logic it is possible to construct very complex logic functions. The normal representation when writing code is 8 bits, also called a byte. This actually consist of 8 logic switches resulting in $2^8 = 256$ different combinations. Each added bit increases this number with a factor of 2. Figure 5.2 show an example how the numeric values are represented using the binary numeric system.

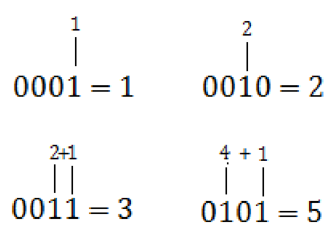

Figure 5.2: Binary weight

To represent negative numbers in binary the MSB is considered a sign bit. If the bit is 0 the the number is positive, and if it is a 1 it is negative [15]. An 8 bit signed number may represent values between -128 to 127. To convert a positive value for example 3 to the corresponding negative value, a method called two complement is used. As shown in figure 5.3, the bits are inverted and a 1 is added. The idea behind this is that addition should continue to work without adding extra steps to handle negative numbers [17].

$$
\begin{array}{l}
0101 \\
1010 - \text{invert bits} \\
\underline{\quad +1} \\
1011
\end{array}
$$

Figure 5.3: Two complement

To represent fractional numbers with binary coding, the position of the radix point needs to be decided. The value of the first decimal bit is $2^{\ 1}$ (1/2) and for each next less significant bit the value is decreased with a factor of two, $2^{\ 1} = \frac{1}{2}$; $2^{\ 2} = \frac{1}{4}$; $2^{\ 3} = \frac{1}{8}$; $2^{\ 4} = \frac{1}{16}$; $2^{\ 5} = \frac{1}{32}$ ... For example using 4 bits where the MSB is an integer and the rest is considered fractions the representation of 0.75 would be 0110. Other values like 0.1 or 1/3 can not be represented with a finite number of bits but will be more accurate the longer the vector is.

Addition The rules for binary addition are seen in figure 5.4.

Addition

$0 + 0 = 0$

$0 + 1 = 1 + 0 = 1$ (Commutative)

$1 + 1 = 0 +$ Carry

Figure 5.4: Rules for binary addition

The two first rows are quite obviously but the last row with the carry bit needs a bit of explaining. Since each bit can only contain two different values, either one or zero, a carry 1 is created to be added to the next position. The normal decimal system is actually working in the same manner but has a radix of 10 (0-9) instead of just two.

An example of this is seen in figure 5.5 left side where a carry is created in the LSB (least significant bit) and added to the next more significant bit.

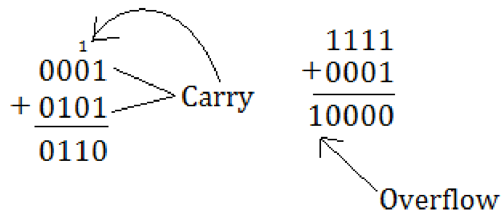

Figure 5.5: Binary addition and overflow

In the right part of figure 5.5 another situation can be seen. If for example the value no longer can be represented by the number of bits available, then an overflow bit is created. This is normally the carry bit but if the bit vector is not long enough to represent a larger number the carry bit is lost [18].

Subtraction The rules for subtraction can be seen in figure 5.6 on the left side.

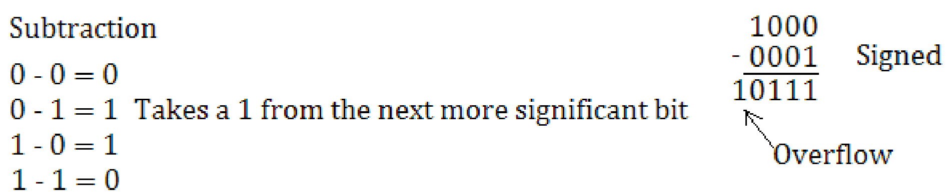

Figure 5.6: Binary subtraction rules

As can be seen in the right figure it is important to observe any overflow bits and if the value is considered signed or not. If the top value would not be signed the equation would be correct since the overflow bit would be ignored and the resulting value of 8 1 would be 7, however if it is signed the top value would be -8 and the resulting should be -9. But since the resulting vector is only 4 bits long, it can not contain any number lower then -8 and therefor an overflow bit is created. The result would be a positive value of 7 and an overflow bit. In the left part of figure 5.7 normal subtraction is done in a very similar fashion to addition but instead of creating a carry bit, subtraction "borrows" a bit from the next more significant bit position [19].

```
Carry bit is taken from higher position
            1
          0101          0101
        - 0011         +1101
          0010         10010
      overflow bit is ignored
```

Figure 5.7: Binary subtraction

Another way to see the subtraction is seen in the right part of figure 5.7 where the negative part is converted using two-complement and is instead added to the value resulting in an addition of an overflow bit that is ignored.

Multiplication The rules for binary multiplication are seen in figure 5.8.

$0 * 0 = 0$
$1 * 0 = 0 * 1 = 0$ Commutative
$1 * 1 = 1$

Figure 5.8: Rules for binary multiplication

Similar to how normal multiplication is done in the decimal system each position notation indicate how many times each value should be multiplied for that position. For example in 13 24 the number 13 is first multiplied with the number 4 that has the position value of 1 and then by 2 which has the position value of 10. Multiplication is actually a number of additions as can be seen in figure 5.9 [20].

```
    1011
   *0111
    1011
   1011
  1011
 0000
 1001101
```

Figure 5.9: Binary multiplication

The number of additions will increase depending on the length of the binary number. Since the number of additions increase with longer bit sequences but also the size of the adders as explained previously under section 5.2.1, the time and FPGA area will heavily increase when multiplying large numbers. The resulting vector in figure 5.9 is 7 bits long. However since it is possible to get an overflow if the two original binary numbers are large enough (for example the MSB bit is 1 in each) the resulting vector needs to be at least twice as long as the vectors that are multiplied for it to be guaranteed to fit the resulting value.

Division Just like multiplication was a number of additions, division is a number of subtractions. However division has been avoided in the DSP due to limitations in the FPGA compiler. Division exist but can not be used unless the division is a factor of two [21]. This can be solved by writing added functions to handle those situation but since this would take a considerable amount of time and the algorithms used for the prototype could be rewritten to avoid using division this has not been done.

Square root The calculation of the square root can be done in several ways. One of those is Newton's iteration [22]. This approximation sequence is seen in equation 5.1, where x_0 is a first, very rough, estimate and S is the original value.

$$x_1 = \frac{1}{2}(x_0 + \frac{S}{x_0}) \Longrightarrow x_2 = \frac{1}{2}(x_1 + \frac{S}{x_1}) \Longrightarrow ... \tag{5.1}$$

Each iteration of this equation will make the result more accurate. To get an accurate result with this solution, several divisions and additions are needed which would increase the calculation time and area needed significantly and the problem with the division mention earlier would need to be solved. Similar to division the square root could be avoided in the prototype and therefor no algorithm was done to solve this.

5.2.2 Tilt algorithm

If an objects predefined axis no longer align with the axis of earths gravitational field, it is considered being tilted. If an object is not tilted and not accelerated in any direction, it is exposed only to the gravitational pull that is aligned with the z-axis. In that case, the z-axis will register a 1 g acceleration and the x and y axis a zero g acceleration. The calculations for the tilt has gone through a couple of iterations before arriving at the final solution. Tests with simpler equations based on only two of the three axes has been tried but the accuracy of the result was to low to be of use for this application ($\pm 5^{\circ}$ tilt angle). The equations are not advanced in a theoretical sense but take a lot of performance to implement in an FPGA, more of this is explained in section 5.2.1.

The tilt is calculated by vector addition of X and Y and compare the resulting vector with the Z vector [23]. The absolute value of the resultant acceleration of X and Y can be calculated using Pythagoras theorem, this is shown in equation 5.2. The direction of the resulting vector is not of interest since it is only the amplitude that is used by the algorithm.

$$XY = \sqrt{X^2 + Y^2} \tag{5.2}$$

By building a coordinate system where XY is one of the unity vectors and Z is the other, the equation will be in the simple form of equation 5.3.

$$\tan(\theta) = \frac{XY}{Z} = \frac{\sqrt{X^2 + Y^2}}{Z} \tag{5.3}$$

Since many mathematical operations are quite complicated to execute in VHDL as explained in section 5.2.1 and take both time and performance these equations have been changed. By rearranging the equation to the form seen in equation 5.4.

$$\tan(\theta)^2 * Z^2 = X^2 + Y^2 \tag{5.4}$$

The first part, $\tan(\theta)^2$, will be calculated in Matlab and inserted as a parameter to avoid unnecessary calculations within the DSP. Since the accelerometer do not measure tilt but only the forces it is exposed to, some added algorithms are needed in order be able to discriminate between tilt and acceleration. So far the tilt equations have been based on the assumption that the accelerometer is only exposed to the gravitational force of 1 g. However since this value changes if the car is exposed to other forces like speeding up or braking this no longer is an adequate assumption. By looking at the total acceleration vector resultant, based on the three axes and comparing this value with a parameter, it is possible to tell within which values the accelerometer is consider tilting. Since the absolute value of the three axes acceleration will never go over 1 g if the vehicle is only tilting, the parameter is set to one or something very close.

5.2.3 Acceleration algorithm

Acceleration is the rate of change of velocity with time. Velocity is defined as a vector with both direction and amplitude. Any changes to this vector is an acceleration. That means that not only during velocity increase but also if the objects velocity is decreased (called retardation) or if there are changes in the objects direction of travel this will also be within the definition of acceleration.

The equation for the acceleration is based on the same principle as for the tilt calculations. The difference is that instead of $\tan(\theta)^2$ as an input parameter the acceleration limit is used, in this case 0.7 g. Figure 5.1 is only illustrative but the rightmost arrow show how the scaling work depending on the z axis. If the vehicle is subject to a force aligned with the gravitational force this will result in a higher threshold and if the force is opposite to the gravitational force this will lower the threshold.

Chapter 6

Conclusion

The prototype car sensor was tested for both tilt and acceleration scenarios in Autolivs test laboratory. The performance shown by the results were comparable to a mechanical car sensor. It may therefore be concluded that it is possible to make a car sensor with integrated electronics. The prototype car sensor operates silently, apart from the instance when locking the belt. Keeping the amount of unwanted belt lockings at a minimum by fine tuning and modifying to the algorithms, sounds generated from the car sensor will be scarce. This show that it is possible to achieve a silent car sensor using integrated electronics.

6.1 Discussion

When looking at the effect of using different filters, it is evident that the choice makes a difference. It is a trade off between unwanted belt locks and amount of payout. A cutoff frequency lower than 1200 Hz could be used to get a low amount of unwanted belt locks while the payout still is within the requirement limit. Thus, using an intermediate cutoff frequency e.g. 150 Hz, will give a better comfort for the user than if 1200 Hz was used. The comfort might be furter enhanced with more advanced algorithms, especially for the tilt, since that is the most sensitive case. The prototype car sensors behavior in scenarios like driving on a cobbled street is not tested, but should be possible to manage without any unwanted belt locking.

As was described in the beginning of section 4.5 the noise observed had strange characteristics, the amplitude of the noise was the same each time and only occurred at certain values. It is however not certain that the noise actually exists or if it is the transfer to the GUI that generates the error. This is not very likely since, as mention before, the number of spikes are reduced when lowering the cutoff frequency. Since no modifications are done to the digital platform and the host interface when changing cutoff frequency of the accelerometer the spikes should remain if it was any of those systems that were the cause. However, since the system changes values more seldom at low cutoff frequency and it is the changes that seems to generate the spikes this could actually reduce the number of spikes even if no changes to the program were done. The results from the test in section 4.5.2 supports the hypothesis that the spikes does not exist since the values, when using a cutoff frequency of 1200 Hz, should fluctuate much more due to the frequent occurrence of spikes.

According to the data sheet for BMA 180 noise is inserted when the SCK, SDI or CSB are used and therefor any read and write operations should be kept to a minimum to avoid adding noise [25]. As mention in section 4.2.1 the prototype system is not using the interrupt functionality of the accelerometer, instead it reads the accelerometer values with no regards if the data has been updated or not which results in unnecessary readouts. This could be the reason for the spikes but it is unlikely that the amplitude of the noise would be constant and of that large magnitude.

Another hypothesis is based on the idea that it is the number of readouts that generates the spikes. If the update of the accelerometer value and the readout occurs simultaneously it is possible that the data could become corrupted. This is especially sensitive when large bit changes occur, for example when switching from (1111 1111 to 0000 0000). The reason for these spikes remain unknown and there is even some uncertainty if they actually exist. Nevertheless, they should not exist. In a commercial system the stationary noise level will be very low and have very little to none affect on the application, but there will be other noise sources that the system needs to be tested for, e.g. electromagnetic compatibility (EMC).

The variation of measured values seen in the tilt measurments in section 4.5.1 could be explained both by the calibration and by the error caused by the resulotion limitation mentioned in section REF. Due to this limitation of resolution the tilt measurment will differ depending on how the accelerometer is tilted. If more accurate measurments of especially tilt but also acceleration is demanded more bits could be added in the calculations with the drawback of increased FPGA resource usage.

The algorithms explained in section 5.2 are based on the assumption that the accelerometer is calibrated in its starting position i.e. when the vehicle is standing still without tilting, however the accelerometer do not necessary need to be horizontaly mounted in the vehicle as long as it has been calibrated for this. According to the sensor applications expert Thomas Kepcija from Bosch it should be possible to calibrate the accelerometer with an angle of inclination of up to 15º or more. The resulting values should be accurate and work according to the data sheet for the accelerometer. Since the accelerometer can be calibrated to change the bias, the algorithms should work without the need to change any parameters or modify the equations. It is however important to ensure that the axes are calibrated orthogonal to each other. Otherwise the algorithms will suffer inaccuracies resulting in incorrect assumptions regarding when to lock and unlock. This is one of the main features with this type of system since it is quite easy to recalibrate the axes resulting in flexibility for mechanical design and low cost for implementation into new systems. Although no cost estimates were made to compare this system with the previous mechanical system it can supposedly be cheaper in development cost since the mechanical system needs to be redesigned if the bias for the mechanical sensor is changed. Comparing the price per unit of a mechanical car sensor and an integrated electronics car sensor, the price for the mechanical car sensor will be hard to beat. It will hopefully be possible to get a price within the same order of magnitude.

Regarding the speed of the system it can be seen based on the requiments in section 2.1 and the algorithm in section 5.2 that the actuator is the main source of delay. Since the update rate of the accelerometer is less the a half millisecond and the calculated delay of the DSP system is only 5.7 µs, the payout acording section 2.1 should be less then half a mm for 150 g and one mm for 17.5g if ignoring the delay from the actuator. Using this information and compare with the payout from the tests some estimations regarding the delay of the actuator may be made. However those values are only speculated and further tests are needed to establish how large delay each component add.

The mechanical and prototype car sensor are fundamentally different in respect to their response times. Both sensors are more sensitive to a large onset, but the ball in the mechanical car sensor will move faster, giving a shorter response time, the higher the velocity difference is. The prototype car sensor will not be significantly affected by the magnitude of the velocity difference, having a constant response time.

6.2 Further work

The prototype car sensor implement two algorithms, one for tilt and one for acceleration sensing. It is possible to implement more algorithms for better safety. An algorithm for combinations of tilt and accelerations would not yield better results in Autolivs test benches, but could possibly enhance the performance in terms of safety in real life scenarios. Another kind of algorithm that might be useful is an algorithm that senses trends and locks the belt preclusively by extrapolating the sensed acceleration.

An interesting field of science that might be useful to investigate is energy harvesting. Using clever ways of generating energy to the sensor, it might be possible to do without an external power supply. Energy may for example be extracted from vehicle movements and vibrations or when spooling in and out seat belt.

The locking mechanism used in the prototype car sensor belt retractor, is originally designed for mechanical car sensors. When the car sensor is made electronic, it is possible that a new design of the belt retractor with a new locking mechanism would get higher performance in terms of payout, actuation sound and production cost.

6.3 Summary

This report answer the question; Is it possible to make a silent car sensor by using integrated electronics? To find the answer, a prototype of an integrated electronics car sensor was developed. It uses a digital accelerometer as sensor, an FPGA for interfacing and signal processing and a truck belt retractor containing an electromechanical actuator. Tests were made on the prototype car sensor. A comparison between a mechanical car sensor and the prototype car sensor showed that it is possible to get comparable performance and a more silent operation using the prototype. So the answer is, yes. Unlike the mechanical car sensor, the prototype car sensor is possible to automatically calibrate so that the sensor might be installed with an inclination relative to the horizontal plane, enabling development cost cutting being able to use a belt retractor model for several car models. A drawback is that the developed car sensor demands a voltage supply.

Bibliography

[1] Andrejaši, M. (2008) MEMS Accelerometers, University of Ljubljana Faculty for mathematics and physics Department of physics

[2] Björklöf, D. (1986) Computers in systems for measurement and control, Stockholm: DAXAB Edutronics

[3] Pacheco, M.A.C. Vellasco, M.M.R. (2002) Automatic Repair Techniques for Analog Circuits, IEEE Industrial Technology: IEEE ICIT '02. p.804-809 vol.2

[4] Singh, R. Singh, B.P. (2002) Microprocessor Interfacing and Applications, New Age International Publishers, New Delhi India, pp.2-13

[5] Rhodes, E. (2005) ASIC Basics: Black & White Edition, Raleigh, North Carolina USA: LULU, pp.13

[6] Larsson, S-G. Arebrink, L-E. (1998) Grundläggande digital- och datorterknik Del 2 - Datorteknik, Institutionen för datorteknik Chalmers Tekniska Högskola, pp.10-20

[7] Choi (2010) Piezoelectric actuators Control Applications of Smart Materials pp.1

[8] Novotny, M. Ronkanen, P (N.A.) Piezoelectric Actuators, Tampere: Tampere University of Technology, pp.6

[9] Hemsel, T. Mracek, M. Twiefel, J. Vasiljev, P. (2006) Piezoelectric linear motor concepts based on coupling of longitudinal vibrations. Elsevier, Ultrasonics 44

[10] Hemsel, T. Mracek, M. Twiefel, J. Vasiljev, P. (2006) Piezoelectric linear motor concepts based on coupling of longitudinal vibrations. Elsevier, Ultrasonics 44 pp.594

[11] Manuspiya, S. Laoratanakul, P. Uchino, K. (2003) Integration of a piezoelectric transformer and an ultrasonic motor. Elsevier, Ultrasonics 41, pp.1

[12] Hartl (2008) Introduction to shape memory alloys, pp.3

[13] Duerig, T.W. (1990) Engineering aspects of Shape Memory Alloys, London : Butterworth-Heinemann, pp.207

[14] Duerig, T.W. (1990) Engineering aspects of Shape Memory Alloys, London : Butterworth-Heinemann, pp.192-193

[15] Johnson, E. Larsson, S-G. Arebrink, L-E. (1997) Grundläggande digital- och datorteknik Del 1, Institutionen för datorteknik Chalmers Tekniska Högskola, pp.6-12

[16] Propagation of uncertainty, 2011 Wikipedia [online] Available at: <http://en.wikipedia.org/wiki/Propagation_of_uncertainty> [Accessed 3 June 2011]

[17] Two's complement, 2011 Wikipedia [online] Available at: <http://en.wikipedia.org/wiki/Two's_complement> [Accessed 31 May 2011]

[18] Johnson, E. Larsson, S-G. Arebrink, L-E. (1997) Grundläggande digital- och datorteknik Del 1, Institutionen för datorteknik Chalmers Tekniska Högskola, pp.6-5

[19] Johnson, E. Larsson, S-G. Arebrink, L-E. (1997) Grundläggande digital- och datorteknik Del 1, Institutionen för datorteknik Chalmers Tekniska Högskola, pp.6-7

[20] Johnson, E. Larsson, S-G. Arebrink, L-E. (1997) Grundläggande digital- och datorteknik Del 1, Institutionen för datorteknik Chalmers Tekniska Högskola, pp.6-2

[21] IEEE Standard VHDL Language Reference Manual (2000) Institute of Electrical & Electronics Engineering, New York

[22] Weisstein, E. W. 2002 Square Root Algorithms, MathWorld - A Wolfram Web Resource

[23] Tuck, K. 2007 Tilt Sensing Using Linear Accelerometer, Freescale Semiconductor

[24] BMA 145 Triaxial, analog acceleration sensor Datasheet, 2008 Bosch Sensortec

[25] BMA180 Digital, triaxial acceleration sensor Datasheet, 2009 Bosch Sensortec