Lorem ipsum dolor sit amet

Consectetur adipisicing elit, sed do tempor incididunt ut labore et dolore magna aliqua

Håvard Rustad Olsen

Master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics,
Bergen University College

Department of Informatics,
University of Bergen

June 2015

HØGSKOLEN
I BERGEN

Acknowledgements

Hvard Helstrup, Johan Alme, Dieter, Arild, Christian, (Damian).

Contents

List of Figures

List of Tables

List of Algorithms

Listings

Acronyms

ALICE A Large Ion Collider Experiment. 10–16, 18

C++ A object-oriented programming language.. 18, 19

CERN European Organization for Nuclear Research. 10–12

LHC Large Hadron Collider. 10, 12, 13, 16

QCD Quantum Chromodynamics. 14

SystemC A simulation library building on C++. 18–20

TeV Terra Electron Volt. 13

TPC Time Projection Chamber. 10, 13, 15, 16

Chapter 1

Introduction

This chapter will cover the motivation, as well as the scope and goal of this report.

1.1 Motivation

The Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN) is the world's largest particle accelerator, hosting multiple ongoing experiments. After a run period of more than 3 years, the LHC at the CERN will be shut down from 2018 until 2021.[1] The purpose of this shutdown is to do maintenance on various equipment in the LHC, as well as significant upgrades to the different detectors, one of which is the detector for the A Large Ion Collider Experiment (ALICE). ALICE consists of multiple sub-detectors, which combined collect an enormous amount of data. This amount is expected to increase after the shutdown period as the interaction rate of the LHC will increase. Due to the increase in data output, the ALICE collaboration is seeking to upgrade and enhance the detector capabilities.[2] This includes a partial redesign of the readout electronics, upgrades to multiple sub-detectors and additional hardware upgrades.

The Time Projection Chamber (TPC) is the ALICE detector's main sub-detector for tracking and identifying particles. A starting/temporary? design for the new TPC readout electronics is made, and the different components are currently being developed. As this is still being worked on, many questions about the different components are yet to be answered. Are the current specifications sufficient to handle the expected increase in output from the detector? Do they have the necessary bandwidth to be able to send the data with minimal sample loss. Are the buffer memory enough to handle the traffic. Is it possible to optimize the current solution in any way?

The previous paragraph motivates us to find a reliable way of determining a sufficient design for the readout electronics, while being both time and cost efficient. One strategy for solving this problem, which will be further explored in this thesis is creating a simulation of the system. Doing a simulation requires designing a accurate representation of the readout electronics, and creating a testbench where it is possible to configure and run multiple tests.

1.2 Research Question and thesis goal

Given the motivation and introduction given in section 1.1 the research question for this thesis becomes:

Is it possible to design and implement a simulation which directly represent the readout electronics, and in doing so will it have an optimizing effect?

Further explained, the main tasks of this thesis will be to create a computer model of the readout electronics main components, and run multiple simulations on it. Experimenting with different configurations in order to find bottlenecks, faulty design or areas of improvement. The experiments should be logged, and the results will be presented in an organized fashion.

1.3 Report structure

Chapter 2 will give the reader the background information to be able to understand the different academic and scientific terms used, as well as some information about the context of the report. This includes information about CERN, the ALICE experiment and the physics most relevant to the thesis. It will discuss the current readout electronics as well as the proposed upgrade. Chapter 3 is going further into the problem discussed in this report, initial plans on solving the problem, and information about the tools used. Chapter 4 will talk about the implementation of the simulation, what problems occurred along the way, and the chosen solution. The chapter will go into the design, as well as code snippets from the implementation. With the information given in chapter 4, chapter 5 will discuss the results of the different simulations run, and evaluate the solution. Chapter 6 will conclude the thesis with some closing words, and work that can be done in the future.

Chapter 2

Background

This chapter will give the reader the background needed to set the rest of the thesis in context.

2.1 CERN

CERN is a European research and scientific organization based out of Geneva near the Franco-Swiss border.[3] CERN is a collaboration between 21 countries with a member staff of over 2500, and more than 12000 associates and apprentices. The organization was founded in 1954 and has since then been the birthplace of many major scientific discoveries. These are not limited to discoveries in the field of physics, but includes the creation of the World Wide Web.[4] Currently the biggest project at CERN is the LHC particle accelerator, which serves as the foundation for multiple experiments in the field of particle physics.

2.2 The Large Hadron Collider

Starting up on 10 September 2008, LHC is the latest construct added to CERN's particle accelerator complex.[5] It consist of a 27 kilometre underground ring of superconducting magnets and accelerators which boost the energy of the particles travelling inside the collider. The collider contains two adjacent parallel high-energy particle beams. These beams consist of protons extracted from standard hydrogen atoms by stripping them of electrons. Along the collider ring there are four intersect points where collision occurs. Each point corresponds to the location of a particle detector - ATLAS, ALICE, CMS and LHCb. The particle detectors are each built and operated by a large collaborations, with thousands of scientists from different institutes around the world. The beams travel at close to the speed of light and are guided by a magnetic field, which is created and maintained by superconducting electromagnets. Superconducting meaning that its in a state where it can most efficiently conduct electricity, without resistance or energy loss. Achieving this state requires cooling the magnets to -271.3° C , which is done by the distribution of liquid helium. The layout of the LHC ring as well as its four collision points can be seen in Figure 2.1.

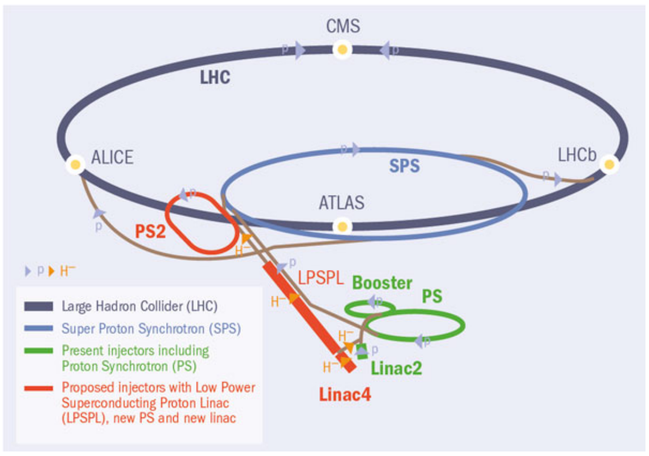
CMS
LHC
ALICE
SPS
LHCb
ATLAS
PS2
LPSPL
Booster
PS
Linac2
Linac4
p
H⁻
Large Hadron Collider (LHC)
Super Proton Synchrotron (SPS)
Present injectors including Proton Synchrotron (PS)
Proposed injectors with Low Power Superconducting Proton Linac (LPSPL), new PS and new linac

Figure 2.1: The Large Hadron Collider

The beams travelling inside the LHC reach an energy-peak of 7 Terra Electron Volt (TeV), which means that on impact with each other the collision reach an energy of 14 TeV.[6] During a normal run of the collider there will be about 600 million particle collisions per second during a period of 10 hours. This leads to a huge amount of data for each of the detectors to read out. ALICE is the detector which produce the most data per collision, with a design value of about 1.25 GB/s. The high amount of data per collision is produced primarily by the TPC sub-detector, which records a high number of points per track, and has a low momentum threshold. Detectors like ATLAS and CMS are designed with a higher momentum threshold, but can cope with significantly higher collision rates than ALICE. ALICE is designed for the study of heavy ion reactions, where particle correlations at low momentum is an important measure.

2.3 ALICE

2.3.1 Introduction

ALICE is a so called heavy-ion detector, which means it studies collisions between heavy nuclei of high energy.[7] When the experiment started heavy-ion collision lead ions was used, but they have now switched to colliding lead with protons, both produce a extreme high amount of temperature and density. They will eventually run collisions with lead to lead ions again as both types create interesting results. Lead to lead collisions create Hot Nuclear Matter, while lead to protons create Cold Nuclear Matter. The explanations for these types of matter is beyond the scope of this thesis and will not be discussed further. The high temperature and density is necessary to produce a phase of matter called quark-gluon plasma.

2.3.2 Quark-gluon plasma

Shortly after the big bang, the universe was filled with a extremely hot cluster of all kinds of different particles moving around at near light speed.[8] Most of these particles were quarks, fundamental building blocks for matter, and gluons which ties quarks together in order to form heavier particles. Normally quarks and gluons are very strictly tied together, but in the conditions of extreme temperature and density as in the time shortly after the big bang, that they are allowed to move freely in an extended volume called quark-gluon plasma. The existence of quark-gluon plasma and its properties is one of the key issues in Quantum Chromodynamics (QCD). The ALICE collaboration studies this, observing how it behaves.

2.3.3 The detector setup

The detector weighst is about 10,000 ton, it is 26 m long, 16 m wide, and 16 m high.[9] It consists of 18 sub-detectors, each with its own set of tasks regarding tracking particles and collecting data. This large number of sub-detectors are needed in order to get the full picture of the complex system which is being studied(i.e different types of particles and the correlations between them). Most of the detector is embedded in a magnetic field, created by a large solenoid magnet, which makes particles formed in collision bend according to their charge, and behave differently relative to their momentum. High momentum equals near straight lines while low momentum makes the particles move in spiral-like tracks. During lead to lead collisions the collision rate peaks at 8 kHz(Where Hz is defined as number of events per second). This number is still a lot lower in practice because the ALICE detector uses a triggered readout, which only triggers on head-on(central) collisions. The maximum readout rate of the current ALICE detector is 500 Hz, which is more than enough to track central collisions. Figure 2.2 shows a cross section of the detector as it is today with the red solenoid magnet, and all sub-detectors labelled.

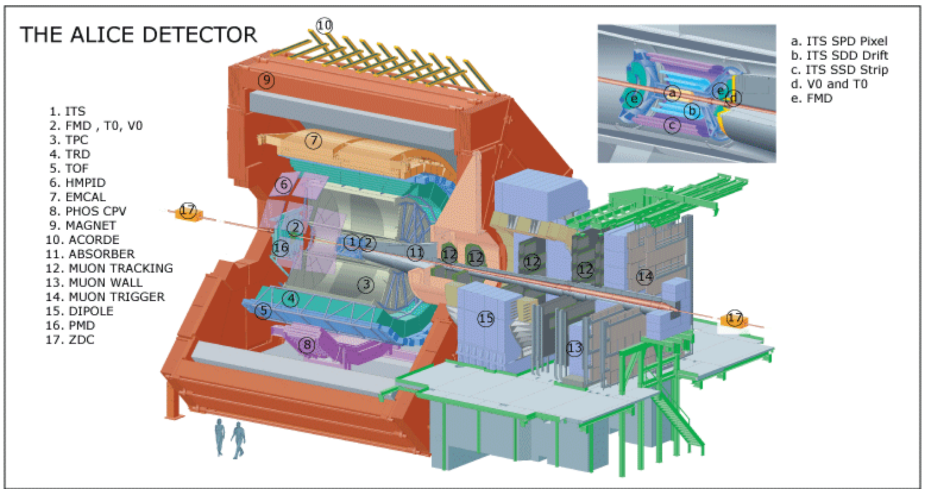

Figure 2.2: The ALICE detector

2.4 The TPC detector

2.4.1 Intro

One of the most important sub-detectors, and the one that is relevant for this thesis is the TPC detector. Located at the center of the ALICE detector it is among the first entry point when gathering data from a particle collision. It is a $88m^3$ cylinder filled with gas. The gas works as a detection medium, which means that charges particles from a collision crossing will ionize the gas atoms, freeing electrons that move towards the end plates of the detector. The readout is done by specially designed readout chambers, which are capable of handling the high amount of data produced in heavy-ion collisions.

2.4.2 Readout electronics

Signals from the readout chambers are passed along to the front-end readout electronics, which today consist of 4356 ALTRO chips. The ALTRO chip is made up of 16 asynchronous channels that digitise, process and compress the analogue signals from the readout chambers. It operates on a so called triggered readout mode. In short when ALTRO receives the first trigger, it stores the following data stream into memory, holding on to it until it is ready to pass on the data. The front-end electronics are able to readout data at a speed of up to 300 MB/s.

The ALTRO chip sends the digitised signals further down the readout chain to the readout control unit(RCU), where it is further processed and shipped to and stored in the online systems. The schematics is shown in Figure 2.3.

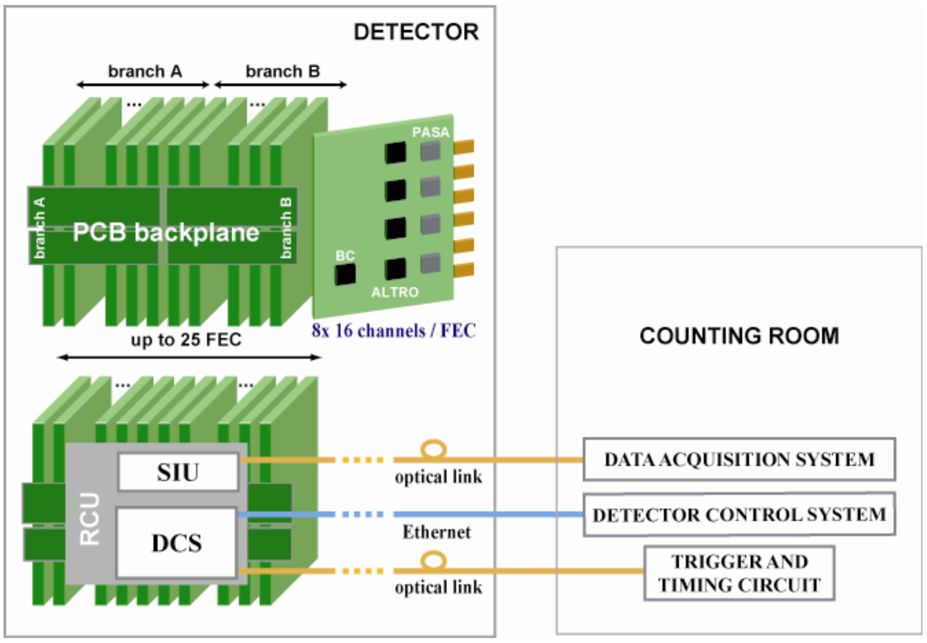

Figure 2.3: Readout schematics for the current TPC detector

2.5 Long Shutdown 2

As mentioned in 1.1 the LHC ring will be shut down for about 3 years, starting 2018. During that time the ALICE detector will undergo an extensive upgrade. The upgrade strategy for ALICE is based on the expected increase in collision rate to 50 kHz, and will now track every collision. Essentially this comes down to a increase by a factor of 100, compared to what is achievable today.

To be able to handle the increase in collision rate the TPC will receive upgrades to both its readout chambers, and front-end readout electronics. *Her skal jeg skrive mer. (Sampa, cru, etc..)*

Chapter 3

Problem description

SystemC, Starting design of the simulation, plans for implementation and test runs

3.1 Simulation

General simulation theory (Should this be in a previous chapter?)

3.1.1 Theory

A simulation can be seen as the imitation of a real-world system and its operations over time. This requires a model representation of the system which is accurate enough to conduct experiments on, which produce real-like results. The model should include key characteristics, specifications and functions of the selected system, but in a simplified fashion. A simulation model can take many forms as it can be used in different contexts ranging from physical object such as electrical circuits, bridges, and even entire cities too abstract systems like a mathematical equation or a scientific experiment.

As the model represent the system itself, the simulation represents its operations over a set period of time. The simulation is normally conducted in a controlled environment that makes it possible to observe, monitor and log results. To achieve efficient experimenting using a simulation, it should be easy to change its parameters with respect to what is being tested.

There are many benefits of simulating a system instead of creating and test the real thing. A simulation will in most cases be very time efficient, you can conduct the same kinds of experiments on the system in a much shorter time compared to the real thing. This means that more information about the systems behaviour and limitations in less time, which in turn can result in a better final product. Creating the real-world system can often be very expensive, which may limit the amount of prototypes or testproducts that are possible to create. Therefore using results of a simulation to fine tune the specifications before starting to produce prototypes will cut unnecessary development costs by a significant margin.

Taking the upgrade of the readout electronics for the ALICE detector as an example to further address this point one can see the usefulness of not having to create multiple custom hardware components, all with different purposed specification. In regards to the readout electronics, another important point is that the proposed designs might work well from the start, but there is always room for improvement. Finding out that the design doesn't need as much memory, or less optic fibre cables can impact the overall production costs. One way to efficiently and accurately simulate hardware components is by creating a virtual computer simulation.

3.1.2 Computer Simulation

Using computers to do simulations is in this day and age become more and more useful because of their incredible computational power, and ability to produce fast results.

Computer simulations often becomes quite complex, and it is often wise to use existing tools to help make the process easier. There is an array of different tools that can be used to various kinds of simulations. They vary from complete frameworks, with graphical user interfaces too tools which helps programmers write there own simulation program. The later requires of course the most work, but will most often result in the best results as you can custom tailor your simulation on a lower level then with a complete framework. A programming tool that is made for creating simulations is the SystemC library, which will be discussed in the following section.

3.2 SystemC

Explain how SystemC works, what benefits and downsides

3.2.1 Background

SystemC is a system design library based on C++. It provides an interface for easily create a software models that representing a hardware architecture, and together with standard C++ development tools it is possible to quickly build a full scale simulation. Following the standards of C++, SystemC is built to be easy to understand for both software and hardware developer, resulting in clearer cooperation between them while developing the hardware design. The SystemC library provides a object-oriented approach to model design, where a single C++ class represents a model. This makes it easy to separate concern between the different models in your simulation.

When simulating a hardware system there is a couple of key points to be aware of, firstly you need to be able to handle hardware timing, clock cycles, and synchronisation. One of the benefits of SystemC is that it takes care of all of this, again taking advantage of the object-oriented nature of C++ to extend its capabilities through `classes`. Here is some of the other features SystemC provides, with emphases on the ones needed to understand code snippets shown in this thesis.

- **Modules**
  - Container `class` representing a hardware model.
- **Processes**
  - In short, processes are methods inside a module which describe the module functionality.
- **Ports**
  - Port represent the input and output points of a module, they can be connected to other modules through Channels. Ports can be both one directional or bidirectional.
- **Channels**
  - Channels are the wires connecting two ports. SystemC comes with three predefined channels: fifo(First-In-First-Out), mutex, and semaphore. It is possible to configure custom channels, but in most cases it is not necessary.
- **Signals**
  - Signals represent data sent between modules via ports. They can be arbitrary data types like `bool` or `int`, but also user defined types.
- Rich set of data types
  - SystemC supports all data types defined in C++ as well as multiple custom types.
- Clocks
  - SystemC comes with the clocks, which can be seen as timekeepers of the system during a simulation.

3.2.2 Small example

To get a basic understanding of how a SystemC simulation looks like, it is useful to see it in action. The following Figure Figure 3.1 and Listings 3.1-3.3 make up a very trivial example with only 2 modules; a `Producer` and a `Consumer`. The `Producer` will increase a counter every clock cycle, and send a `bool` value based if the count is an even number, and send this value to the `Consumer`, which registers how many times the `Producer` counted an even number. The a

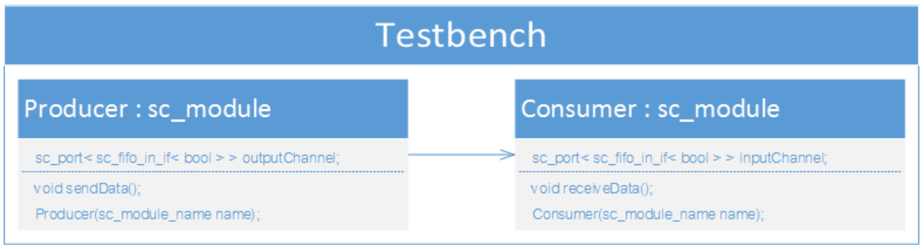
Testbench
Producer : sc_module
sc_port< sc_fifo_in_if< bool > > outputChannel;
void sendData();
Producer(sc_module_name name);
Consumer : sc_module
sc_port< sc_fifo_in_if< bool > > inputChannel;
void receiveData();
Consumer(sc_module_name name);

Figure 3.1: Basic SystemC example

```
SC_HAS_PROCESS(Producer); //macro to indicate that the module has process

//Constructor with name of module as parameter
Producer::Producer(sc_module_name name) : sc_module(name){
  SC_THREAD(sendData); //Registrer the sendData thread
}

//Thread which runs until the simulation is over.
//Clock frequency: 100 Mhz; 1 / 10 ^ 7 = 10 nanoseconds
void Producer::sendData(){

  bool signal = false; //signal value
  int count = 0; // count variable

  while(true){ //infinite loop

    if(!(count % 2)){ // if count is even, signal = true
      signal = true;
    }

    outputChannel->nb_write(signal); //write signal to output channel

    signal = false; //reset signal
    count++; //increase count
    wait(10, SC_NS); //End of a clock cycle, wait 10 nanoseconds
  }
}
```

Listing 3.1: Hello

```
SC_HAS_PROCESS(Consumer); //macro to indicate that the Module has 1 or more processes

//Constructor with name of module as parameter
Consumer::Consumer(sc_module_name name) : sc_module(name){
  SC_THREAD(receiveData); //Registrer the receiveData thread

}

//Thread which runs until the simulation is over.
//Clock frequency: 100 Mhz; 1 / 10 ^ 7 = 10 nanoseconds.
void Consumer::receiveData(){

  int numberOfEvens = 0; // counts number of evens
  bool receivedSignal = false; //received signal variable

  while(numberOfEvens < 10){ //stop lopp when received 10 evens

    if(inputChannel->nb_read(receivedSignal)){ //receiving signal; nb_read returns true if signal is
      if(receivedSignal){
        numberOfEvens++; // if signal is true, count was even.
      }
    }
    wait(10, SC_NS); //End of a clock cycle, wait 10 nanoseconds
  }
  sc_stop(); //Force stop simulation.
}
```

Listing 3.2: Hello

```
int sc_main(int argc, char* argv[]) {

  Producer producer("Producer");
  Consumer consumer("Consumer");

  sc_fifo<bool> channel(20); //(First-In-First-Out) channel with depth of 20.

  //Connecting Producer-Consumer channel.
  producer.outputChannel = channel;
  consumer.inputChannel = channel;

  sc_start(); //Alternative: sc_start(30, SC_NS) - Specified simulation lenght.

  return 0;
}
```

Listing 3.3: Hello

3.3 Model Designs

Different design patterns, and plans for the electronic

3.3.1 Zero suppression

3.3.2 Huffman Encoding

Chapter 4

Solution implementation

Code snippets, Incremental implementation stages and the final implementation, (before and after huffman), using real data vs random. Implementing fluxiation into the simulation

Chapter 5

Evaluation and results

Running the tests, results from different tests, Evaluating the final product

Chapter 6

Conclusion and Future work

Conclude the thesis, talk about the impact it has and its usefulness in future planing of the front end electronics.

Bibliography

[1] Long Shutdown 2 @ LHC. https://indico.cern.ch/event/315665/session/7/contribution/37/material/paper/1.pdf. Accessed: 2015-01-09.

[2] Werner Riegler. The ALICE Upgrade plans - Article. http://ph-news.web.cern.ch/content/alice-upgrade-plans. Accessed: 2015-01-12.

[3] CERN - Article. http://home.web.cern.ch/about. Accessed: 2014-11-12.

[4] The birth of the web - Article. http://home.web.cern.ch/about. Accessed: 2014-11-12.

[5] The Large Hadron Collider - Article. http://home.web.cern.ch/topics/large-hadron-collider. Accessed: 2014-11-14.

[6] The Large Hadron Collider - Brochure. http://cds.cern.ch/record/1165534/files/CERN-Brochure-2009-003-Eng.pdf. Accessed: 2014-11-16.

[7] The ALICE experiment - Homepage. http://aliceinfo.cern.ch/Public/en/Chapter2/Chap2Experiment-en.html. Accessed: 2014-11-17.

[8] Quark-Gluon plasma - Article. http://home.web.cern.ch/about/physics/heavy-ions-and-quark-gluon-plasma. Accessed: 2014-11-18.

[9] The ALICE experiment - Article. http://home.web.cern.ch/about/experiments/alice. Accessed: 2014-11-17.