The Real-Time Data Analysis and Decision System for Particle Flux Detection in the LHC Accelerator at CERN.

by Christos Zamantzas

A thesis submitted for the degree of

Doctor of Philosophy

in Electrical and Electronic Engineering

BRUNEL
UNIVERSITY
WEST LONDON

School of Engineering and Design

Brunel University

January 2006

Abstract

The superconducting Large Hadron Collider (LHC) under construction at the European Organisation for Nuclear Research (CERN) is an accelerator unprecedented in terms of beam energy, particle production rate and also in the potential of self-destruction. Its operation requires a large variety of instrumentation, not only for the control of the beams, but also for the protection of the complex hardware systems.

The Beam Loss Monitoring (BLM) system has to prevent the superconducting magnets from becoming normal conducting and protect the machine components against damages making it one of the most critical elements for the protection of the LHC.

For its operation, the system requires 3600 detectors to be placed at various locations around the 27 km ring. The measurement system is sub-divided to the tunnel electronics, which are responsible for acquiring, digitising and transmitting the data, and the surface electronics, which receive the data via 2 km optical data links, process, analyze, store and issue warning and abort triggers. At the surface installation, the processing units (BLMTCs) include Field Programmable Gate Array (FPGA) devices. Each FPGA is treating the beam loss signals collected with a rate of 25 kHz from 16 detectors. It calculates and maintains 192 moving sum windows, giving loss duration integrations of up to the last 84 seconds. For the generation of the abort triggers, demanding the extraction of the circulating beams, it compares the moving sums calculated with threshold values chosen for the given beam energy. Those thresholds can be uniquely programmable for each detector allowing calibration and offset factors to be included.

In this thesis, the BLMTC's design is explored giving emphasis to the methods followed in providing very reliable physical and data link communication layers, in merging the data from a Current-to-Frequency converter and an ADC into one value, and in keeping the moving sums updated in a way that gives the best compromise between memory needs, computation, and approximation error.

Keywords: FPGA, Processing, Data Acquisition, BLM, Ionisation Chambers, Beam Instrumentation.

Table of Contents

ABSTRACT .... **II**

ACKNOWLEDGEMENTS .... **VII**

INTRODUCTION **1**

CHAPTER 1. THE LARGE HADRON COLLIDER AT CERN **5**

1.1 THE ACCELERATOR CHAIN 5

1.2 LHC MACHINE OPERATION 7

1.2.1 Filling 7

1.2.2 Acceleration and Squeeze 7

1.2.3 Collision 8

1.3 LHC CHALLENGES IN ACCELERATOR PHYSICS 9

1.3.1 The beam-beam effect and Beam Loss 9

1.3.2 Particles have to remain stable for long times 9

1.3.3 Beam Losses and Quench of Magnets 10

1.4 LHC MACHINE PROTECTION 12

1.4.1 Beam Loss Monitoring System (BLM) 13

1.4.2 Beam Energy Tracking System (BETS) 15

1.4.3 Beam Interlock System (BIS) 16

REFERENCES 18

CHAPTER 2. BEAM LOSS MONITORING SYSTEM FOR THE LHC **21**

2.1 BEAM LOSS DETECTORS 22

2.2 FRONT-END ELECTRONICS 24

2.2.1 The BLMCFC Card 24

2.2.2 Current-To-Frequency Converter (CFC) 25

2.2.3 Front-End FPGA Processes 30

2.3 OPTICAL FIBRES 31

2.4 SIGNAL RECEPTION AND PROCESSING ELECTRONICS 32

2.4.1 The DAB64x Card 32

2.4.2 The BLM Mezzanine Card 35

2.4.3 Surface FPGA Processes 35

2.5 OUTPUT SIGNAL CONCENTRATION 37

2.6 THE BACK-END CRATES 39

2.7 RELIABILITY AND AVAILABILITY ENHANCEMENT 40

2.7.1 Tunnel System 40

2.7.2 Surface System 41

2.8 SIMULATION AND HARDWARE VERIFICATION TOOLS 42

2.8.1 Static Timing Analysis 42

2.8.2 Simulation Tests 42

2.8.3 SignalTap II Logic Analysis 43

2.8.4 In-System Updating of Memory & Constants 44

REFERENCES 45

CHAPTER 3. DATA TRANSMISSION 48
3.1 THE BLM DATA PACKET 49
3.1.1 Formatting of the BLM Packet 51
3.2 PHYSICAL COMPONENTS 52
3.2.1 The GOL (Gigabit Optical Link) 52
3.2.2 The GOH (Gigabit Opto-Hybrid) Configuration 54
3.2.3 The TLK Transceiver 55
3.2.4 The NTPPT-3 Photodiode 56
3.2.5 The Non-Volatile Memory 57
3.3 IMPLEMENTATION OF THE BLM MEZZANINE CARD 58
3.4 VERIFICATION OF THE BLM MEZZANINE CARD 60
3.4.1 Using the GOL Test Board 60
3.4.2 Using the CFC Card 61
3.5 SUMMARY 63
REFERENCES 64

CHAPTER 4. DATA RECEPTION & ERROR DETECTION 67
4.1 RELIABILITY INCREASE OF LINK 68
4.1.1 Redundant Signal Assessment 68
4.1.2 Evaluation of Comparison Scenarios 68
4.2 THE RCC (RECEIVE, CHECK AND COMPARE) PROCESS 70
4.3 CRC CODING 72
4.3.1 CRC Calculation 72
4.3.2 Probability of CRC Errors Non-Detection 73
4.3.3 CRC Parallel Implementation 74
4.3.4 CRC VHDL Implementation 75
4.4 8B/10B CODING 77
4.4.1 Encoding algorithm 78
4.4.2 Decoding algorithm 79
4.4.3 Error Detection 79
4.4.4 Error checking using redundancy of 8B/10B code 79
4.5 COMBINED ERROR DETECTION OF CRC AND 8B/10B CODE 80
4.6 SELECTION OF SIGNAL 82
4.6.1 System Availability Enhancement 82
4.6.2 Implementation of the Signal Select Function 83
4.7 METASTABILITY FROM CLOCK DOMAIN CROSSING 84
4.8 IMPLEMENTATION OF THE RCC PROCESS 85
4.8.1 Resource Utilisation by the RCC Process 87
4.9 STATIC TIMING SIMULATION OF THE RCC PROCESS 88
4.10 HARDWARE VERIFICATION TEST OF THE RCC PROCESS 90
4.11 SUMMARY 92
REFERENCES 93

CHAPTER 5. DATA ACQUISITION & MERGING ALGORITHM 96

5.1 DATA ACQUISITION 96
5.2 DATA MERGING ALGORITHM 98
5.2.1 Positive Difference Result 98
5.2.2 Negative Difference Result..........99
5.3 NOISE ON THE ADC OUTPUT 100
5.3.1 Visualisation of the Noise Pattern 100
5.3.2 Minimum-Value-Hold Function 102
5.3.3 Data-Combine Process Expected Outputs 104
5.4 SYSTEMATIC ERROR DUE TO THE DATA-COMBINE PROCESS...........105
5.5 IMPLEMENTATION OF THE DATA-COMBINE PROCESS 107
5.6 STATIC TIMING SIMULATION OF THE DATA-COMBINE PROCESS..........108
5.7 HARDWARE VERIFICATION TEST OF THE DATA-COMBINE PROCESS..........110
5.8 SUMMARY..........112
REFERENCES 113

CHAPTER 6. REAL-TIME DATA PROCESSING 115

6.1 CHOICE OF DATA PROCESSING METHOD 117
6.2 SUCCESSIVE RUNNING SUM (SRS) DATA PROCESSING..........119
6.2.1 Running Sums..........119
6.2.2 Successive Running Sums 120
6.3 OPTIMAL CONFIGURATION OF THE SRS..........122
6.4 IMPLEMENTATION OF THE SRS 123
6.4.1 Running Sum Calculation..........123
6.4.2 Production and Maintenance of a Running Sum 124
6.4.3 Optimisations of the Running Sums 126
6.4.4 Successive Running Sums in the BLMTC 128
6.5 STATIC TIMING SIMULATION OF THE SRS PROCESS..........133
6.6 HARDWARE VERIFICATION OF THE SRS PROCESS 134
6.7 SUMMARY..........135
REFERENCES 136

CHAPTER 7. THRESHOLD COMPARATOR & CHANNEL MASKING 138

7.1 THRESHOLD LEVEL COMPARATOR (TC)..........140
7.1.1 Threshold Table 140
7.2 CHANNEL MASKING 142
7.2.1 Masking Table 143
7.3 IMPLEMENTATION OF THE TC PROCESS 144
7.3.1 Optimisation of the TC Process 144
7.3.2 Resource Utilisation by the TC Process 145
7.4 IMPLEMENTATION OF THE THRESHOLD TABLE 146
7.4.1 Memory Requirements for the Threshold Table..........147
7.4.2 The Th-Table Function 148
7.4.3 Resource Utilisation by the Threshold Table 149
7.5 IMPLEMENTATION OF THE MASKING PROCESS..........150

7.5.1 Resource Utilisation by the Masking Process 152
7.6 SUMMARY 153
REFERENCES 155

CHAPTER 8. DATA LOGGING & POST-MORTEM RECORDING 157

8.1 DATA LOGGING 159
8.2 POST MORTEM RECORDING 161
8.3 IMPLEMENTATION OF THE ERROR AND STATUS REPORTING 162
8.3.1 The ESLog Function 162
8.3.2 Error and Status Reporting (ESR) in the BLMTC 163
8.3.3 Resource Utilisation by the ESR Process 165
8.4 IMPLEMENTATION OF THE MAXIMUM VALUES LOGGING 166
8.4.1 Maximum Value Calculation 166
8.4.2 Maximum Values Logging for the BLMTC 167
8.4.3 Resource Utilisation by the Maximum Value prosess 170
8.5 IMPLEMENTATION OF THE POST MORTEM 171
8.6 DATA LOGGING USER INTERFACE 172
8.7 SUMMARY 177
REFERENCES 178

CONCLUSIONS 180

APPENDIX A. IMPLEMENTATION OF THE BLMTC 185

A.1 DESIGN REPORT – RCC (RECEIVE, CHECK AND COMPARE) PROCESS 185
A.1.1 VHDL Code for CRC Check 185
A.2 DESIGN REPORT – DATA-COMBINE PROCESS 189
A.2.1 VHDL Code for Data-Combine 189
A.3 DESIGN REPORT – MASKING PROCESS 191
A.3.1 VHDL Code for the Inhibit Function 191
A.4 DESIGN REPORT –LOGGING PROCESS 192
A.4.1 VHDL Code for the OneShot Circuit 192
A.4.2 VHDL Code for Calculating the Maximum Value 193

APPENDIX B. BLM MEZZANINE SCHEMATIC DRAWINGS 194

APPENDIX C. BLMTC VERSIONS 196

C.1 BLMTC FOR THE PS BOOSTER AT CERN 197
C.2 BLMTC FOR THE HERA AT DESY 200

APPENDIX D. LINEARITY TEST OF THE BLM SYSTEM 202

LIST OF FIGURES 203

LIST OF TABLES 207

LIST OF ACRONYMS 209

Acknowledgements

I wish to express my gratitude to all the people who have contributed to the completion of this thesis with their knowledge, encouragement and by making my stay at CERN that enjoyable. The following people are only some of those whose help was invaluable.

I would like to start with Cinzia Da Via and Bernd Dehning, not only because they trusted me with this very important task but also for their help to accomplish it. I was still an undergraduate when Cinzia became my supervisor at Brunel University. In this function, for the last six years she led my studies. Between other, she introduced me to CERN, an institute at the time unknown to me. Bernd, my supervisor at CERN and leader of the Beam Loss section, welcomed me and introduced me in this community. He gave me my first accelerator physics knowledge, and found the funds to allow me to attend many courses and conferences. His exceptional character is always an example to follow. Both Cinzia and Bernd deserve my eternal gratitude for believing in me, treating me as equal, but mostly for allowing me to grow in a much better person. From Brunel, I would like also to thank Stephen Watts, my second university supervisor, for arranging all matters with the university and hence providing me with this opportunity.

Just as no research and development can be done in one day, no product can be produced in a vacuum. There are a number of people I would like to acknowledge for their contribution in this project. Starting with Gianfranco Ferioli, our chief engineer, always ready to share his immense knowledge and experience in electronics and accelerators. Ewald Effinger and Jonathan Emery, my companions in crime, whose help allowed me to reach many goals. They relieved me from any other tasks in the last couple of months to allow me to prepare my thesis undisturbed. Gianluca Guaglio, the source of reliability, pushed and helped to provide a much better design. Claudine Chery who created the prototypes, and the rest of the BL section, that is Stian Erlend Sundsoy Forde, Michael Hodgson, Eva Barbara Holzer, Jan Koopman. Daniel Kramer, Roman Leitner, Laurette Ponce, Virginia Prieto, Helge Haakon Refsum, Ion Savu, Markus Stockner, and Raymond Tissier, for their support and for making a really nice working environment.

Others have also contributed in this work without being directly involved in this project. I would like to recognise Paulo Moreira, from the Microelectronics group, for providing his test-bench and for his time and effort to pass me his valuable insights into many of the issues on the GOL device. Jean-Marc Combe, from the DEM group, for the help and his proposals in the layout of the BLM Mezzanine card. Daryl Bishop, from TRIUMF, even though being in the other side of the Atlantic he was always available to discuss and provide solutions on the DAB card. Graham Waters, also from TRIUMF, for providing an interface for flashing my code over the VME. Jose Luis Gonzalez for making possible the production of this card's last minute modifications for our system. Stephen Jackson for his ideas and the construction of the Navigator. Subroto Datta, from Altera, for his support with the Quartus II software. Ray Andraka for his help, through the FPGA newsgroup, in the DSP techniques and for introducing me to the CIC filters. Paschalis Vichoudis, always available to discuss current trends in the digital design domain. Sotiris Vlachos for providing corrections and the much-needed moral support during the writing process.

Without friends life would be so empty. For filling this gap, I would like to thank Federico Roncarolo, for being my first friend at CERN and introducing me to the most interesting group of people. Too many to list but so warm that welcomed me in their houses from the first day, as if I was an old friend. Boris Bellesia, for wasting many of his weekends to teach me the art of snowboarding. Antonio Vergara for passing to me the best apartment in Geneva and Christos Sagianos for offering help with the deposit and the furniture. Of course, I would never omit the Super Trois Genève, the funniest group of people, whose stories of lost chicken have become epic.

Finally, this cannot be dedicated to any other than Elena and the rest of my family for making my life full, for putting up with my mumblings and sketches all over the house, and for their unconditional love, eternal support and inspiration all these years...Love always.

Introduction

The Large Hadron Collider (LHC) is the next circular accelerator being constructed at the European Organisation for Nuclear Research (CERN). It will provide head-on collisions of protons at a centre of mass energy of 14 TeV for high energy particle physics research. The LHC is currently being installed in the 27 km long LEP tunnel. In order to reach the required magnetic field strengths, superconducting magnets cooled with superfluid helium will be installed.

The energy stored in the LHC can potentially damage many elements of the accelerator or could make its operation very inefficient. The 10 GJ stored in the magnetic fields of the magnets needs a protection system to dissipate the energy in case of the transition of the electrical conductors from the superconducting to the normal conducting state (quench). Moreover the 700 MJ stored in the circulating beams could initiate quenches or cause damages if only a very small fraction of the circulating beam particles deposit their energy in the equipment. The consequence will be either the warming up of the magnet coil, causing a downtime of several hours necessary for cooling down again the magnet to reach the superconducting state, or a repair time of months in the case of equipment damage. This thesis has been developed inside the CERN team responsible for the machine protection and more specifically in its section assigned to provide to the LHC a protection process from the beam losses. Thus, a system has been implemented for measuring continuously such losses and acting when those exceed specified thresholds.

The thesis begins with a general introduction to the aim and the specifications of the LHC, as well as the objectives and important parts of the machine protection system. The task, as described in Chapter 2, of observing continuously the particle losses and signalling the safe extraction of the circulating beam before the losses reach the quench level is accomplished by the Beam Loss Monitoring (BLM) system. The monitors are formed by ionization chambers that produce an electric current proportional to the particle rate traversing their volume. It is measured by an analogue front-end, which is located near those detectors in the LHC tunnel and has to cover a dynamic range of nearly nine orders of magnitude.

The second part of this thesis, which marks the beginning of the work done by the author, is committed to the effort to provide very reliable implementations of the physical and data link layers for the BLM system. Because of the radiation environment in the tunnel, the evaluation of the detector signal has to be performed in the surface buildings. This leads to long transmission distances of up to 2 km between the front-end in the tunnel and the processing module on the surface. In Chapter 3, the optical link construction is shown together with the data arrangement in a packet. The link operates in the gigabit region to provide low system latency and it is using radiation tolerant devices for the parts residing in the tunnel installation. A significant portion of the data packet is occupied with extra information, to be used at the far end of the link by the processing module, for monitoring the correct operation of the tunnel installation. This reception process, facilitated in the entry stage of the surface FPGA, is shown in Chapter 4. In addition, its implementation has been done in a way that ensures the correct reception and detection of erroneous transmissions by using digital techniques like the Cyclic Redundancy Check (CRC) and the 8B/10B algorithms.

The third part is dedicated to the real-time processing of the data. In order to extend the dynamic range of the system the digitisation of the detector signal is done by a Current-to-Frequency converter (CFC) together with an ADC. The two types of data acquired for each detector signal with those digitisation methods are different and pre-processing is needed in order for those to be merged into one value. More specifically, the measurement of the pulses, produced by the CFC, using a counter relates to the average current induced from the previous acquisition. On the other hand, the voltage measured by the ADC is the fraction of the charge in the capacitor. Chapter 5 investigates and proposes a merging method of those two different types of data, as well as it sets an acquisition strategy that provides accuracy and avoids errors. Consecutively, the BLM data main processing requires analysis of the signal pattern in time. Given the tolerance acceptable for quench prevention by the specifications, the quench thresholds versus loss duration curves have been approximated with a minimum of steps fulfilling the tolerance. In this way, the number of sliding integration windows has been reduced to twelve. Chapter 6 shows how this was made possible and proposes a highly efficient strategy of keeping many moving windows that can span to long integration time periods for each detector data.

The forth and final part, of this thesis, discusses the processes reaching to the output stages. Every Running Sum, after every new calculation, needs to be compared with its corresponding threshold value that was chosen by the beam energy reading given that moment. If the level is found to be higher, the comparator will initiate the necessary dump request. Chapter 7 proposes an implementation of a quench level Threshold Comparator that allows also the possibility of channel masking. It is using unique tables for each detector that provide the threshold values depending on the beam energy reading. Finally, Chapter 8 shows the concentration of the data for the Control Room. The main purpose of the Logging System is to continuously record and provide an online display to the Control Room of the machine status and show slow or infrequent changes. The BLM system will contribute by providing the loss rates, normalised with respect to the quench levels so that abnormal or higher local rates could thereby be spotted easily.

In the appendixes, the reader can find important parts of the VHDL code used in the construction of the FPGA firmware, the schematics of the BLM Mezzanine card, information on two similar configurations of the BLMTC used for measurements in available accelerators, and the results of a linearity test made on those systems.

01

The Large Hadron Collider at CERN

Chapter 1. The Large Hadron Collider at CERN

CERN is the European Organization for Nuclear Research, the world's most influential particle physics centre. Founded in 1954, the laboratory was one of Europe's first joint ventures, and is an example of international collaboration. From the original 12 signatories of the CERN convention, membership has grown to the present 20 Member States. The laboratory sits astride the Franco-Swiss border west of Geneva at the foot of the Jura mountains.

Its research programme attracts scientists not only from the Member States but literally worldwide. Some 6500 scientists, half of the world's particle physicists, use the available facilities. They represent 500 universities and over 80 nationalities.

CERN's primary function is to provide research facilities and basic support to this community of users. By accelerating particles to very high energies and smashing them into targets or into each other, the forces acting between them can be unravelled. CERN's accelerators are amongst the world's largest and most complex scientific instruments.

1.1 THE ACCELERATOR CHAIN

The accelerator chain at CERN is a succession of machines with increasingly higher energies, injecting the beam each time into the next one, which takes over to bring the beam to an energy even higher, and so on. The flagship of the complex will be the Large Hadron Collider (LHC).

CERN's accelerator complex (see Figure 1-1) is the most versatile in the world. It includes particle accelerators and colliders, can handle beams of electrons, positrons, protons, antiprotons, and "heavy ions" (the nuclei of atoms, such as oxygen, sulphur, and lead). Each type of particle is produced in a different way, but then passes through a similar succession of acceleration stages, moving from one machine to another. The first steps are usually provided by linear accelerators, followed by larger circular machines.

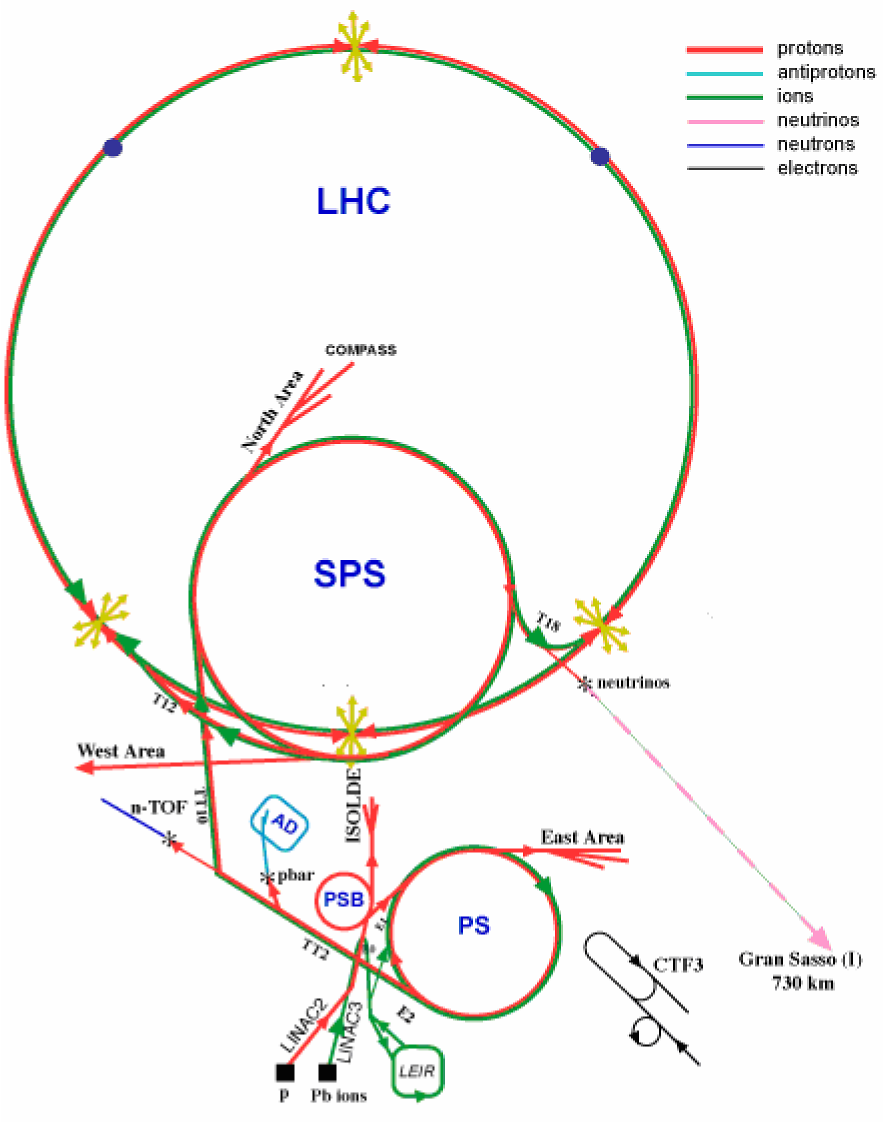


Figure 1-1: The CERN Accelerator Chain. [1]

CERN's first operating accelerator, the Synchro-Cyclotron, was built in 1954, in parallel with the Proton Synchrotron (PS). The PS is today the backbone of CERN's particle beam factory, feeding other accelerators with different types of particles. The 1970s saw the construction of the SPS, at which Nobel-prize winning work was done in the 1980s. The SPS continues to provide beams for experiments and was also the final link in the chain of accelerators providing beams for the LEP machine. It will do also the same for the next big machine, due to start operating in 2007, the LHC.

1.2 LHC MACHINE OPERATION

The LHC will consist of two "colliding" synchrotrons installed in the 27 km LEP tunnel. They will be filled with protons delivered from the SPS and its pre-accelerators at 0.45 TeV. Two superconducting magnetic channels will accelerate the protons to 7-on-7 TeV, after which the beams will counter-rotate for several hours, colliding inside the experimental detectors, until they become so degraded that the machine will have to be emptied and refilled.

1.2.1 FILLING

The first stage in the machine operation is the filling. It begins with the extraction process from the SPS, the transfer of the beam through the transfer lines, the injection into the LHC, and the establishment of captured circulating beam. The filling is done by transferring from the SPS to the LHC bunches of particles. Each bunch contains about 1.1 x 10^{11} protons and 2808 of these bunches are transferred during each extraction process. There are mainly three types of equipment involved: the fast kicker magnets (extraction, injection), the acceleration equipment (RF), and the magnetic elements (extraction bumpers in the SPS, transfer-line magnets and the LHC magnets).

1.2.2 ACCELERATION AND SQUEEZE

In the acceleration process, all beam control systems have to start synchronously the execution of the energy ramp. Once the ramp start signal is given, these systems have to maintain the same pace to step through the pre-programmed ramp functions. The synchronisation is provided by the slow timing system: a common start ramp event will be used to trigger the ramp, whilst the ramp speed is synchronised by the distributed timing signal. The ramp will last approximately 30 minutes and the energy of each circulating beam will eventually reach the 7 TeV nominal value.

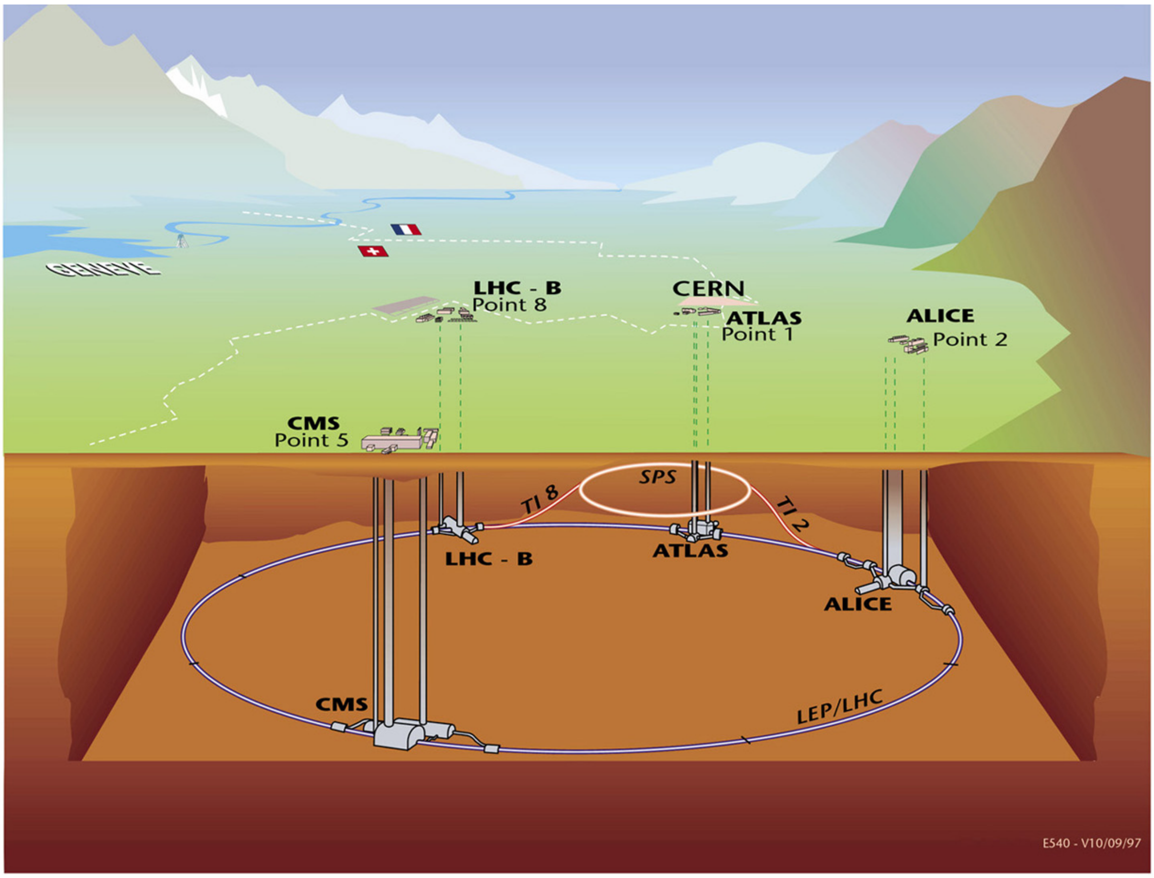


Figure 1-2: LHC Layout. [1]

1.2.3 COLLISION

The high number of bunches circulating in the LHC will be brought together 40 million times per second at four collision points spaced around the ring. These bunch crossings will result in collisions with an event rate of several hundreds of MHz. That is, conditions needed to make new (and maybe unexpected) elementary particles. For these points, detectors are being built to take ‘snapshots’ of the produced particles and measure their momentum and energy.

Four particle detectors are currently being constructed and will eventually be housed in underground caverns. Figure 1-2 shows their placement in the LHC. They will record the tracks left by debris from the collisions. The primary task of the LHC is to make an initial exploration of the TeV range. The two major LHC detectors, called ATLAS (A Toroidal LHC Apparatus) [3] and CMS (The Compact Muon Solenoid) [4], should be able to accomplish this for any Higgs mass in the expected range. Apart from those, there will be

ALICE (A Large Ion Collider Experiment) [5], which will be built to exploit the unique physics potential of nucleus-nucleus interactions at LHC energies, and LHCb [6], which will carry out precision measurements of CP-violation and rare decays of B mesons. They are scheduled to record their first data from collisions in July 2007.

1.3 LHC CHALLENGES IN ACCELERATOR PHYSICS

In the LHC, the energy available in the collisions between the constituents of the protons (the quarks and gluons) will reach the TeV range, that is about 10 times that of LEP and the Fermilab Tevatron.

1.3.1 THE BEAM-BEAM EFFECT AND BEAM LOSS

When two bunches cross in the centre of a physics detector only a tiny fraction of the particles collide head-on to produce the wanted events. All the others are deflected by the strong electromagnetic field of the opposing bunch. These deflections, which are stronger for denser bunches, accumulate turn after turn and may eventually lead to particle loss. In order to reach the desired luminosity the LHC has to operate as close as possible to this limit. Its injectors, the old PS and SPS, are being refurbished to provide exactly the required beam density.

1.3.2 PARTICLES HAVE TO REMAIN STABLE FOR LONG TIMES

The beams will be stored at high energy for about 10 hours. During this time, the particles make four hundred million revolutions around the machine. Meanwhile the amplitude of their natural oscillations around the central orbit should not increase significantly, because this would dilute the beams, degrade luminosity and causes particle impacting on equipment. This is difficult to achieve, since, in addition to the beam-beam interaction already mentioned, tiny spurious non-linear components of the guiding and focusing magnetic fields of the machine can render the motion slightly chaotic, so that after a large number of turns the particles may be lost.

In the LHC the destabilizing effects of magnetic imperfections is more pronounced at injection energy, because the imperfections are larger owing to persistent current effects in the superconducting cables, and also because the beams occupy a larger fraction of the coil surrounded cross section. The Dynamic Aperture, within which particles remain stable for the required time, needs to exceed the dimension of the injected beam with a sufficient safety margin. For the time being, no theory can predict with sufficient accuracy the long-term behaviour of particles in non-linear fields. Instead fast computers are being used to track hundreds of particles step by step through the thousands of LHC magnets for up to a million turns. Their results are used to define tolerances for the quality of the magnets at the design stage and during production to limit the particle loss.

1.3.3 BEAM LOSSES AND QUENCH OF MAGNETS

The two magnetic channels will be housed in the same yoke and cryostat, a unique configuration that not only saves space but also gives a 25 % cost saving over separate rings. Figure 1-3 shows this architecture. High energy LHC beams need high magnetic bending fields, because the machine radius was not a parameter which could have been increased to provide gentle curves. To bend 7 TeV protons around the ring, the LHC dipoles must be able to produce fields of 8.36 Tesla, over five times those used a few years ago at the SPS proton-antiproton collider, and almost 100,000 times the earth's magnetic field. Superconductivity makes this possible. This is the ability of certain materials, usually at very low temperatures, to conduct electric current without resistance and power losses. Therefore, higher currents are possible to be applied that will create the necessary high magnetic fields. For comparable power consumption, the LHC can deliver 25 times the energy and 10,000 times the luminosity of the SPS collider.

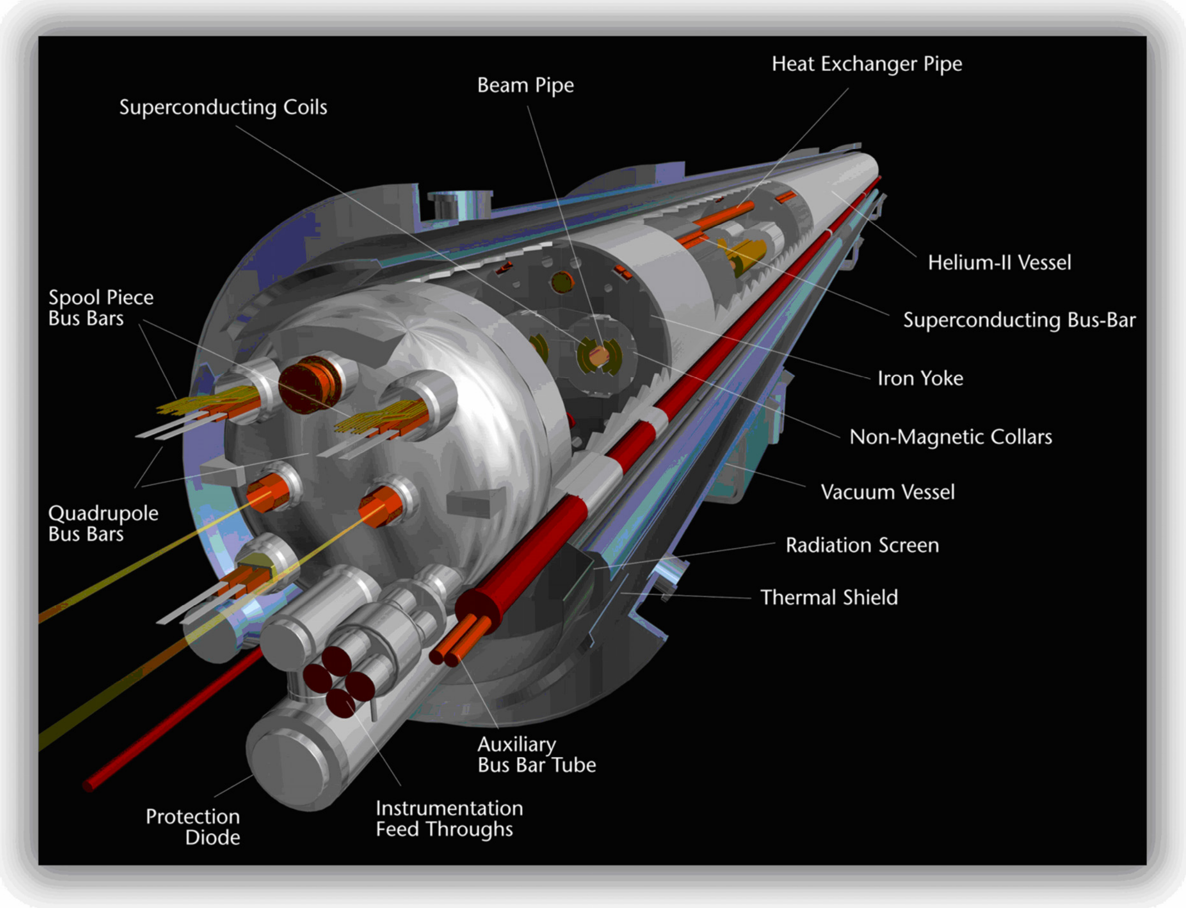


Figure 1-3: Superconducting Magnet for the LHC. [1]

Despite all precautions the beam lifetime will not be infinite, in other words a fraction of the particles will diffuse towards the beam pipe wall and be lost. In this event, the particle energy is converted into heat in the surrounding material and this can induce a quench of the superconducting magnets or damage the coil, thus interrupting operation from few hours to several months. To avoid this, a collimation system will catch the unstable particles before they can reach the beam pipe wall, so as to confine losses in well shielded regions far from any superconducting element. The LHC combines for the first time a large beam current at very high energy with the most sophisticated superconducting technology. As a consequence, it needs much more efficient collimation and beam loss measurement systems than previous machines.

1.4 LHC Machine Protection

For the LHC both the particle momentum and the beam intensity increases to unprecedented values. The proton momentum is at least a factor of seven above accelerators such as SPS, Tevatron and HERA, whereas the energy stored in the beams is more than a factor of 100 higher. The transverse energy density as relevant factor for equipment damage is a factor of 1000 higher than for other accelerators. Figure 1-4 illustrates the difference in the energy stored in the beam for LHC and other accelerators, as well as the energy stored in the LHC magnets.

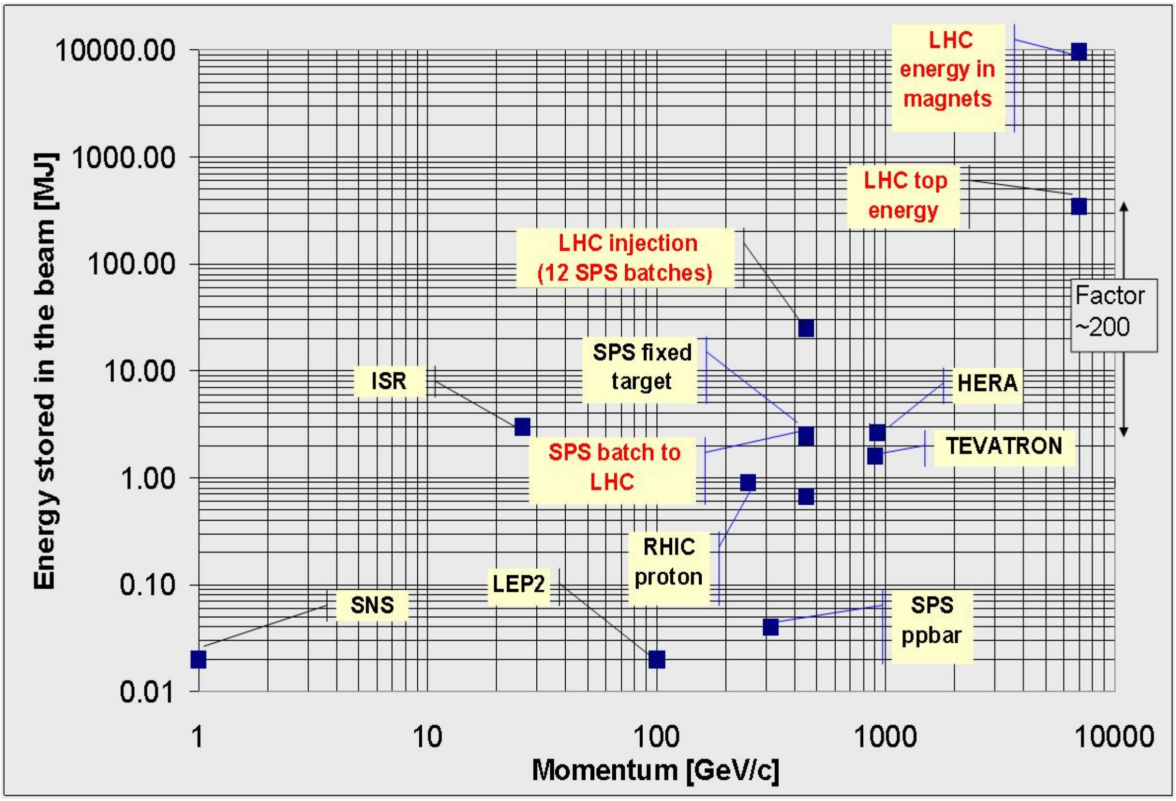


Figure 1-4: Comparison of Energy stored in the beam for LHC and other accelerators. [10]

The LHC, with about 8000 magnets powered in 1700 electrical circuits, can suffer several different kinds of accidents or malfunctions that could lead to beam losses if any of those systems fails. Other accidents can be due to aperture restrictions provoked by beam screens, interconnections, vacuum valves or collimator jaws, more than 100, which can obstruct the beam passage.

Those beam losses can happen either in a single turn, with a sudden beam loss, or in progressive losses during numerous turns. One-turn failures are called ultra-fast losses. Multi-turn failures can be divided between very-fast losses, those which happen in less than 10 ms, fast losses, which happen in more than 10 ms and steady losses, where the beam is lost in one second or more. [9]

The machine protection comprises several systems and sub-systems, such as the beam loss related protection, the quench protection, the beam interlock system, and the beam dump system. Here, the introduction will be restricted to those directly linked to the Beam Loss Monitoring system. More information can be found for the rest of the systems here [7].

1.4.1 Beam Loss Monitoring System (BLM)

The Beam Loss Monitoring (BLM) system converts the signal given by the recording shower particles into electrical signals and decides if threshold limits are exceeded. The Beam Permit signal is deasserted if one detector exceeds the limit. The Beam Permit signals are transferred to the Beam Dump system by the Beam Interlock system.

The beam loss measurement system is part of the equipment protection system [13]. The protection as foreseen for LHC is schematically shown in Figure 1-5. The number of beam dump requests, which reaches the dump system over the machine interlock, is up to 60 % by operator initiated request (inspired distribution by HERA [14]). The remaining dump requests are up to 30 % caused by beam loss initiated dumps and up to 10 % by various other reasons.

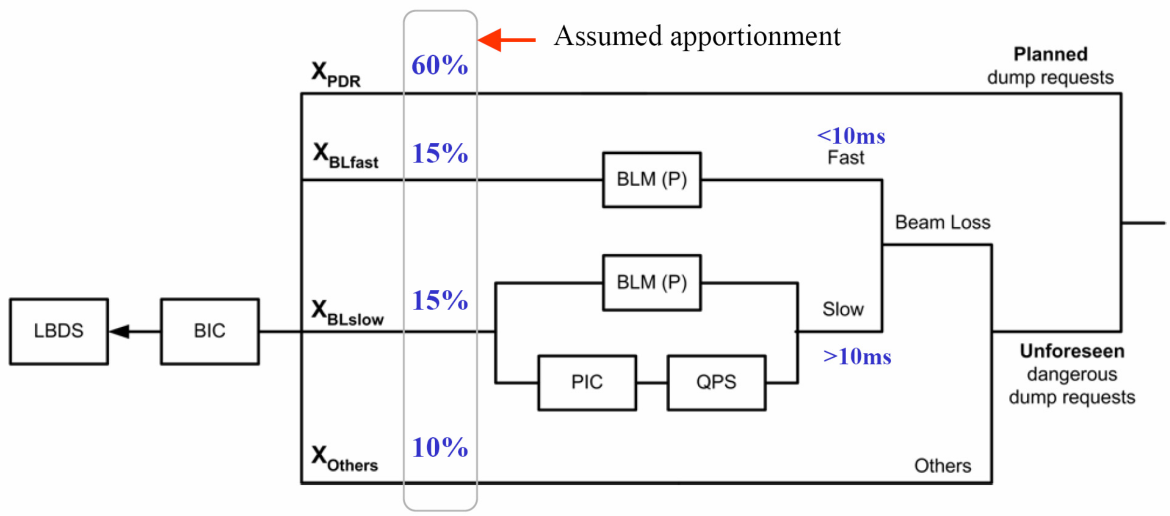


Figure 1-5: Dump request distribution and the employment of the beam loss system. [15]

The beam initiated requests are equally subdivided in losses with a duration below 10 ms and above. The long losses can be detected in addition with the quench protection system (QPS, PIC). In this case, two independent systems are available for the detection. The short losses can only be detected by the BLM system, augmenting, in this way, its criticality in the machine protection.

The proton loss initiated quench of magnets is depending on the loss duration and on the beam energy. Figure 1-6 shows a plot of the LHC bending magnets quench level curves as a function of the beam energy for different loss durations. It can be observed that, in the quench level for the same loss duration, there is about three orders of magnitude difference between the injection and the nominal energy. The BLM will need to read the beam energy delivered by the Beam Energy Tracking System (BETS), shown in the following section, to adjust the threshold levels accordingly in order to detect a possible quench.

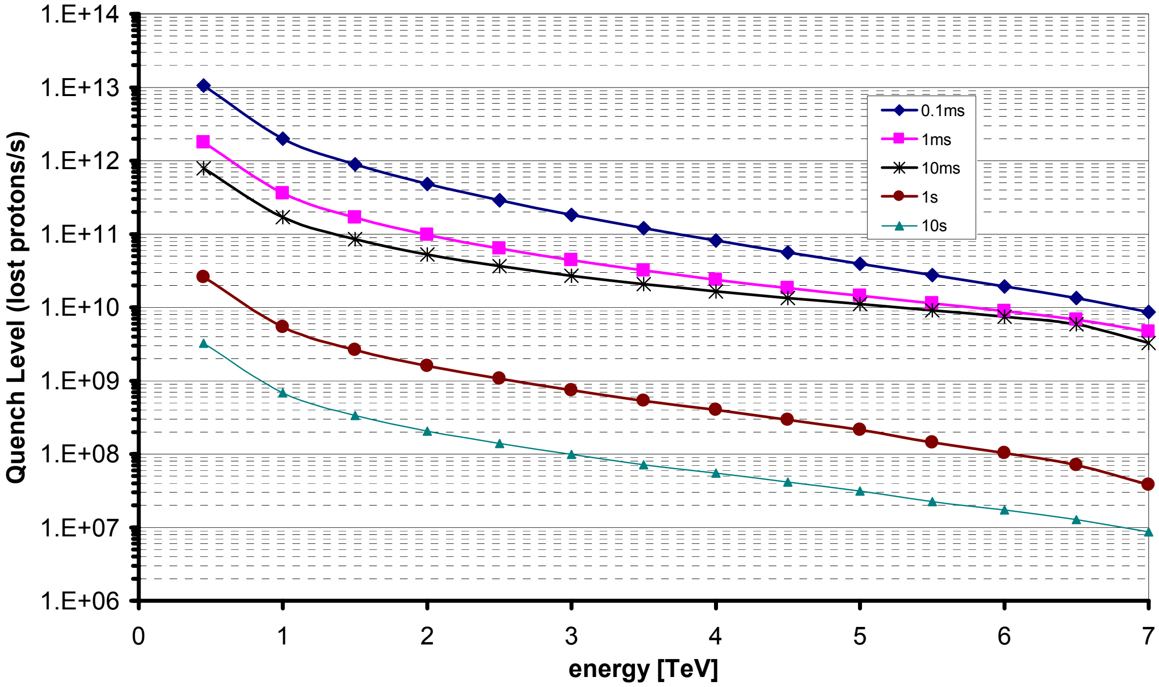


Figure 1-6: Quench Level curves for various Loss Durations as function of the Beam Energy.

In the case of LHC, a quench of a magnet will create a downtime in the order of hours where in the possibility of its destruction the downtime will be in the order of months to accommodate only its replacement.

1.4.2 Beam Energy Tracking System (BETS)

The Beam Energy Tracking System (BETS) binds the bending magnet current to the beam energy. The bending current is measured in two of the eight sectors of the LHC. At each sector the current is measured twice with two independent current transformers (DCCTs). It is then able to distribute the acquired beam energy to the external clients through the Safe LHC Parameters (SLP) system. Figure 1-7 illustrates an overview of the beam energy acquisition by the BETS.

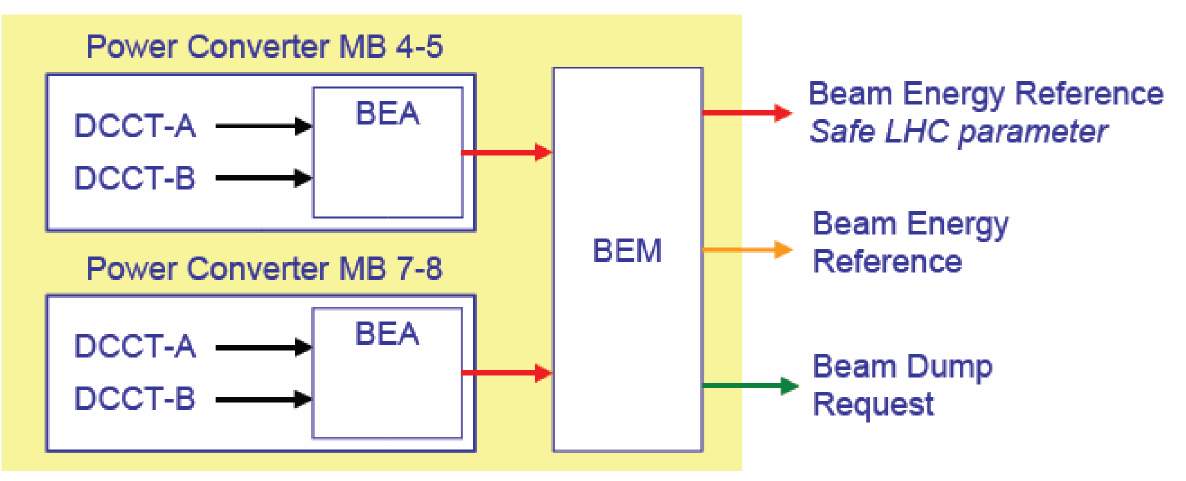


Figure 1-7: Overview of the Beam Energy Acquisition. [12]

The Beam Energy Acquisition (BEA) module acquires and digitises two independent unipolar channels with 16-bit resolution ADCs. The Beam Energy Meter (BEM) module receives four digital measurements proportional to beam energy from the BEA. These measurements are compared within the BEM with a 3 out of 4 logic and a relative error of ± 0.5%. Failure results in a beam dump request. The mean value of the four measurements is then converted into an absolute beam energy reference through a calibration look-up table.

1.4.3 BEAM INTERLOCK SYSTEM (BIS)

The Beam Interlock System for the LHC is comprised by 16 Beam Interlock Controller (BIC) modules installed right and left from each interaction point (IP). Figure 1-8 illustrates an overview of its architecture.

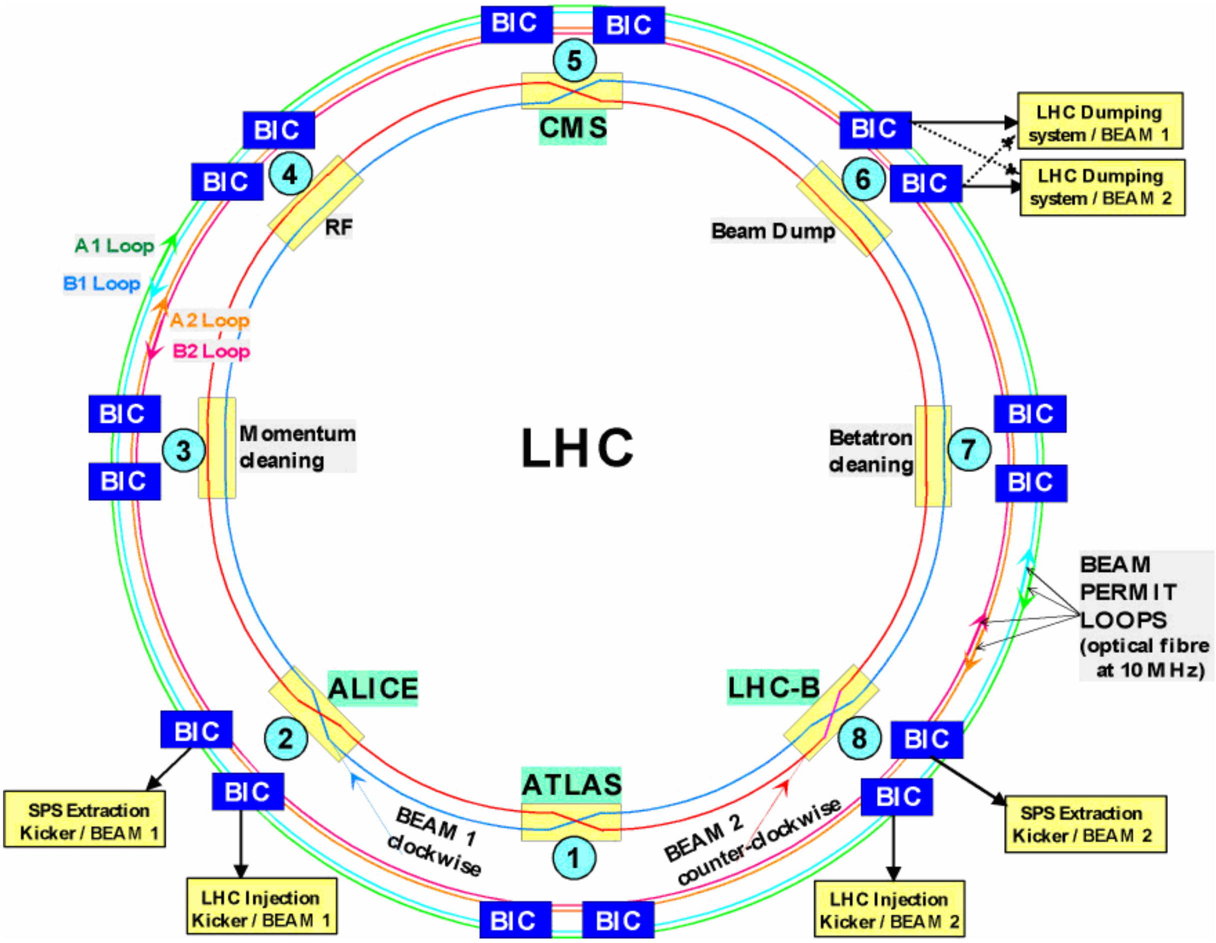

BIC
BIC
5
CMS
BIC
LHC Dumping system / BEAM 1
LHC Dumping system / BEAM 2
BIC
4
BIC
RF
6
BIC
Beam Dump
A1 Loop
B1 Loop
A2 Loop
B2 Loop
BIC
3
BIC
Momentum cleaning
LHC
Betatron cleaning
7
BIC
BIC
BEAM PERMIT LOOPS (optical fibre at 10 MHz)
ALICE
LHC-B
BIC
2
8
BIC
SPS Extraction Kicker / BEAM 1
BIC
ATLAS
BIC
SPS Extraction Kicker / BEAM 2
BEAM 1 clockwise
1
BEAM 2 counter-clockwise
LHC Injection Kicker / BEAM 1
BIC
BIC
LHC Injection Kicker / BEAM 2

Figure 1-8: Architecture of the Beam Interlock System. [11]

The controllers are connected by two loops, named Beam Permit Loops. When the loops are broken, the beams are extracted into the beam dump blocks by the Beam Dump System (LBDS). The two loops distinguish between beams I and II.

The system allows breaking one of the loops leading to a dump of only one beam and the LHC has a Beam Dump System for each beam. This property could be used for example during injection where one beam may have been successfully injected but the attempt to inject the other beam leads to a stored beam with unacceptable beam parameters. Another example is a degraded vacuum in one beam tube, where operation with the other beam will still be possible.

In order to inject beam, the Beam Dump System must be ready, all vacuum valves in the beam tube must be in the "open" position, magnets in the transfer line need to be powered, etc. The Beam Interlock Controllers ensure that these conditions are met.

REFERENCES

[1] What is CERN?, [online:http://public.web.cern.ch/public/Content/Chapters/AboutCERN/WhatIsCERN/WhatIsCERN-en.html]

[2] CERN Accelerator Complex webpage, [online:http://public.web.cern.ch/public/Content/Chapters/AboutCERN/HowStudyPrtcles/CERNAccelComplex/CERNAccelComplex-en.html]

[3] ATLAS public information pages, [online: http://atlas.ch/]

[4] CMS Outreach, [online: http://cmsinfo.cern.ch/Welcome.html]

[5] ALICE public information pages, [online: http://na49info.cern.ch/alice/html/intro]

[6] LHCb public information pages, [online: http://lhcb-public.web.cern.ch/lhcb-public/default.htm]

[7] F.Bordry, R.Denz, K-H.Mess, B.Puccio, F.Rodriguez-Mateos and R.Schmidt, *"The Architecture of the Machine Protection System of the LHC"*, LHC Project Report 521.

[8] F.Bordry, R.Denz, K-H.Mess1, B.Puccio, F.Rodriguez-Mateos and R.Schmidt, *"Machine Protection for the LHC: Architecture of the Beam and Powering Interlock Systems"*, CERN, LHC Project Report 521, Geneva, 21 December 2001.

[9] R.Schmidt et al., *"Beam loss scenarios and strategies for machine protection at the LHC"*, HALO '03, Montauk, NY, 19-23 May 2003

[10] R.Schmidt, "Machine Protection System(s) – Overview", LHC Project Workshop - Chamonix XIV.

[11] Engineering Specification, *"The Beam Interlock System For The LHC"*, LHC Project Document No. LHC-CIB-ES-0001-00-10, version 1.0, 17-02-2005.

[12] E.Carlier, G.Gräwer, N.Voumard, R.Gjelsvik, *"Update on the Beam Energy Tracking System"*, presentation given at the LHC Machine Protection Working Group, 03/06/2005.

[13] B. Dehning, *"Beam loss monitor system for machine protection"*, 7th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators, DIPAC'05 , Lyon, France , 06 - 08 Jun 2005.

[14] K. Wittenburg, *"Quench levels and transient beam losses at HERAp"*, Workshop on Beam generated heat deposition and quench levels for LHC magnets, CERN, March 2005.

[15] R. Filippini, B. Dehning, G. Guaglio, F. Rodriguez-Mateos, R. Schmidt, B. Todd, J. Uythoven, A. Vergara-Fernandez, M. Zerlauth, *"Reliability Assessment of the LHC Machine Protection System"*, Particle Accelerator Conference PAC 2005, Knoxville, TN, USA, 16 - 20 May 2005.

02

Beam Loss Monitoring System for the LHC

Chapter 2. Beam Loss Monitoring System for the LHC

Beam Loss Monitors are devices used to measure and localise beam losses over an accelerator. A Beam Loss Monitoring (BLM) system should be able to determine the number of lost particles or, in a more dramatic sense, to request a beam dump in those cases which are considered seriously dangerous for the machine.

The complete BLM system for the LHC can be easily divided geographically by the installations at the tunnel and at the surface building. The basic structures of the two parts communicating over an optical link are shown in Figure 2-1.

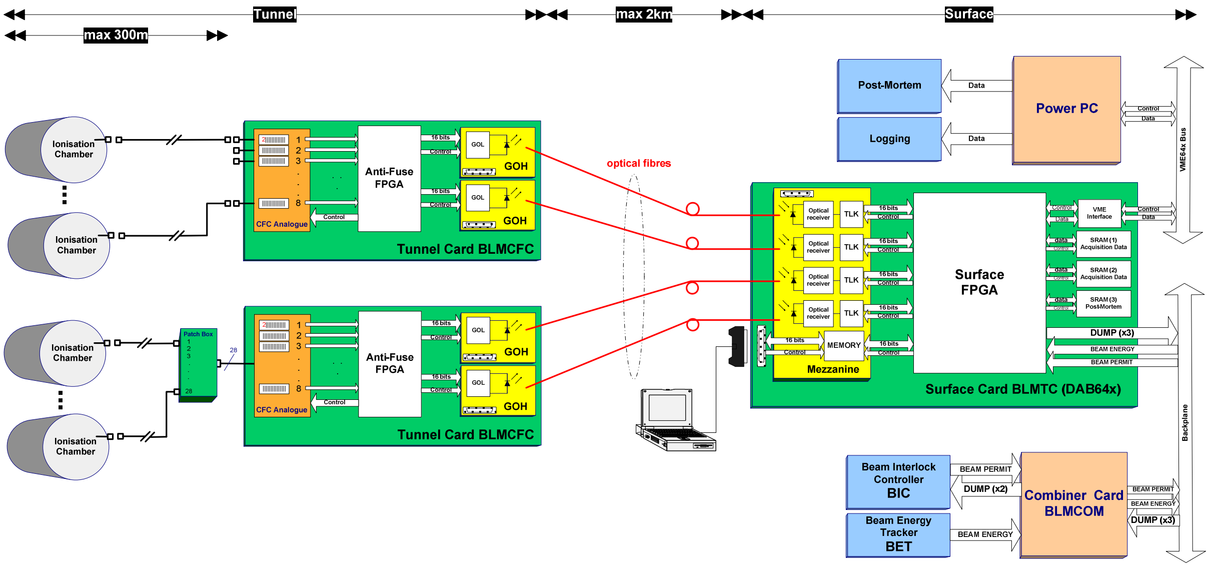


Figure 2-1: LHC Beam Loss Monitoring System Overview.

Around 3600 Ionization Chambers are the detectors of the system. A set of up to eight of them can be connected to each of the tunnel cards. In those tunnel cards, named BLMCFCs, the digitisation of the detector signal is done by using Current-to-Frequency converters (CFC) and Analogue-to-Digital converters (ADC). A Field Programmable Gate Array (FPGA) device acquires the digitised data and transmits them to the surface using Gigabit Optical Links. At the surface installations, the data are received by the data analysis modules, named BLMTCs.

The backbone of those cards, which is also an FPGA device, will analyse by keeping histories of those data and decide whether a beam dump request should be initiated by comparing those histories with predefined threshold values. Each surface card receives data from two tunnel cards, meaning that it can treat up to sixteen channels. Finally, the BLMTC cards are accommodated in VME crates that include Combiner cards for concentrating the beam dump requests and distributing the beam energy.

2.1 Beam Loss Detectors

The majority of the detectors of the BLM system will be ionisation chambers. For regions with very high loss rates, secondary emission monitors (SEM) are foreseen as well. The ionisation chamber consists of a stack of parallel electrodes, which is inserted in a stainless steel tube. The schematic drawing of the ionisation chamber used for the LHC can be seen in Figure 2-2. The volume of the chamber is filled with gas under normal pressure. The electrical field strength in between of the electrodes is 3 kV/cm. The only passive electronic components are a resistor and a capacitor of a low pass filter mounted at the feedthroughs of the chambers. This filter smoothes drift voltage variations and, in the case of a break down of the voltage power supply, it keeps the drift voltage almost constant at the electrodes of the chamber. The correct functioning of the ionisation chamber is therefore ensured for minutes after switch off, even in case of a break down.

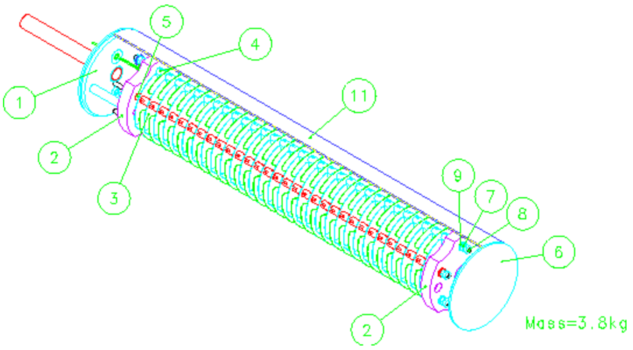


Figure 2-2: Drawing of the LHC ionisation chamber. [16]

The SEM detector will be based on the same design as the ionisation chamber. Instead of having the electrode volume filled with a gas it will be under vacuum and only one sensitive foil will be used to reduce the resolution. The schematic drawing of the SEM detector used for the LHC can be seen in Figure 2-3.

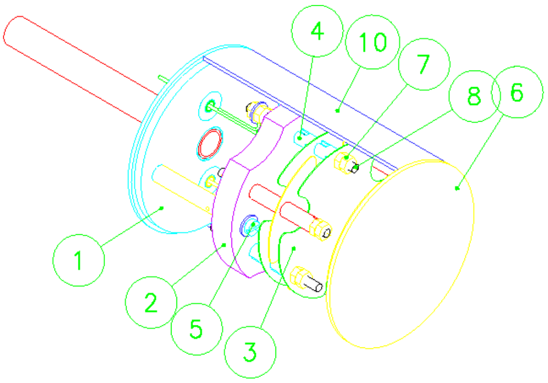


Figure 2-3: Drawing of the secondary emission monitor. [16]

It is foreseen to have one tube shape assembly with a short SEM part on one side and a longer ionisation chamber part on the other side. Both detectors will be mounted on the outside of the cryostat in the horizontal plane given by the two vacuum pipes of the LHC magnets. At this position, the secondary particle flux is highest and the best separation of the losses from the two beams is reached.

From shower simulations at the different loss locations [17], it was found that a set of six detectors around the quadrupoles is sufficient for localising the beam losses and to distinct them between the two beams. Figure 2-4 illustrates the arrangement of the detectors around the magnet, as well as their interconnection.

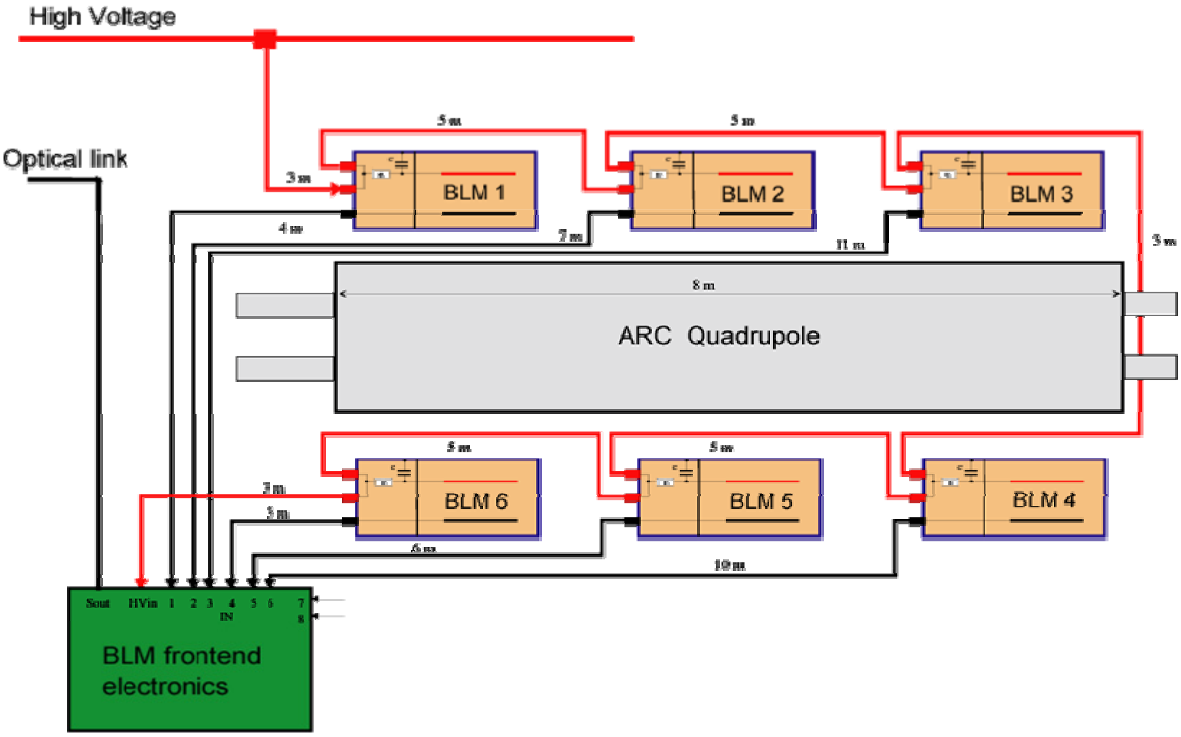


Figure 2-4: Arrangement of the Detectors around the Magnet. [18]

2.2 FRONT-END ELECTRONICS

The placement of the BLM front-end electronics for the Left and Right regions in the arc of each Interaction Point (IP) is in a crate below the quadrupole magnet, where also other systems' electronics are mounted. For the Middle regions of the straight sections, where increased levels of radiation are occurring, the signals are transported to the closest stub tunnel (alcove) before digitisation.

2.2.1 THE BLMCFC CARD

The FPGA used in the front-end electronics is from Actel. The 54SX72A device, belonging in the SX-A [19] family of Actel's devices, comes in a 208 pin PQFP (plastic quad flat pack) package and provides 72,000 system gates for building its functionality. It belongs to the one-time-programmable (OTP) type of devices using electrically fusible links, a feature Actel calls AntiFuse technology. While re-programmability is generally desirable, OTP technologies like that used in Actel's antifuse families provide other important performance, security, and radiation-tolerant features that are necessary for many critical designs like the BLM system.

The Current-to-Frequency Converter was built using off-the-shelves components. The basic elements comprising its functionality are an amplifier, the OPA627, a JFET, the J176, a Comparator, the NE521, and a monostable, the 74HCT123. More about the functionality achieved will be shown in the following section.

The AD74240 [20] is a Quad 12-bit ADC, thus, two of those CMOS devices will be necessary in the card to cover the eight connected detectors. It operates with a sampling speed of 40Ms/s and provides the data in a parallel output. The outputs from the ADC are in Low Voltage Differential Signaling (LVDS) levels using the Double Data Rate (DDR) scheme.

The signalling scheme used for the digital data output of the ADC uses the LVDS standard but the FPGA is built with CMOS level I/Os. This requires a level converter next to each ADC. An LVDS to CMOS transceiver ASIC is used to this purpose. The LVDS_MUX

[21] CMOS device can receive and convert eight LVDS signals, thus six of them will be used for each card.

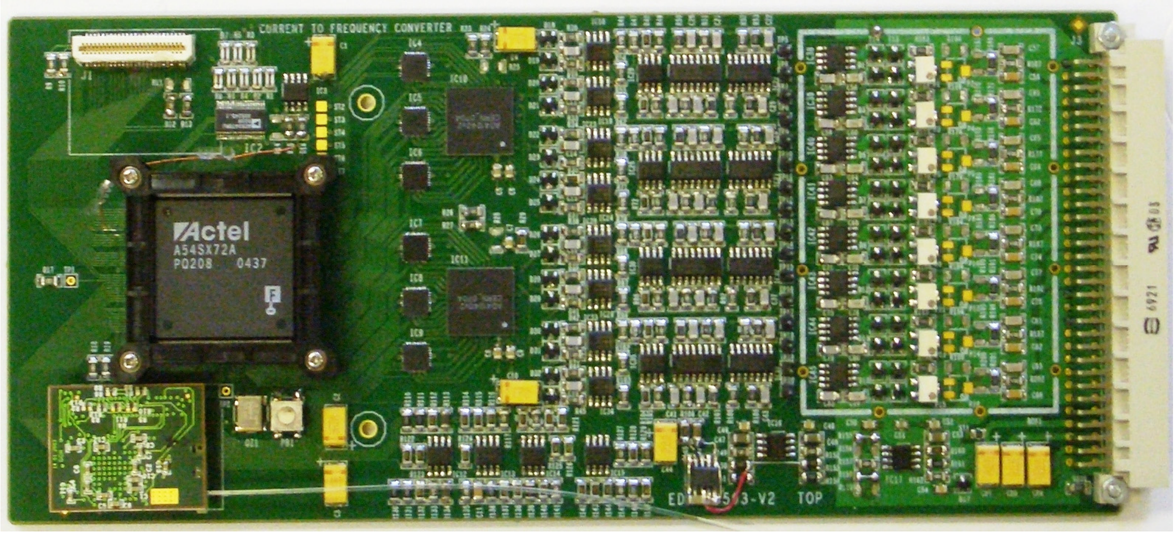


Figure 2-5: Picture of the Tunnel (BLMCFC) Card.

Two of the GOL (Gigabit Optical Link), also belonging to the CERN custom-made ASICs with radiation tolerant design, will be used for each card. They include the analogue parts needed to drive the laser, an algorithm running that corrects SEU and the output is providing the data already with 8b/10b encoding for either 16 or 32-bit input. It also provides error reporting, like number of SEU detected or loss of synchronisation, that will be used to discover failing components.

In Figure 2-5, it is shown a picture of the frond-end card, named BLMCFC, which accommodates all the above-mentioned components. More about the communication protocols used will be discussed in the following chapter.

2.2.2 CURRENT-TO-FREQUENCY CONVERTER (CFC)

The particle losses are measured, in the analogue part in the front-end card, and transmitted to the surface, where the final evaluation takes place. To measure the detector signal, a current-to-frequency converter (CFC) was designed. It works on the principle of balanced charge and is shown in Figure 2-6.

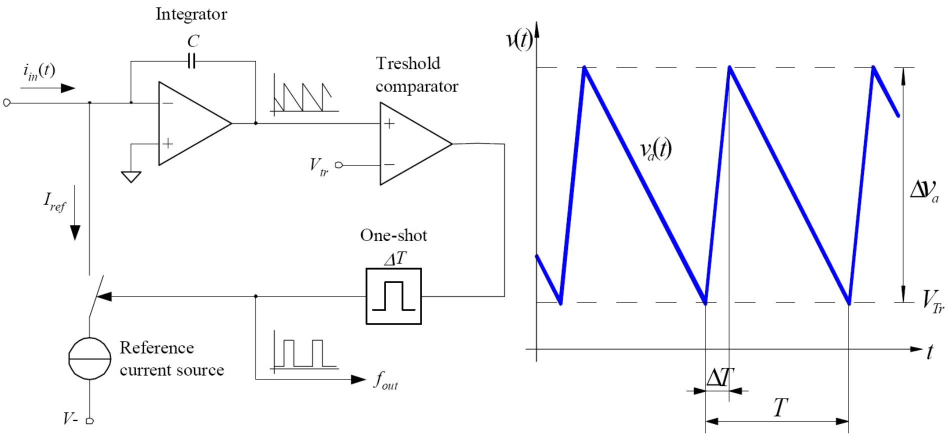


Figure 2-6: Principle of the balanced charge Current-to-Frequency Converter. [17]

The signal, i.e. the current induced from the detector, is integrated during the whole period T. If a constant chamber current is assumed, the integrator output ramps down with a constant slope. After reaching a threshold, the reference current *Iref* is induced into the summing node of the op-amp for a fixed time ΔT, driving the integrator output back again.

The output frequency is related to the chamber current by $f = \dfrac{\bar{i}_{in}}{I_{ref}\Delta T}$ [2-1]

2.2.2.1 CFC Characteristics

The characteristics of the CFC circuit designed and the ionisation chamber at use can be seen in Table 2-1. The use of the CFC was dictated mainly by the high dynamic range needed for the BLM system. The CFC has the ability to give a value of 5.00E+08, much higher than all other options evaluated in a previous study [22].

Table 2-1: Current-to-Frequency Converter and Ionisation Chamber Specifications.

CFC		Ionisation Chamber	
Frequency	0.01 - 5.00E+06 Hz	**Length**	19cm
Current	2.5 - 1.00E+06 pA	**Surface**	63cm^2
Dynamic	5.00E+08	**Volume**	1000cm^3
Resolution	200 pC/clock	**Resolution**	2500ch./Mip
	1.25E+09 elect./clock	**High Voltage**	1500V
	5.00E+05 Mips/clock	**Rise Time**	~100ns
Clock max	5.00E+02 clock/0.1ms	**Fall Time**	~100µs

The secondary particle flux at the location of the detectors, normalised to the number of inducing protons is called fluence. The fluence distribution expected in the LHC is sought to be resolved with simulations of proton impacts at the magnets. The uncertainty of these simulations is mainly determined by uncertainties in the shower developments and in the knowledge of the geometry. The current estimated minimum and maximum fluence levels can be found in Table 2-2.

Table 2-2: Estimated Fluence Levels at 0.45 and 7 TeV.

Fluence		0.45TeV		7TeV	
		Min	Max	min	Max
BL ARC	Mips/cm^2/p	5.00E-04	3.00E-03	8.00E-03	4.00E-02
BL	Mips/63cm^2/p	3.15E-02	1.89E-01	5.04E-01	2.52E+00

Applying the estimated fluence levels to the characteristics of the CFC and the Ionisation Chamber it is possible to calculate the number of counts expected to be collected for different time periods. Table 2-3 and Table 2-4 show a summary for the minimum and maximum number of counts expected, as well as, the number of counts needed to reach the magnet quench level for beam energies of 450GeV and 7TeV respectively.

Table 2-3: Number of Counts Produced at 0.45TeV.

0.45TeV		Fluence						
	p/m/s	Min	Max	Resolution	Int.time(s)	Resolution	No counts	
		(Mips/63cm2/p)		(ch/mip)		elct./count		
min. rate								
	5.00E+08	3.15E-02	1.89E-01	2.50E+03	1.00E-04	1.25E+09	0.00315	0.01890
	3.00E+08	3.15E-02	1.89E-01	2.50E+03	2.50E-03	1.25E+09	0.04725	0.28350
	4.00E+05	3.15E-02	1.89E-01	2.50E+03	1.00E+01	1.25E+09	0.25200	1.51200
	1.00E+05	3.15E-02	1.89E-01	2.50E+03	1.00E+02	1.25E+09	0.63000	3.78000
max. rate								
	2.00E+13	3.15E-02	1.89E-01	2.50E+03	1.00E-04	1.25E+09	126	756
	2.00E+12	3.15E-02	1.89E-01	2.50E+03	2.50E-03	1.25E+09	315	1,890
	1.00E+09	3.15E-02	1.89E-01	2.50E+03	1.00E+01	1.25E+09	630	3,780
Quench								
	2.58E+13	3.15E-02	1.89E-01	2.50E+03	4.00E-05	1.25E+09	64.99295	389.95767
	1.08E+13	3.15E-02	1.89E-01	2.50E+03	1.00E-04	1.25E+09	67.98236	407.89419
	1.19E+12	3.15E-02	1.89E-01	2.50E+03	2.50E-03	1.25E+09	187.55911	1,125.35468
	3.20E+09	3.15E-02	1.89E-01	2.50E+03	1.00E+01	1.25E+09	2,016.00000	12,096.00000
	9.50E+08	3.15E-02	1.89E-01	2.50E+03	1.00E+02	1.25E+09	5,985.00000	35,910.00000

Table 2-4: Number of Counts Produced at 7TeV.

7TeV								
		Fluence						
	p/m/s	Min	Max	Resolution	Int.time(s)	Resolution	No counts	
		(Mips/63cm2/p)		(ch/mip)		elct./count		
min. rate								
	5.00E+08	5.04E-01	2.52E+00	2.50E+03	1.00E-04	1.25E+09	**0.050400**	**0.252000**
	3.00E+08	5.04E-01	2.52E+00	2.50E+03	2.50E-03	1.25E+09	**0.756000**	**3.780000**
	4.00E+05	5.04E-01	2.52E+00	2.50E+03	1.00E+01	1.25E+09	**4.032000**	**20.160000**
	1.00E+05	5.04E-01	2.52E+00	2.50E+03	1.00E+02	1.25E+09	**10.080000**	**50.400000**
max. rate								
	2.00E+13	5.04E-01	2.52E+00	2.50E+03	1.00E-04	1.25E+09	**2,016**	**10,080**
	2.00E+12	5.04E-01	2.52E+00	2.50E+03	2.50E-03	1.25E+09	**5,040**	**25,200**
	1.00E+09	5.04E-01	2.52E+00	2.50E+03	1.00E+01	1.25E+09	**10,080**	**50,400**
Quench								
	2.18E+10	5.04E-01	2.52E+00	2.50E+03	4.00E-05	1.25E+09	**0.878753**	**4.393764**
	1.22E+10	5.04E-01	2.52E+00	2.50E+03	1.00E-04	1.25E+09	**1.229202**	**6.146011**
	6.05E+09	5.04E-01	2.52E+00	2.50E+03	2.50E-03	1.25E+09	**15.247176**	**76.235880**
	1.24E+07	5.04E-01	2.52E+00	2.50E+03	1.00E+01	1.25E+09	**124.992000**	**624.960000**
	8.26E+06	5.04E-01	2.52E+00	2.50E+03	1.00E+02	1.25E+09	**832.608000**	**4,163.040000**

Figure 2-7 shows a plot (using a logarithmic scale) of the number of counts, which will be needed to reach the quench level of a magnet for different integration times. It was calculated using the fluence levels estimated from simulations shown previously.

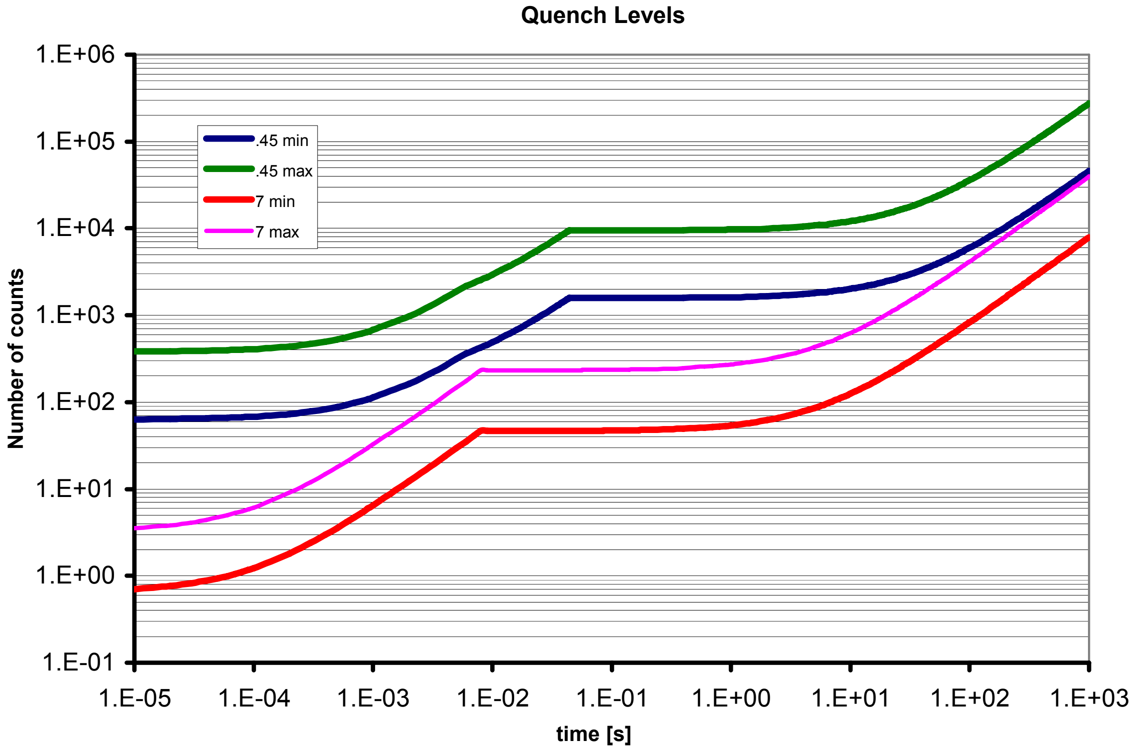
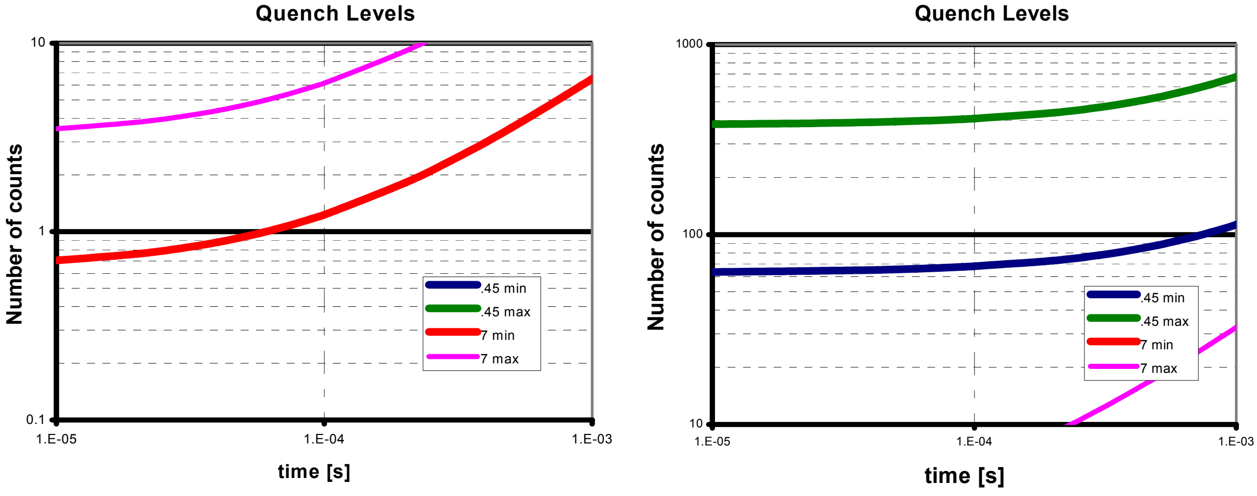


Figure 2-7: Quench Levels defined by number of counts.



(a) (b)

Figure 2-8: Quench Levels defined by number of counts (zoom in 100μs region).

The number of CFC output counts needed, when the accelerator has 7TeV of energy, to exceed the quench level is just around one (see Figure 2-8a), where in the similar case but for lower energies (see Figure 2-8b) it is significantly larger.

Reaching in some cases the quench level with a value that is equal to the resolution of the CFC output is a limitation that is only possible to be overcome by increasing the dynamic range of the system. For that reason an ADC was in addition decided to be used. The intention is to cover the region below the one count with the ADC and provide a more accurate reading when very slow losses are occurring. The idea lies in measuring the output of the integrator and thus knowing the discharge state of its capacitor at the time of the acquisition. Using, in later stages, some simple pre-processing of those data it will be possible to be merged with the data from the CFC counter.

2.2.3 FRONT-END FPGA PROCESSES

The functionality that the BLMCFC card will need to exhibit can be simply described as the acquisition of the data produced, by the counter and the ADC, and the production of the packet to be transmitted. A block diagram of the BLMCFC card functionality is illustrated in Figure 2-9.

In more detail, the main tasks assigned to be performed by its FPGA would be, to measure the frequency output of eight CFCs and read the digitized voltage of the feedback capacitor by the ADCs.

In addition, it will collect the status information for the high tension, for the system's temperature and for any possible CFC errors. It will count the number of packets transmitted and hold a unique number, to be used as the Card ID. Finally, all the above data collected will be multiplexed and word aligned for transmission.

In parallel to the data being forwarded to the optical link, a CRC encoding process will calculate a checksum for this packet and append it to provide error detection capabilities to the system. The collected packet transmission will be done through a gigabit optical link that the card will control. More about the data transmitted and the link utilised will be discussed in Chapter 3.

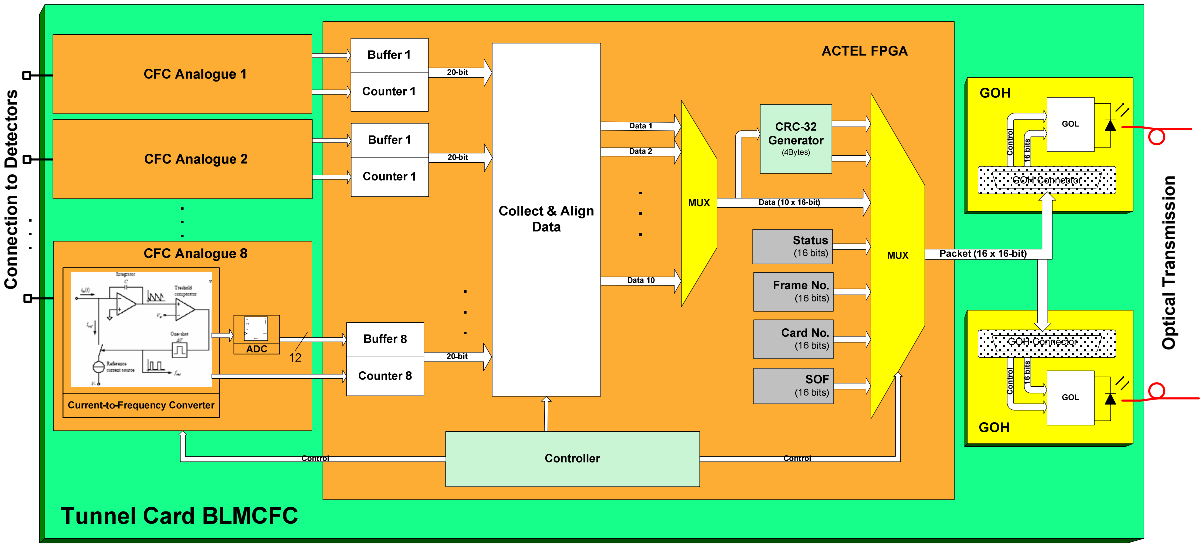


Figure 2-9: Block Diagram of the BLMCFC.

2.3 OPTICAL FIBRES

Optical fibres will be used to support the backbone of the data communication networks in the LHC. The optical fibres are installed in mini-tubes, which may contain up to 70 fibres each. The mini-tubes are inserted in protective ducts, which can be directly buried in surface trenches or laid on cable trays, where such infrastructure is available. Fibre "blowing" technology is used wherever possible in order to facilitate fibre maintenance and future extensions. [23]

The cables situated in the LHC tunnel will be subject to radiation, which will damage the fibre over time. It is expected that the light transmission capability of the fibre will deteriorate so much that it must be replaced in the most exposed zones of the machine after every 3-5 years of operation at nominal beam intensity. The installation technique, using mini-tubes contained in ducts, lends itself to rapid replacement of defective fibres. The optical characteristics of the fibres in the main fibre paths will be continuously monitored. Based on trend analysis from the monitoring system, the replacement of defective fibres can be planned within the normal maintenance windows during machine operation. The BLM system will perform a similar monitoring of all its optical components. This procedure will be discussed in Chapter 8.

For the transmission of the BLM data from the tunnel to the surface installation, it is foreseen to multiplex the six signals from the detectors around every quadrupole magnet with two spare channels. This is done mainly to allow the possibility of adding two more "mobile" ionisation chambers, if at a later stage that is found necessary.

This will result in 368 optical fibres, to cover the left and right regions, that include two spare channels per fibre, and 197 optical fibres to cover the middle regions of the LHC. All the data transmissions for the BLM system were decided to have some sort of redundancy as a measure of improving the reliability and the availability of the whole system. In this context, the optical communication channel has been doubled making the total number of optical fibres needed for the BLM system equal to 1130.

2.4 Signal Reception and Processing Electronics

A VME card, that provides the processing power, and a mezzanine card, that links the tunnel with this surface installation, comprise the BLMTC processing module.

Its main task will be to analyse the acquisition data, by keeping a history of those data and calculating various moving sum windows for each detector, and decide whether a dump request should be initiated, by comparing those histories with predefined threshold values. Additionally, it will need to include the back-end of the optical links. Each module will need to host the optical receiver parts for four optical links, meaning that it can treat up to 16 channels.

Those modules will establish communication with the VME-bus but at the same time will be able to work autonomously for protection against main CPU failures. Data will be sent over the VME-bus for on-line viewing and storage by the Logging and Post-Mortem systems.

2.4.1 The DAB64x Card

The data analysis card is based on the general purpose PCB that was implemented for the whole Beam Instrumentation group, named DAB64x [24]. It is comprised from an Altera Stratix [25] FPGA, an Altera MAX CPLD [26] for power-on configuration and VME functionality, and three SRAM memories. The functionality of the system is realised by using different firmware on the FPGA and CPLD devices, and different mezzanine cards that can be placed on either of the six general-purpose connectors available.

The whole Stratix family of FPGAs offers various devices that their features can include from 10,570 to 79,040 Logic Elements, packages that have from 672 to 1,508-Pin FBGAs (FineLine Ball Grid Array) and up to 7,427,520 RAM bits in embedded memory blocks. The DAB64x card was designed with the intention to serve as a processing module for various systems except the BLMS.

For this reason, it was proposed the implementation of the PCB layout to support the possibility of a later exchange of the device within the same family, a feature called vertical migration. Vertical migration [27] means that it is possible to migrate to devices whose dedicated pins, configuration pins, and power pins are the same for a given package across different device densities. Table 2-5 shows the different package options provided by the Stratix family. It was finally decided to allow migration between the EP1S10, EP1S20, EP1S25, EP1S30, and EP1S40 devices in the 780-pin FBGA package.

Table 2-5: Stratix Package Options & I/O Pin Counts. [27]

Device	672-Pin BGA	956-Pin BGA	484-Pin FBGA	672-Pin FBGA	780-Pin FBGA	1,020-Pin FBGA	1,508-Pin F BGA
EP1S10	345		335	345	426		
EP1S20	426		361	426	586		
EP1S25	473			473	597	706	
EP1S30		683			597	726	
EP1S40		683			615	773	822
EP1S60		683				773	1,022
EP1S80		683				773	1,203

The FPGA to be used for the BLM system will be the EP1S40, which has 41,250 LE, 615 user I/O pins, and 400KBytes of internal memory as available resources for the system to be built from. At the same time, the Beam Position Monitor (BPM) system, for example, will be able to use the same module but populated with the EP1S20 device. Table 2-6 shows the FPGA features for each of the devices available to be used in the DAB64x card.

Table 2-6: Stratix Device Family Features. [27]

Feature	Device				
	EP1S10	**EP1S20**	**EP1S25**	**EP1S30**	**EP1S40**
Logic Elements	10,570	18,460	25,660	32,470	41,250
M512 RAM blocks (32×18bits)	94	194	224	295	384
M4K RAM blocks (128×36bits)	60	82	138	171	183
M-RAM blocks (4K ×144 bits)	1	2	2	4	4
Total RAM bits	920,448	1,669,248	1,944,576	3,317,184	3,423,744
DSP blocks	6	10	10	12	14
Embedded multipliers	48	80	80	96	112
PLLs	6	6	6	10	12
Maximum user I/O pins	426	586	597	597	615

Additionally, the BLMTC module includes three of the AS7C33512PFS32A/36A [28] high-performance CMOS 16-Mbit synchronous Static Random Access Memory (SRAM) devices. Each of them is organized as 524,288 words x 32/36 and it incorporates a two-stage register-register pipeline for higher frequency throughput.

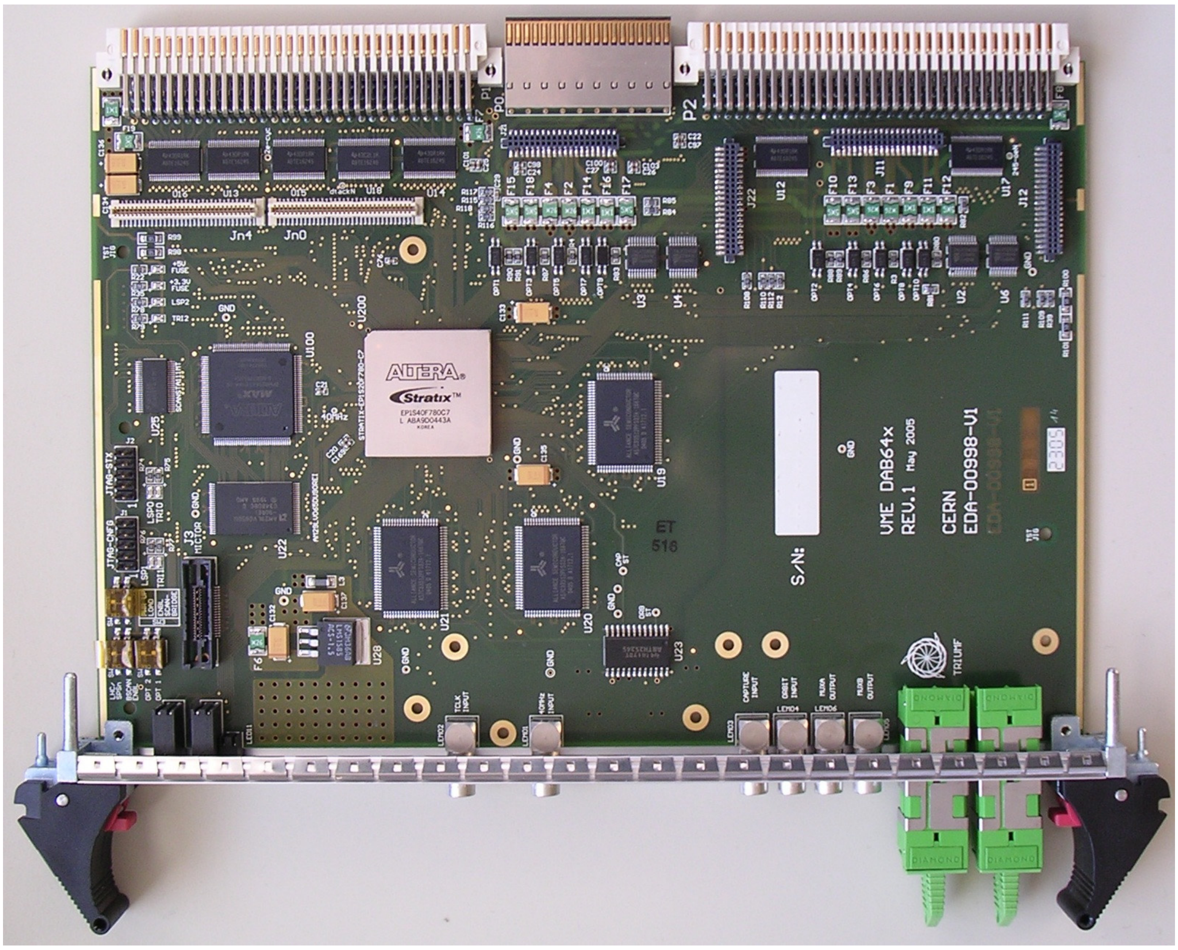


Figure 2-10: Picture of the BLMTC Processing Module.

Figure 2-10 shows a picture of the final version of the DAB64x card. The front-panel is mounted with handles to allow easier insertion to the VME64x backplane of the three 5-row connectors, as well as the two double E2000 adapters for optical fibre separations.

2.4.2 THE BLM MEZZANINE CARD

The packet will be transmitted through the optical link will arrive first to a mezzanine card. The intention is to include only the photodiode and only necessary electronics on the mezzanine in order to keep it as minimal as possible. The synchronisation and decoding could be assigned to be executed by the FPGA placed on the card but instead a more dedicated chipset (that includes an 8B/10B decoder) is added in the mezzanine for this process.

As a result, the BLM mezzanine card was constructed to include all necessary components for the four gigabit optical receivers and a Mbit of Flash memory for system specific data. The data are delivered to the DAB64x card through two connectors. Those connectors used for the mezzanine are 2 x 64pin PMC connectors that will provide the power supply signals, that is, the 3.3V, 5V and GND, as well as, connection to 114 FPGA I/O pins. A detailed explanation of the choices used and the final implementation of the BLM mezzanine can be found in the following Chapter 3.

The advantage of keeping those design constraints and adding a mezzanine card is that the main card will maintain the universality to be used for more applications except the BLM.

2.4.3 SURFACE FPGA PROCESSES

The processes assigned to the surface FPGA will be described in detail on the following chapters. Figure 2-11 illustrates an overview of the main BLMTC processes in the FPGA.

The RCC (Receive, Check and Compare) process (discussed in Chapter 4) will receive, de-serialise and decode the transmitted packets. It will check for errors on both transmissions and compare the packets in order to select one error-free packet.

The Data-Combine process (discussed in Chapter 5) will receive the two types of data, the counter and the ADC data, coming from the same detector and will merge them into one value, filtering at the same time noise passing through the ADC circuitry.

The combined values from each detector will be given, each time they become available, to the SRS (Successive Running Sums) process. This process, discussed in Chapter 6, produces and maintains a number of histories, in the form of moving sum windows, for each detector.

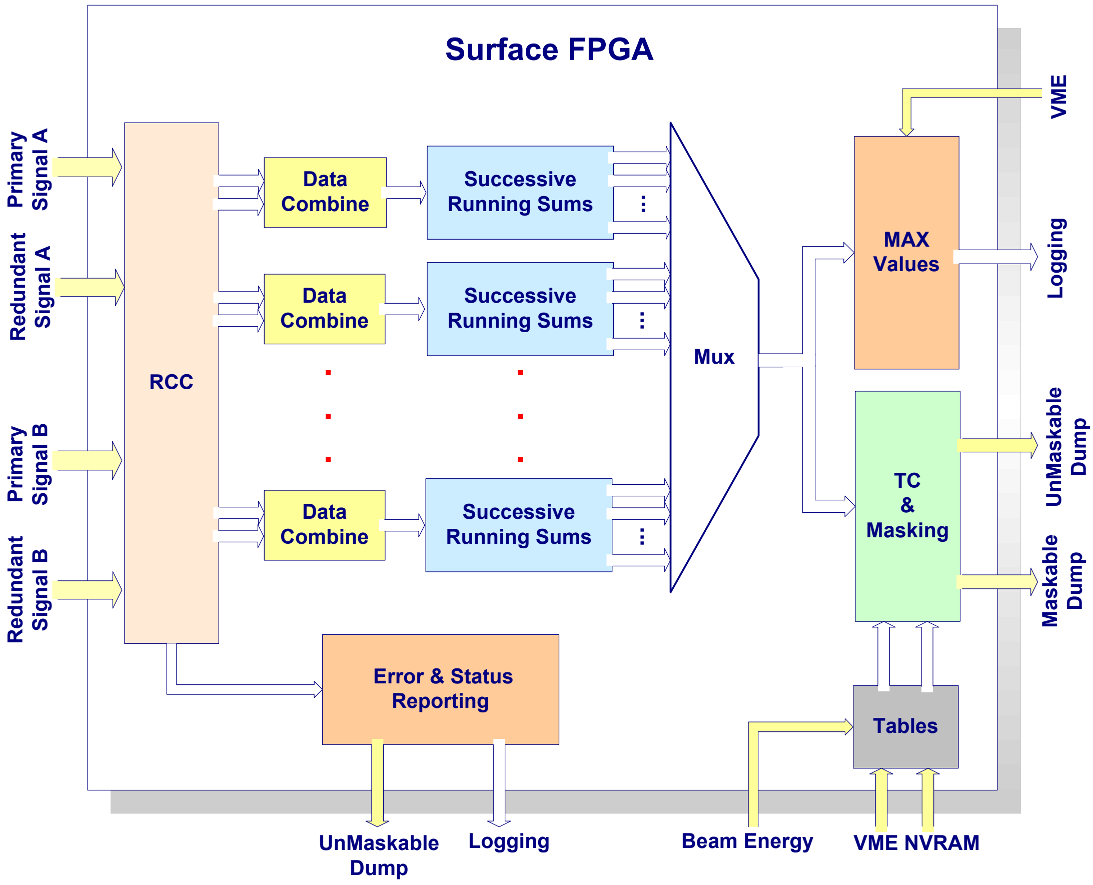


Figure 2-11: Block diagram of processes running in the surface FPGA.

Those sums are compared with their corresponding threshold limits in the TC (Threshold Comparator) process every time they become updated. If one or more of them are found to be higher, a dump request is signalled. All dump requests are gathered initially by the Masking process, which serves the purpose of distinguishing between “Maskable” and “Not-Maskable” channels. The information necessary for the operation of the TC and the Masking processes, that is, the threshold and the masking values, are stored in tables uniquely created for each card. Those two processes together with their tables are discussed in Chapter 7.

Finally, for supervision and logging purposes, two more processes are created and are discussed in Chapter 8. The MaxValues process, which calculates and keeps the maximum values given in a period of time by each moving sum window, and the ESR process, which reports the errors found in the transmissions and the status of the tunnel electronics. Both of them are collected by the crate PowerPC and projected in the Control Room displays.

2.5 Output Signal Concentration

The two beam permit lines, one for the Maskable and another for the Not-Maskable channels, are daisy chained through each BLMTC card using a custom-made backplane for the crates. The final receiver of those lines is the Combiner card located at the last slot of the crate. If any of the 16 BLMTC cards accommodated in the crate decides to break any of the lines a beam dump request will be given to the LHC Beam Interlock System (BIS) through the de-assertion of the Beam Permit signal. As an additional use, those lines will be used by the Combiner card to provide a continuous supervision of the operation of the BLMTC cards in the crate. Thus, it will be able to discover immediately a disconnection from the circuit or its power failure and a dump will be requested for any of those cases.

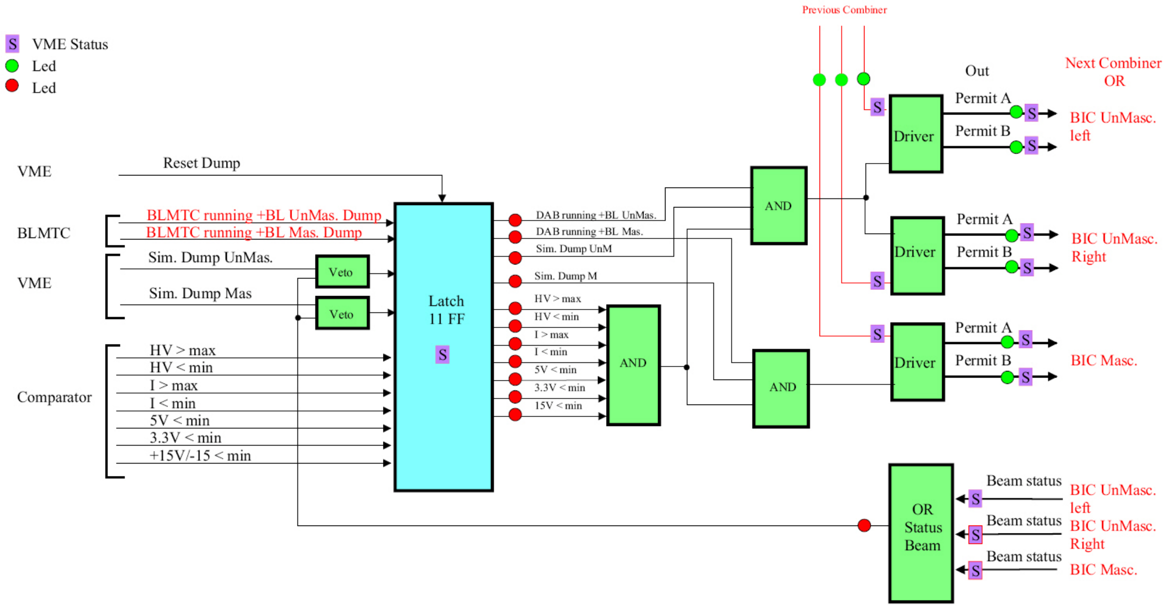


Figure 2-12: Block Diagram of Combiner Card. [29]

Figure 2-12 illustrates a block diagram of the Combiner card. It can be observed that additional inputs from supervisory circuits for the powering of the Ionisation Chambers and the crate or VME initiated requests can also become responsible for requesting a beam dump.

The LHC Beam Interlock System [30] is designed with the ability to receive beam dump requests separately for the two LHC rings. This feature will be used by the BLM system to add an additional redundancy to its already doubled outputs. Figure 2-13 illustrates the configuration foreseen for the BLM system at each LHC point. The Combiner cards from the three crates will be daisy chained and a patch box will distribute the beam permit signals to BIS modules. Two modules will be used to transmit the signals for the Not-Maskable and one for the Maskable channels.

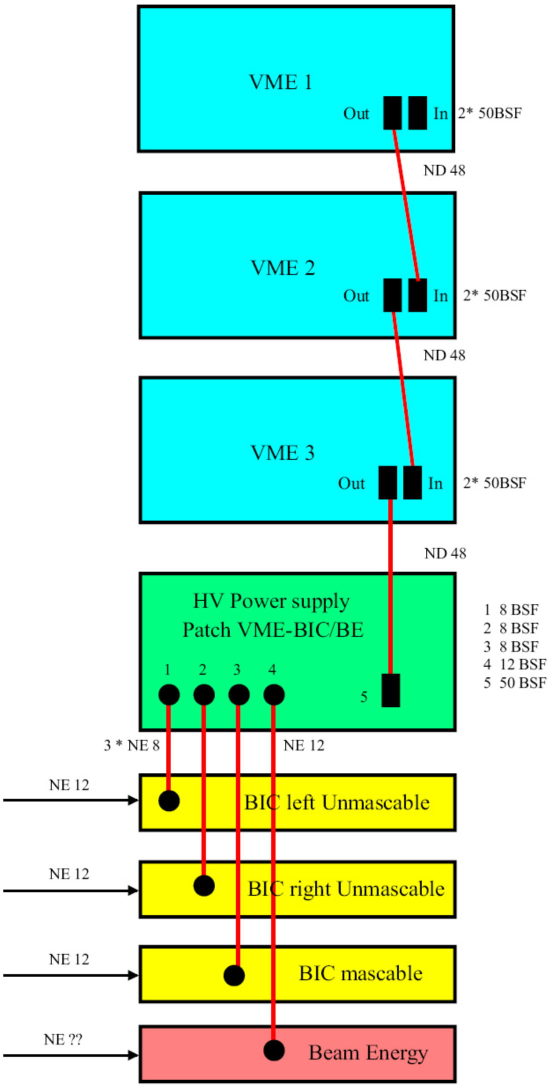


Figure 2-13: Beam Loss VME to BIC\BE Connection. [29]

Finally, the Combiner card will receive the beam energy from the Beam Energy Tracking System's (BETS) module, previously discussed in Section 1.4.2, using a parallel connection and distribute it in the same way to all the BLMTC cards using dedicated VME backplane signal lines.

2.6 THE BACK-END CRATES

The CPU card, the Combiner card and the BLMTC processing modules are accommodated in VME (VERSAmodule Eurocard) crates. The VME standard became the industrial bus of choice in the 80’s and since then has been utilised in thousands of applications. It is also a standard widely accepted in CERN.

In the late 80's, the VME's draft standard was expanded for 64 bit data and address capability, which also doubled the throughput. Locks, Configuration ROM / Control & Status Registers (CR/CSR), rescinding DTACK*, auto system control detection, auto slot ID, plus optional shielded DIN connectors were also added. These additional features effectively transformed VME from an 80's bus to a 90's bus, which allows VME to be used in even more demanding applications for the early 90's. This standard is commonly referred to as VME64.

In 1993 the VITA Standards Organization (VSO) agreed to publish the VME64 Standard. It was also agreed to use additional standards to add features as they are agreed upon by the VSO membership. This standard is a collection of additional features as agreed upon during 1994, 1995 and the first half of 1996.

For the BLM system, 25 crates will be used in total, three at each point except point 7 that will have an extra crate facilitating the detectors observing the Collimator system. All of them with the extended VME64 [31] specification. Its extension defines a set of features that can be added to VME and VME64 boards, backplanes and subracks. Some of the features included are a 160-pin connector, a P0 connector, and geographical addressing.

On the first slot of the crate, the bus master and system arbiter is placed. Its functionality is realised by a PowerPC running the LynxOS real-time operating system. It is able to communicate to the outside world through a gigabit Ethernet link. The BLM system will depend on the crate CPU only for its non-critical processes, for example to collect the Logging and Post-Mortem data from each card.

2.7 Reliability and Availability Enhancement

In order to guarantee the correct operation and the fail-safety of the BLM system, that is, for the system to have a much higher probability of preventing superconductive magnet destruction caused by a beam loss, many additional measures have been taken.

2.7.1 Tunnel System

To ensure the Ionisation Chamber connection, high voltage modulation tests will be initiated when no beam is present. By introducing a sine wave signal in the high tension (see Figure 2-4), it will be possible to check the connections and the acquisition system at low currents.

The CFC card's correct and continuous operation will be ensured by applying a constant current of ~10pA. That will produce a constant reading in the surface electronics of approximately three counts per 20 seconds, which will be checked if present. At the same time the Status monitor, running on the tunnel card, will check that the high tension and the temperature are well inside specified limits and will send its latest information inside every transmitted packet.

Table 2-7: Radiation Dose withstood without Error.

Part Name	Remarks	Integral Dose (KGy)
CFC	Off-the-shelf components	0.5
ACTEL	One time programmable FPGA	3.2
AD41240	Custom-made ASIC	10
LVDS_MUX	Custom-made ASIC	10
GOH	Custom-made ASIC and Laser Diode	3.14

Since the tunnel electronics will be placed in a radioactive environment the SEU probability was minimised with the use of radiation tolerant components, custom made ASICs, and FPGAs that do not need configuration data. For the rest of the components, where no other option was available, radiation qualification has been made in the test beam facilities of the Paul Scherrer Institut (PSI) in Villigen, Switzerland, and the Centre de

Recherches du Cyclotron in Louvain La Neuve, Belgium, for each of them. Table 2-7 shows the integral dose limit specified for each of them.

In addition, TMR (triple modular redundancy) [32] [33] techniques have been used where possible resulting in the tunnel FPGA's design many processes to have been tripled with output voting, as well as, doubled or tripled the most important I/Os. For example, the Counter inputs have been tripled and the Data and Control outputs have been doubled.

2.7.2 SURFACE SYSTEM

To ensure a reliable communication link a double, that is, redundant optical link has been utilised as a first measure. The error free reception of the packet will be ensured using the CRC-32 error check algorithm and augmented by the 8b/10b encoding scheme.

It will be shown in Chapter 4 that the CRC is able to discover any error with a length less than the length of CRC and for longer bursts the probability of undetected error is small enough (non-detection probability $Pr = 1.16415*10^{-10}$). The 32-bit checksum of the CRC will be additionally used for comparison of the redundant packets. The 8b/10b encoding will provide not only the clock data recovery (CDR) and a DC-balanced serial stream, but also additional error detection capability.

Extra information embedded in the packet will allow continuous self-checking of the system from problems resulting from user errors or aging. For example, to avoid misplacement of the Threshold or Masking table the Card ID transmitted on each packet will be used. Each tunnel card holds a unique 16-bit number that will be compared with the one loaded together with the tables. Similarly, to avoid loss of data, the Frame ID will be also included on each transmitted packet. It will be a 16-bit number that will increment at every transmission. The Surface FPGA will be able in that way to compare consecutive transmissions and checks for missing frames.

Finally, to ensure system failures and dump request recognition the surface cards outputs carrying the beam dump information will use a frequency signal. At a dump request, reset, or failure the transmitted frequency will be altered, thus, always signalling the beam abort.

2.8 SIMULATION AND HARDWARE VERIFICATION TOOLS

Verification is today the most time-intensive function in the design process. For many design situations, system-level verification is not practical due to long simulation times. In addition, board-level debug has become problematic due to the difficulty of accessing signals at the pins. This situation has become even worse, when using the new BGA-type packages, where test signals could not be routed at all.

2.8.1 STATIC TIMING ANALYSIS

As designs become more complex, the need for advanced timing analysis capability grows. Static timing analysis is a method of analysing, debugging, and validating the timing performance of a design. Timing analysis measures the delay of every design path and reports the performance of the design in terms of the maximum clock speed.

The Quartus II Timing Analyzer [34] can calculate a matrix of point-to-point device delays, determine the setup and hold time requirements at the device pins, and calculate the maximum clock frequency. It was preferred from other EDA tools simply for the simplicity it provided. All the design entry tools are integrated with the Timing Analyzer, which allowed by simply tagging the start and the end points in the design files to determine the shortest and longest propagation delays. In addition, the Message Processor, one more embedded feature, can locate and display critical and failed paths.

2.8.2 SIMULATION TESTS

Static timing analysis does not check design functionality but needs to be used together with simulation to verify the overall design operation. The Quartus II Waveform Editor [35] was used to create and edit waveform design files, as well as input vectors for simulation and functional testing. The Waveform Editor also functions as a logic analyser that allows viewing of the simulation results.

Throughout the whole verification phase, it was the most used tool, simply for the reasons that it allows to copy, cut, paste, repeat, and stretch waveforms; to combine waveforms into groups that display binary, octal, decimal, or hexadecimal values; to compare two sets of simulation results by superimposing one set of waveforms on another; and to annotate files with comments, which served later also for the documentation.

All the final simulation tests, samples of them will be shown in the simulation sections of the relevant chapters, have been carried out using the delays calculated by the static timing analysis. The resulting files from the analysis, calculated after the place-and-route of the process, were back-annotated into the simulation tool to provide the most accurate results.

2.8.3 SIGNALTAP II LOGIC ANALYSIS

For many designs on devices with high I/O BGA packages, system-level verification is very time-consuming and sometimes extremely difficult. The SignalTap II logic analysis [36] facilitated the verification process by integrating the functionality of a logic analyser within the FPGA.

Altera claims that its SignalTap II embedded logic analyzer supports the most channels, fastest clock speeds, largest sample depths available, and the most advanced triggering capabilities available in an FPGA or any other programmable logic embedded logic analyzer.

Nevertheless, it allows to capture a device’s internal node or I/O pin state in real time, operating at system speeds. By working in real time, it ensures that the design will function as specified, under real operating conditions. Its most appreciated feature during the verification cycles was that it could incrementally change or modify which nodes the analyser is monitoring without performing a full design recompile. A feature that saved many hours, which would have been wasted in recompilations, when attempting to discover the source of an error or a strange behaviour of parts of the system.

The SignalTap solution consists of the Quartus Waveform Editor software, the SignalTap megafunction, and Altera's MasterBlaster communications cable. The selection of the signals wished to be observed is done through the Waveform Editor. The last options needed to be defined are the trigger points to be used to capture the data and the clock signal to be used for the sampling. A SignalTap megafunction is then automatically inserted into the design by the Quartus software, like any other synthesisable intellectual property (IP) core. Using the graphical user interface, it is possible thereafter to control the acquisition process. The captured data are stored in the Stratix device's embedded memory blocks for streaming to the Quartus software through the communications cable. The Quartus software then analyses the signals and displays the results on the Waveform Editor. The megafunction has a nominal effect on the size of the design, and can be removed when the analysis is completed.

Whenever a new process was designed, it was verified initially on its own with the use of test vector inputs. When the results taken were found to match the expected functionality, it was added to the rest of the system's implemented processes.

The DAB64x card, used for testing the processes in real-time, was available at that time with the EP1S30 FPGA device. In most of the cases, all the embedded M-RAM memory blocks, which are the largest available types having 4K x 144bits, were assigned to be used by the SignalTap analyser. Similarly, the sampling frequency used throughout the verification cycle was outputted from an embedded PLL that doubled the system clock input. The resulting 80MHz sampling clock was found adequate for most of the cases with the exception of rare initial tests using a 160MHz clock, extracted in the same way. Even though the fastest sampling would provide the most accurate results, a more moderate speed provides longer observation windows for the analysis since the memory size is already fixed. That is additionally supported by the fact that a synchronous design approach was kept throughout the system, meaning that it is only necessary to observe the states around the clock edges.

2.8.4 In-System Updating of Memory & Constants

One more capability embedded in the Quartus II software have been used extensively, named "In-System Updating of Memory & Constants" [37]. It enables FPGA memory contents and design constants to be read or written in-system without recompiling the design or reconfiguring the rest of the FPGA. Actually, it is possible to read, with a slow rate though, and change the memory contents while the FPGA is at full operation in the end system through the Joint Test Action Group (JTAG) interface.

REFERENCES

[16] B. Dehning, G. Ferioli, J.L. Gonzalez, G. Guaglio, E.B. Holzer, C. Zamantzas, *"The Beam Loss Monitoring System"*, LHC Project Workshop - Chamonix XIII, 2004.

[17] E. Gschwentdner, B. Dehning, G. Ferioli, W. Friesenbichler, V. Kain, *"The Beam Loss Detection System of the LHC Ring"*, Presented at EPAC-02, 2002 Particle Accelerator Conference, Paris - France, CERN SL-2002-021 BI.

[18] G. Ferioli, "The Beam Loss Installation", BL Section, unpublished.

[19] Actel Corp. *"SX-A Family FPGAs"*, version 3.0, 2001. [online: http://www.actel.com]

[20] G.Minderico et al, *"A CMOS low power, quad channel, 12 bit, 40MS/s pipelined ADC for applications in particle physics calorimetry"* presented at the 9th Workshop on electronics for LHC, 2003. [online: http://lhc-electronics-workshop.web.cern.ch/LHC-electronics-workshop/2003/sessionsPDF/Eleccal/MINDERIC.PDF]

[21] G. Cervelli, "*CMS Tracker LVDS MUX v3*", [online:http://cmstrackercontrol.web.cern.ch/cmstrackercontrol/documents/LVDSMUX3_datasheet.pdf]

[22] W. Friesenbichler, *"Development of the Readout Electronics for the Beam Loss Monitors of the LHC"*, Diploma thesis in Mechatronics Engineering, Fachhochschule Wr.Neustadt, Jun 2002.

[23] CERN, "LHC Design Report volume.2 : The LHC Infrastructure and General Services", Geneva 2004 - CERN-2004-003-V-2; ISBN: 92-9083-224-0

[24] R. Jones, *"VME64x Digital Acquisition Board for the LHC Trajectory and Closed Orbit System"*, LHC Engineering Specification, LHC-BP-ES-0002.

[25] Altera Corp. *"Altera Stratix Device Family Overview"*, [online: www.altera.com/products/devices/stratix/overview/stx-overview.html]

[26] Altera Corp. *"Altera MAX 3000A CPLD Family Overview"*, [online: www.altera.com/products/devices/max3k/overview/m3k-overview.html]

[27] Altera Corp. *"Stratix Device Handbook, Volume 1"*, July 2005, [online: http://www.altera.com/literature/hb/stx/ch_1_vol_1.pdf]

[28] Alliance Semiconductor Corp., Part Number: AS7C33512PFS32A-36A, [online: http://www.alsc.com/pdf/sram.pdf/fs/as7c33512pfs32_36a.v2.3.pdf]

[29] G. Ferioli, *"The Combiner Card for the BLM System"*, BL Section, unpublished.

[30] LHC Engineering Specification, *"The Beam Interlock System for the LHC"*, Document No: LHC-CIB-ES-0001-00-10, EDMS No:.567256, 17-02-2005

[31] VMEbus International Trade Association, *"American National Standard for VME64 Extensions"*, ANSI/VITA 1.1-1997, Approved October 7, 1998, [online: http://www.vita.com]

[32] Katz, Barto, et. al., *"SEU hardening of field programmable gate arrays (FPGAs) for space applications and device characterization,"* IEEE Trans. Nuclear Science., vol. NS-41, no. 6, pp. 2179-2186, July, 1994.

[33] D. A. Zacher, *"Advanced Synthesis Techniques for Radiation Hardened Antifuse Programmable Logic Design"*, Mentor Graphics, March 2003.

[34] Altera Corp. *"Quartus II Timing Analysis"*, version 5.1.0, Oct 2005, [online: http://www.altera.com/literature/hb/qts/qts_qii53004.pdf]

[35] Altera Corp. *"Introduction to QUARTUS II"*, [online: http://www.altera.com/literature/manual/intro_to_quartus2.pdf]

[36] Altera Corp. *"SignalTap II Embedded Logic Analyzer"*, [online:http://www.altera.com/products/software/products/quartus2/verification/signaltap2/sig-index.html]

[37] Altera Corp. *"In-System Updating of Memory & Constants"*, version 5.1.0, Oct 2005, [online: http://www.altera.com/literature/hb/qts/qts_qii53012.pdf]

03

Data Transmission

Chapter 3. Data Transmission

The acquisition part, i.e. the CFC card, of the BLM system is placed in some cases as far as 2km away from the processing module, i.e. the BLMTC card. The additional requirement for low latency in the response of the system is forcing the use of a high-speed optical link. Moreover, the fact that the CFC card will be placed in a radiation environment complicates more the transmitting electronics and only qualified parts can be used that guarantee its tolerance and preserve its availability.

The data, transmitted through the optical link, will arrive first to a mezzanine card. The reception, synchronisation, decoding and clock data recovery could be assigned to be executed there and provide the data directly to the FPGA. The intention is to include the photodiodes and the necessary electronics for this task on a mezzanine in order to keep the processing card as versatile as possible to allow its usage for other applications except the BLM system.

In this chapter, a more detailed view is shown of the physical link chosen, as well as, the structure of the information decided to be transferred from the acquisition to the processing part of the BLM system.

3.1 THE BLM DATA PACKET

The BLM data packet, that is, the transmitted data from the CFC card after every acquisition, will consist of not only the eight ionisation chambers' data but also some additional bits. Some of them are necessary for the correct reception and decoding of the data, others for checking the status of the chambers (power and error state), and finally some for the transmission errors detection. The proposed general structure of the transmitted packet is illustrated in Figure 3-1.

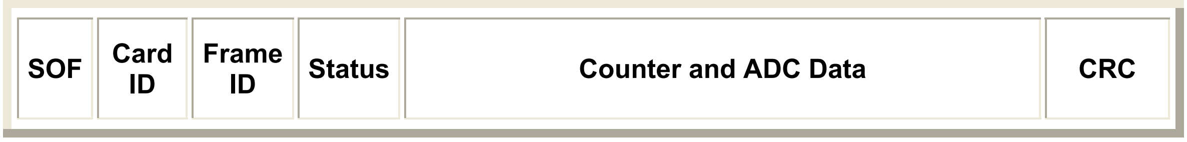


Figure 3-1: Structure of the BLM Packet.

From the dynamic range needed to be covered by the BLM system and the transmission rate chosen it was found that the packets should convey from each detector 8 bits from the Counter and 12 bits from the ADC.

Table 3-1: Formatting of the Status Word.

Bit	Name	Default	Remarks
1	Status_P5V	1	Status +5V supply ('1' when above 4.75V, otherwise '0')
2	Status_M5V	1	Status –5V supply ('1' when below –4.75V, otherwise '0')
3	Status_P2V5	1	Status 2.5V supply ('1' when above 2.3V, otherwise '0')
4	Status_HV	1	Status HV supply ('1' when above VHV > 1326V)
5	TEMP1	1	Temperature < 50C
6	TEMP2	1	Temperature < 60C
7	GOH_Ready_1	1	GOH 1 is ready
8	GOH_Ready_2	1	GOH 2 is ready
9	TEST_CFC	0	TEST_CFC not started (VHV >1584V)
10	TEST_CFC_ON	0	TEST_CFC is not running
11	RST_DAC	0	Reset of DAC (VHV > 1700V)
12	DAC_over_155	1	DAC is over 155bit
13	DAC_overflow	1	DAC reached 255 (clear with SysReset)
14	DAC_ resested	0	DAC was reset (clear with SysReset)
15	Integrator_Level	0	All working = '0', one or more above 2.4V = '1'
16	Reserved	1	Not assigned

Apart from those, the transmitted packet will include the following additional information:

- The SOF (start of frame) word, a control word that will be used by the receiving electronics to initiate the beginning of the packet.
- The Card ID, a unique number embedded on each of the CFC cards' FPGAs to identify the correct inter-connections, a feature expected to be found very helpful also during the installation, where more than a thousand fibres will need to be connected correctly.
- The Frame ID, a 16-bit incremental number by each transmission, that can help to identify lost packets. It will additionally be used to check if the correct packet is compared with its redundant partner.
- The Status word, collected and appended on each packet, to be used for the continuous monitoring of the tunnel electronics. Table 3-1 shows the formatting and the information included in the Status word. During "normal" operation the system should receive 0xFF19 (= b"1111 1111 0001 1001"), that is, the standard word transmitted when the system and its conditions are under the threshold limits, and no test or reset has been initialised.
- Lastly, at the end of the packet two words will carry the calculated CRC to be used by the error detection process.

Table 3-2: Summary of Data included in the Transmission Packet.

Data	Width (Bits)	Remarks
Start of Frame	16	SOF value = 0xF7F7
Card ID	16	~ 800 CFC cards uniquely numbered.
Frame ID	16	An incremental number on each frame.
Status	16	Bits showing the status of the CFC card.
Counter	64	Counter data
ADC	96	ADC data
CRC	32	Cyclic Redundancy Check/Error Detection
TOTAL	256	Total (in bits)
	16	Total (in 16-bit words)

Table 3-2 summarises the investigation on the data needed to be transmitted in each packet. It is showing the number of bits found necessary for each type of data.

3.1.1 FORMATTING OF THE BLM PACKET

The Counter and the ADC data are proposed to be sequentially forwarded to the transmitting electronics from each detector. This scheme is expected to ease the collection of the data in the CFC FPGA. It will allow the transmission of the word as soon as it has been collected minimising the space and the time that the data remain in the radiation environment, thus, minimising the SEU probability.

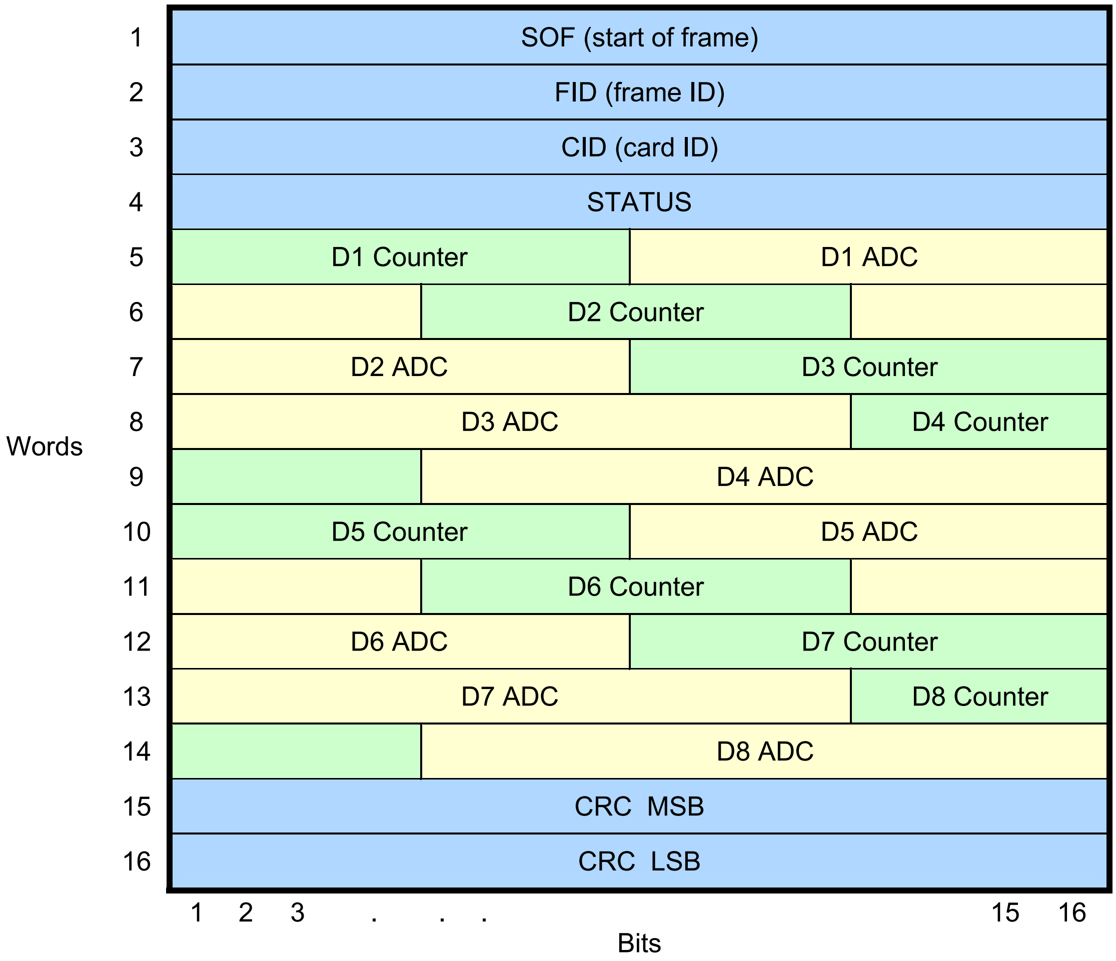


Figure 3-2: Data Arrangement in the BLM Packet.

Figure 3-2 illustrates the proposed arrangement of the data in the BLM packet. The data from the Counters and the ADCs in many cases are split between two words. In all cases, the preceding word holds the most significant bits of the data split. A realignment will be done by the receiving electronics that will reconstruct the data.

3.2 Physical Components

A plethora of devices and complete solutions exist nowadays in the market, which are able to provide and sustain high-speed optical transmission. Their usage though is not possible in the BLM system because of the radiation environment in the LHC tunnel. Figure 3-3 illustrates the proposed implementation of gigabit optical transmission for the BLM system. In the following sections more details for the components used will be given.

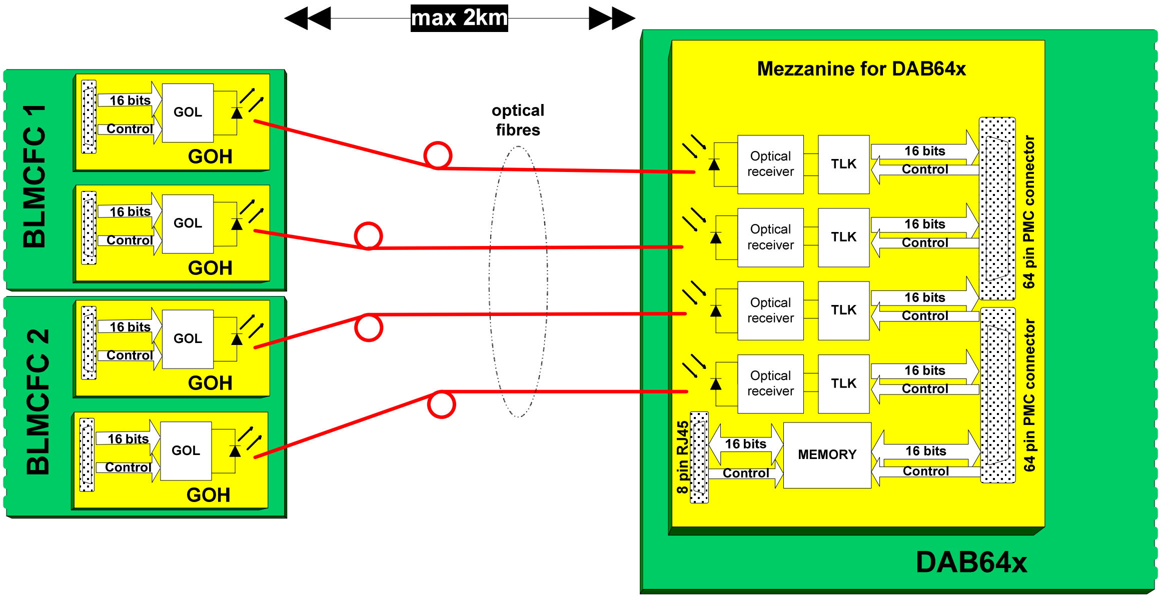


Figure 3-3: Gigabit Optical Transmission for the BLM System.

3.2.1 The GOL (Gigabit Optical Link)

The GOL (Gigabit Optical Link) [38] chip, is a multi-protocol high-speed transmitter ASIC, developed by the CERN Microelectronics group. A block diagram of its functionality is illustrated in Figure 3-4. The IC supports two transmission protocol standards, the G-Link [39] and the Gigabit-Ethernet [40].

The GOL chip can sustain transmission of data at both 800Mbit/s and 1.6Gbit/s. In those rates is included an overhead of 2 bits every 8 bits from either of the two encoding possibilities. Thus, the resulting effective data payload become 640Mbit/s and 1.28Gbit/s respectively. The 8B/10B is the encoding scheme chosen to be used in the BLM case and its operation is explained in the following chapter.

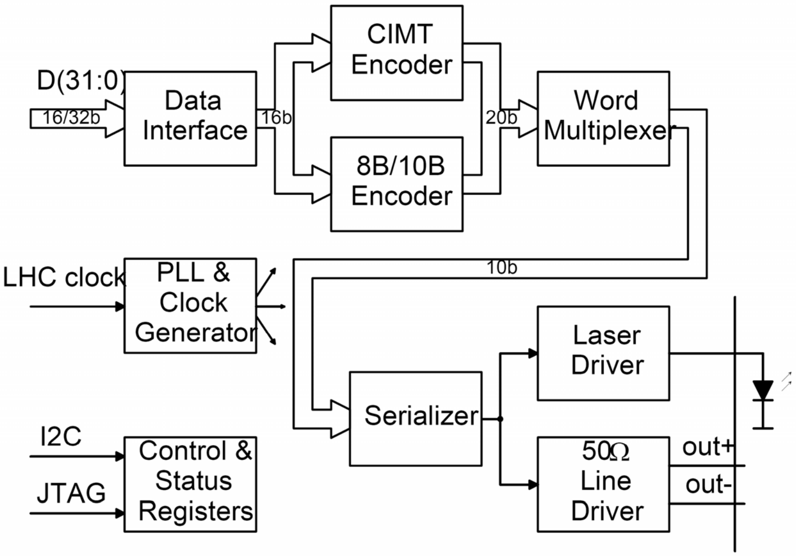


Figure 3-4: Block Diagram of the GOL Device. [38]

One of the GOL design requirements was to be able to withstand radiation doses compatible with the LHC detectors' radiation environment, thus, this ASIC was implemented in a 0.25μm CMOS technology employing radiation tolerant layout practices. It is designed to prevent or recover from Single Event Upsets with minimal impact on data by using supervisory circuits and TMR (Triple Modular Redundancy) techniques. Due to radiation effects, it is expected that the threshold current of the laser diodes will increase with time. To compensate for this, the laser-driver contains an internal pre-bias current generator that can be programmed to sink currents between 0 and 55mA. [43]

The radiation tolerance of the GOL chip has been tested for a total dose of up to 10 kGy and the worst estimated SEU rate for running the link in the CMS experimental cavern is 2.2 x 10^{-8} errors/component hour (1.3 x 10^{-4} for ECAL). The estimates are based on SEU measurements made with 60 MeV protons and heavy ions [42]. The laser driver bias current generator can be adjusted via I2C or JTAG to compensate for possible radiation damage in the laser and for the subsequent rise in its threshold current. [43]

3.2.2 THE GOH (GIGABIT OPTO-HYBRID) CONFIGURATION

The GOH or Gigabit Opto-Hybrid is composed of a connector, the GOL transmitter, a laser diode and passive components all mounted on a 2.4 cm × 3.0 cm 6-layer FR4 substrate. Using its small PCB to PCB mounting connector it can be plugged onto the main card, in this case the CFC card. Pictures of the Gigabit Opto-Hybrid configuration can be seen in Figure 3-5.

The optical transmitter used in the GOH was originally developed for use in the CMS Tracker analogue optical links. In order to avoid repeating the qualification process the same technology has been applied to these links. It consists of an edge-emitting laser diode producing light at a wavelength of 1310nm. It has an optical fibre pigtail terminated with an E2000 connector. An impedance matching network has been developed to couple the laser to the GOL, opening the eye and increasing the margin in the link.

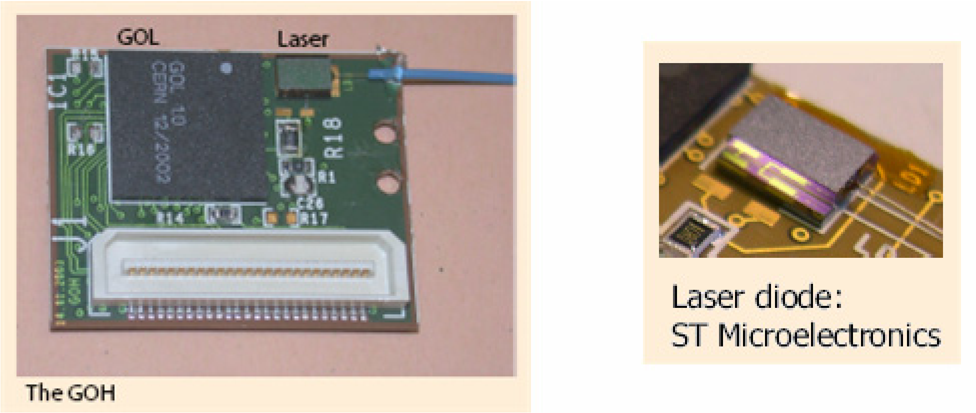


Figure 3-5: The Gigabit Opto-Hybrid Configuration. [41]

The GOH has been extensively irradiated [45] and has shown tolerance up to 3.2 kGy. The BLM front-end electronics will be in a much less harsh environment but have a critical function to perform and thus their resilience to radiation must be guaranteed.

3.2.3 THE TLK TRANSCEIVER

Since the receivers do not require any type of radiation hardness, commercial devices will be used together with GOL IC's to assemble the complete data links. The proposed device, the TLK1501 [46] is a member of the TLK transceiver family of multigigabit transceivers used in ultrahigh-speed bidirectional point-to-point data transmission systems.

The TLK1501 supports an effective serial interface speed of 0.6Gbit/s to 1.5Gbit/s, providing up to 1.2Gbit/s of effective data bandwidth, a direct match to the GOL's slow transmission mode. The TLK1501 is both pin-for-pin compatible with and functionally identical to the TLK2501, a 1.6 to 2.5Gbit/s transceiver, that could be exchanged if higher data speeds will be needed in the future.

Parallel data loaded into the transmitter is delivered to the receiver over a serial channel, which can be a coaxial copper cable, a controlled impedance backplane, or, similar to this case, an optical link. It is then reconstructed into its original parallel format.

The transmitter latches 16-bit parallel data at a rate based on the supplied reference clock. The 16-bit parallel data are internally encoded into 20 bits using an 8B/10B encoding format. The resulting 20-bit word is then transmitted differentially at 20 times the reference clock rate. The receiver section performs the serial-to-parallel conversion on the input data, synchronizing the resulting 20-bit wide parallel data to the extracted reference clock. It then decodes the 20 bit wide data using 8-bit/10-bit decoding format resulting in 16 bits of parallel data at the receive data terminals. The outcome in the TLK1501 is an effective data payload of 480Mbit/s to 1.2Gbit/s (that is, 16 bits of data multiplied by the clock frequency).

Table 3-3: Receive Status Signals.

RECEIVED 20 BIT DATA	RX_DV/LOS	RX_ER
IDLE (<K28.5, D5.6>, <K28.5, D16.2>)	0	0
Carrier extend (K23.7, K23.7)	0	1
Normal data character (DX.Y)	1	0
Receive error propagation (K30.7, K30.7)	1	1

Furthermore, it includes a loss of signal detection circuit for conditions where the incoming signal no longer has a sufficient voltage amplitude to keep the clock recovery circuit locked. To prevent a data bit error from causing a valid data packet from being interpreted as a comma and thus causing the erroneous word alignment by the comma detection circuit, the comma word alignment circuit is turned off after the link is properly established.

Two output signals, RX_DV/LOS and RX_ER, are generated along with the decoded 16-bit data output. The output status signals are asserted as shown in Table 3-3. When the TLK1501 decodes normal data and outputs the data, RX_DV/LOS is asserted and RX_ER is deasserted. When the TLK1501 decodes a K23.7 code (0xF7F7) indicating carrier extend, RX_DV/LOS is deasserted and RX_ER is asserted. If the decoded data is not a valid 8-bit/10-bit code, an error is reported by the assertion of both RX_DV/LOS and RX_ER. If the error was due to an error propagation code, the data output gives 0xFEFE. If the error was due to an invalid pattern, the data output is undefined. When the TLK1501 decodes an IDLE code, both RX_DV/LOS and RX_ER are deasserted and a 0x50BC code is sent on the output.

3.2.4 THE NTPPT-3 PHOTODIODE

The NTPPT-3 series consist of a reliable InGaAs PIN photodiode with trans-impedance amplifier and decoupling capacitor. This photodiode is suitable for receiving the light having a wavelength range of 1000 to 1650nm. It features high quantum efficiency, low dark current and low capacitance for high-speed optical fibre communications. [47]

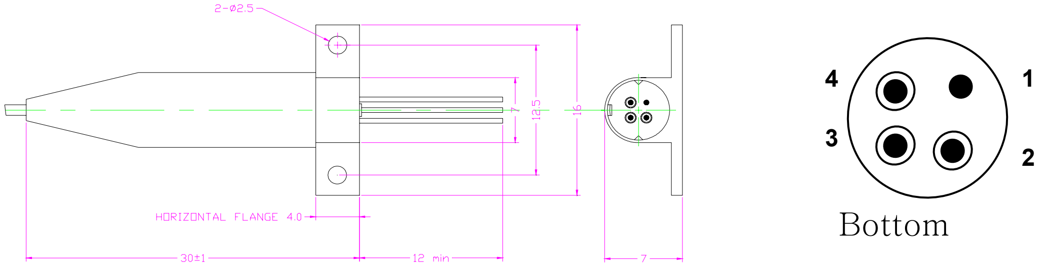


Figure 3-6: NTPPT-3 series PIN Photodiode with horizontal flange. [47]

It can support effectively data rates of up to 1.25GBit/s, matching the above mentioned electronics. The manufacturer will provide a custom-made configuration of this photodiode pigtailed with an E2000 connector. Its schematic view can be seen in Figure 3-6.

3.2.5 THE NON-VOLATILE MEMORY

The M25P10-A [48], proposed device, is a 1 Mbit (128K x 8) Serial Flash Memory, with advanced write protection mechanisms, accessed by a high speed SPI-compatible bus.

The memory can be programmed 1 to 128 bytes at a time, using the Page Program instruction. It is organized as 4 sectors, each containing 256 pages. Each page is 128 bytes wide. Thus, the whole memory can be viewed as consisting of 1024 pages, or 131,072 bytes. The whole memory can be erased using the Bulk Erase instruction, or a sector at a time, using the Sector Erase instruction.

The manufacturer states in the specifications that more than 100,000 Erase/Program Cycles per Sector can be performed and a 20-year data retention.

3.3 IMPLEMENTATION OF THE BLM MEZZANINE CARD

On the far end of the optical transmission, a daughter card was constructed with the main purpose of receiving the four optical signals, and providing them, converted into digital signals, to the BLMTC card.

The mezzanine's connector placement and signal definition are based on the draft standard for a Common Mezzanine Card Family (CMC) [49]. This standard defines the mechanics for a common set of slim mezzanine cards that can be used on VME, VME64 & VME64x boards, CompactPCI boards, MultibusI boards, MultibusII boards, desktop computers, portable computers, servers and other similar computer applications. Mezzanine cards based on this draft standard can be used to provide modular front panel I/O, backplane I/O or general function expansion for the host computer.

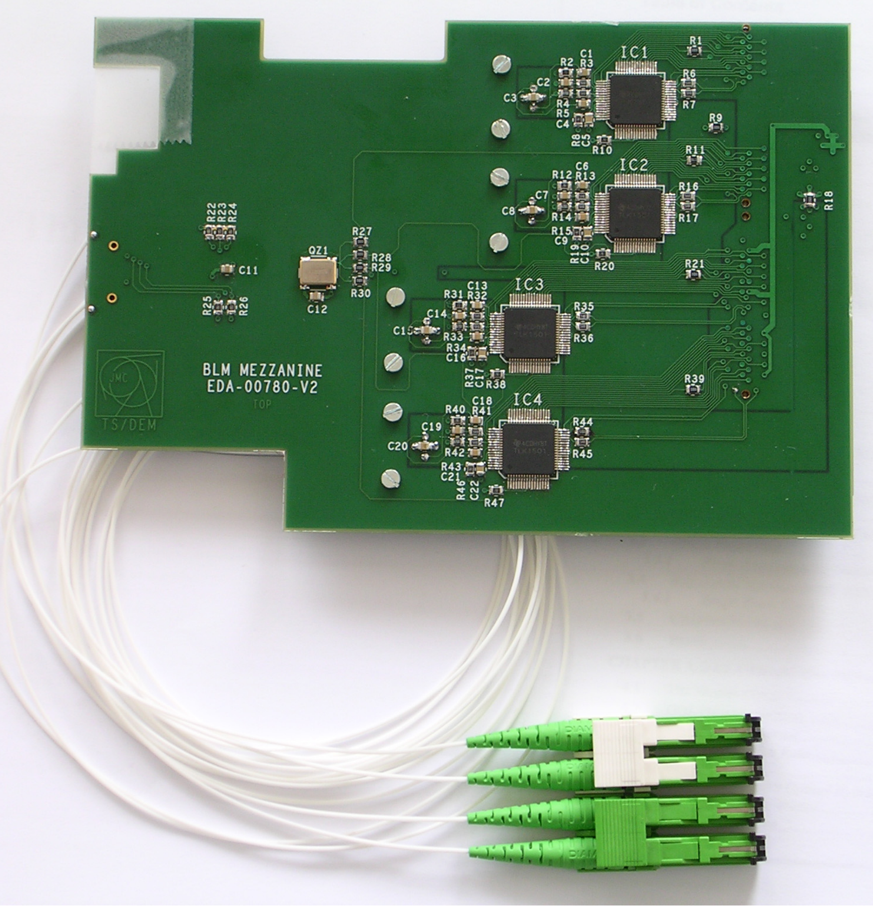


Figure 3-7: Top view of the BLM Mezzanine PCB.

The differential output of the photodiode is received by the TLK together with a reference clock signal provided by an oscillator also included in the mezzanine. The TLK devices are configured only as receivers by placing the transmission control and input pins to either GND or VCC as necessary since there is no bidirectional communication. The photodiode with the TLK transceiver configuration is reproduced four times, once for each optical signal to be received.

In addition, a non-volatile memory has been placed to hold the threshold and masking tables. Those tables and their purpose are discussed in Chapter 7. This Serial Flash Memory can be accessed, when reading or writing of its contents is needed, either by the PMC connectors or by a front-panel connector using the SPI bus [50] protocol.

In Figure 3-7 and Figure 3-8, can be seen the second version of the BLM Mezzanine card and its schematics can be found in the Appendix B.

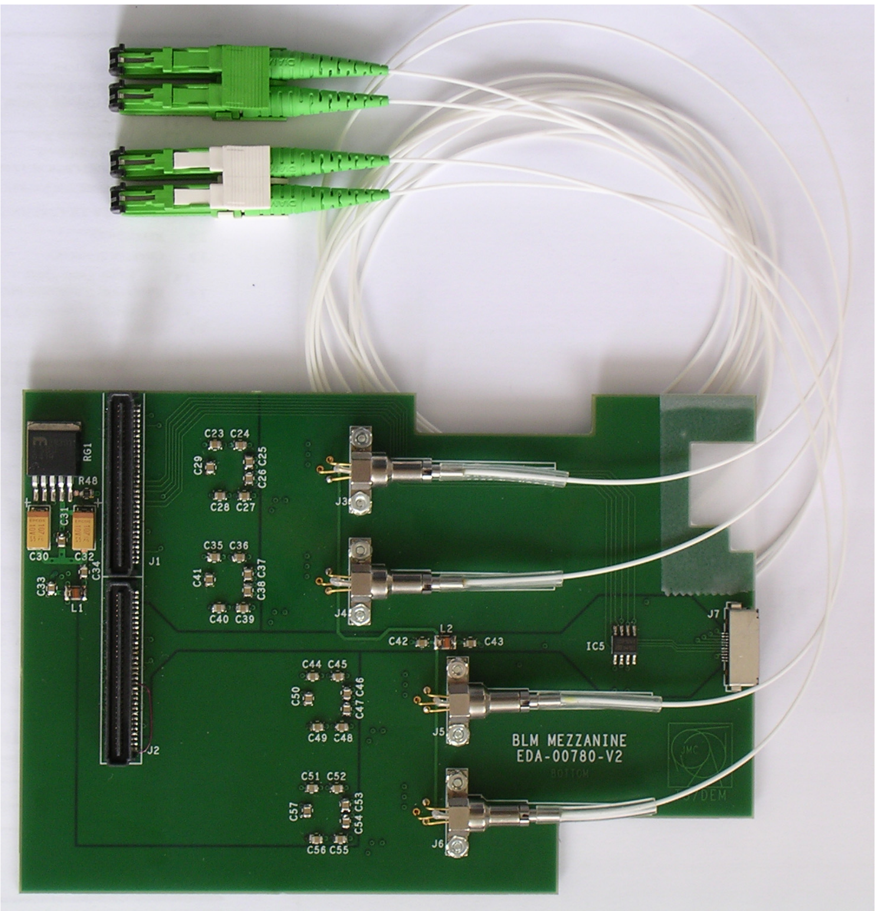


Figure 3-8: Bottom view of the BLM Mezzanine PCB.

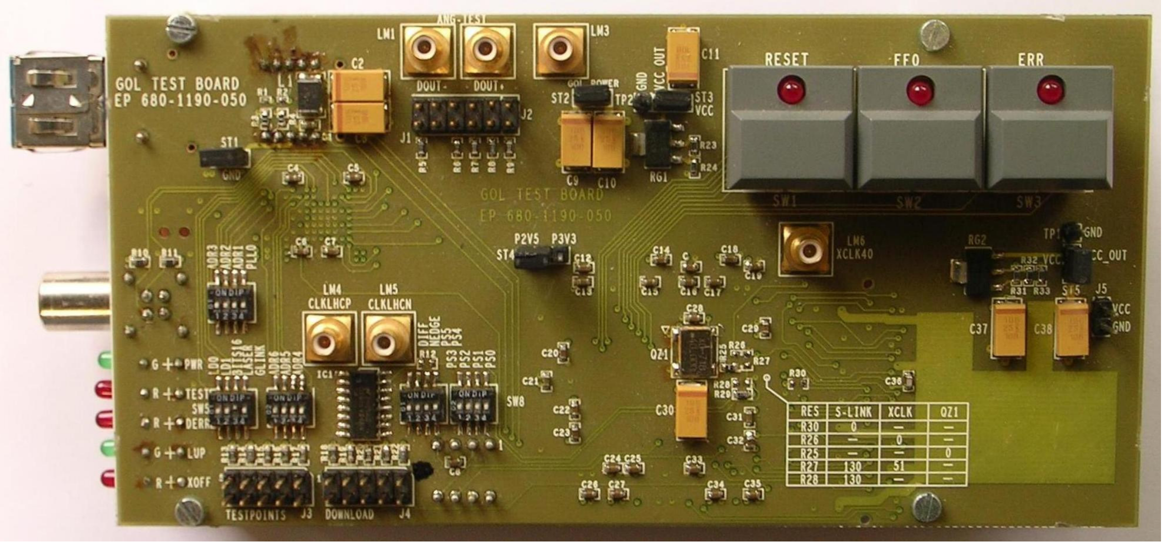
3.4 VERIFICATION OF THE BLM MEZZANINE CARD

In many cases, for the development of a system it is necessary the design to be partitioned between designers or even teams of designers. Each part has to be verified and proven to be under the specifications individually not only for the later seamless integration of the complete system but also because parts might become available at different periods in time. A similar situation has been experienced here and an emulation board has been used before the final system test.

3.4.1 USING THE GOL TEST BOARD

For the initial verification of the mezzanine card, a GOL Test Board [51] was used to emulate the CFC card. This test board has been implemented, by the CERN Microelectronics Group, during the verification of the GOL chip. With relatively simple modifications it could provide an identical gigabit optical signal to that expected from the CFC card output.

The modifications of the tester included replacement of the laser with one that transmits at 1310nm for monomode optical fibres and changes in the CPLD firmware in order to transmit in the formatting of the BLM packets. On the far end, a logic analyser was comparing the TLK outputs with a vector file equal to the one transmitted. The test was successful but only one link was possible to be tested at each time and had the additional uncertainty of the correct emulation of the CFC card.

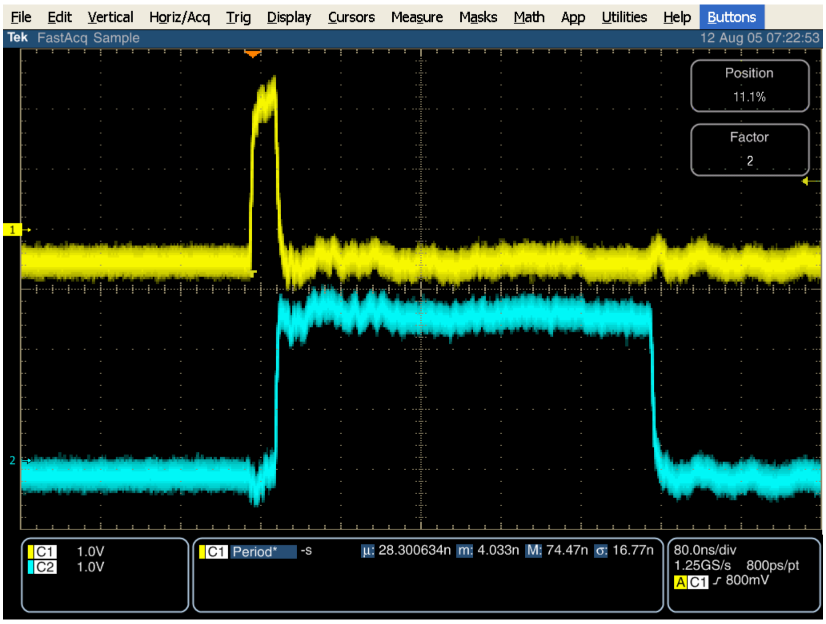


RES	S-LINK	XCLK	QZ1
R30	0	—	—
R26	—	0	—
R25	—	—	0
R27	130	51	—
R28	130	—	—

Figure 3-9: The GOL Test Board.

3.4.2 USING THE CFC CARD

Later, when the CFC cards became available, a full load test with all four links working was conducted. A high performance oscilloscope has been used, in this case, for probing the gigabit serial signals and its control flags in order to help in any troubleshooting. The following figures (Figure 3-10, Figure 3-11 and Figure 3-12) have been outputted by this oscilloscope. The voltage resolution in all cases is set to 1 V/div.



C1 1.0V
C2 1.0V

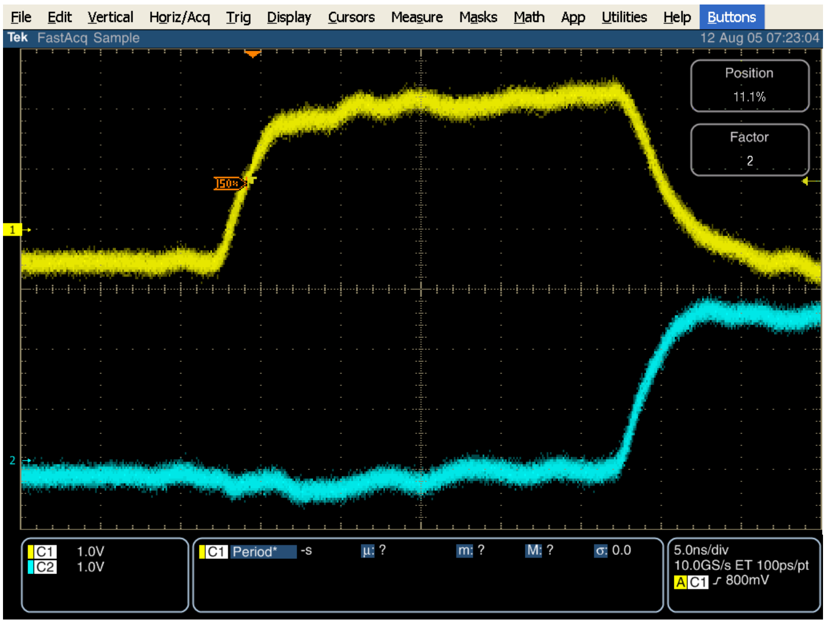
C1 Period* -s µ: 28.300634n m: 4.033n M: 74.47n σ: 16.77n

80.0ns/div
1.25GS/s 800ps/pt
A C1 ⎍ 800mV

Figure 3-10: Oscilloscope / Control Flag Outputs for a Read Cycle.

Figure 3-10 shows the case where the TLK decodes normal data and starts to output them. The start of a new frame is signalled by the assertion of the RX_ER flag. In the next clock cycle the RX_DV/LOS is asserted and the RX_ER is deasserted signalling that valid data can be read from the output. The RX_DV/LOS signal is correctly deasserted one clock cycle before the complete packet is received. (The time resolution used is 80 ns/div and the system's clock cycle is 25ns)

Figure 3-11 is a zoom in the transition from the start of frame to the valid signal state in order to investigate the rising and fall times of the flag signals. The time resolution used is 5 ns/div. Using these settings it is verified that the RX_DV/LOS is asserted and the RX_ER is deasserted in less than 2ns.


File Edit Vertical Horiz/Acq Trig Display Cursors Measure Masks Math App Utilities Help Buttons
Tek
50%
1
2

FastAcq Sample 12 Aug 05 07:23:04

Position 11.1%

Factor 2

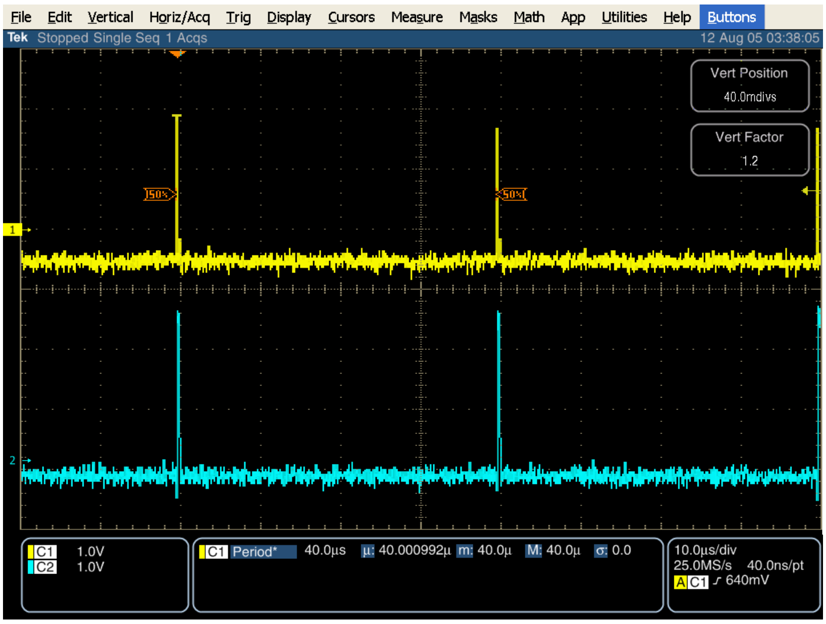
C1 1.0V
C2 1.0V

C1 Period* -s µ: ? m: ? M: ? σ: 0.0

5.0ns/div
10.0GS/s ET 100ps/pt
A C1 ƒ 800mV

Figure 3-11: Oscilloscope / Zoom on the signal transitions.

In Figure 3-12, it is shown a measurement of the packet transmission interval. The time resolution used is 10 µs/div. It is verified that a new packet is received every 40µs and using the instrument's measuring feature it is also verified that the deviation is negligible.


File Edit Vertical Horiz/Acq Trig Display Cursors Measure Masks Math App Utilities Help Buttons
Tek
50%
50%
1
2

Stopped Single Seq 1 Acqs 12 Aug 05 03:38:05

Vert Position 40.0mdivs

Vert Factor 1.2

C1 1.0V
C2 1.0V

C1 Period* 40.0µs µ: 40.000992µ m: 40.0µ M: 40.0µ σ: 0.0

10.0µs/div
25.0MS/s 40.0ns/pt
A C1 ƒ 640mV

Figure 3-12: Oscilloscope / Data available every 40µs.

3.5 SUMMARY

The formatting of the BLM packet has been proposed after an investigation on the data needed to be transmitted. Its 256 bits include not only the Counter and ADC data from each of the eight detectors but also information on the tunnel status, identification of the card and the packet, as well as redundant bits for error detection.

A gigabit optical link has been assembled by using the GOL, a custom made ASIC, in the GOH configuration, that can withstand the radiation levels expected in the LHC tunnel installations, together with a commercially available receiver. It will provide, through its high speed, the necessary low latency of the complete system.

To accommodate the receiving parts of four links a mezzanine card has been implemented and verified for correct operation. Additionally, it includes a flash memory with interconnections to both the front panel and the FPGA to hold card specific information like the Threshold and Masking Tables.

The advantage of keeping those design constraints and adding a mezzanine card is that the main card will maintain the universality to be used for more applications except in the BLM system. The time, effort and production costs in the whole group thus are significantly minimised. To provide a new functional system boils down to having only to produce the FPGA's firmware, that realise the system's processing needs, and design a mezzanine card, that accommodates the connection interface and/or any other specific need.

References

[38] CERN Microelectronics Group, *"Gigabit Optical Link (GOL) Project"*, [online: http://proj-gol.web.cern.ch/proj-gol/]

[39] C. Yen, R. Walker, P. Petruno, C. Stout, B. Lai and W. McFarland, *"G-Link: A chipset for Gigabit-Rate Data Communication,"* Hewlett-Packard Journal, Oct. 92.

[40] IEEE, *"IEEE 802.3z Gigabit Ethernet Standard"*, 1998.

[41] CERN ECAL Group, *"GOH Manufacturing Specifications, Version 3.30"*, 24 June 2003, [online: http://cms-ecal-optical-links.web.cern.ch/cms-ecal-optical-links/content/GOH_Specification.doc]

[42] P. Moreira, G. Cervelli, J. Christiansen, F. Faccio, A. Kluge, A. Marchioro, T. Toifl, J. P. Cachemiche and M. Menouni, *"A Radiation Tolerant Gigabit Serializer for LHC Data Transmission"*, Proceedings of the Seventh Workshop on Electronics for LHC Experiments, Stockholm, Sweden, 10-14 September 2001.

[43] Kukka Banzuzi and Donatella Ungaro, *"Optical Links in the CMS Experiment"*, Proceedings of SPIE, 2003, vol. 5125, [online: http://hep.fuw.edu.pl/cms/esr/docs/CMS-opto.pdf]

[44] M. Matveev, T. Nussbaum, P. Padley, J. Roberts, M. Trapathi. *"Optical Link Evaluation for the CSC Muon Trigger at CMS"*, Proceedings of the Seventh Workshop on Electronics for LHC Experiments, p. 379-382, Stockholm, Sweden, 10-14 September 2001.CERN 2001-005 CERN/LHCC/2001-034. 22 October 2001.

[45] CERN ECAL Group, *"GOH Radiation Testing"*, [online: http://cms-ecal-optical-links.web.cern.ch/cms-ecal-optical-links/content/GOH_irrad_Presentation_040316.ppt]

[46] Texas Instruments Inc, *"TLK1501, 0.6 to 1.5 Gbps Transceiver (Rev. F)"*, 05 Sep 2003, [online: http://www-s.ti.com/sc/ds/tlk1501.pdf]

[47] Neoptek Ltd, "NTPPT-3EP-x-x, 1.25Gbps Pigtailed Coaxial PIN-TIA PD (3.3V)", Data Sheet Ver.1.0, June 2005 [online: www.neoptek.com]

[48] STMicroelectronics, "M25P10-A: 1 Mbit, Low Voltage, Serial Flash Memory with 20 MHz SPI Bus Interface", February 2002, [online: http://www.st.com]

[49] IEEE Computer Society, P1386/Draft 2.4a, *"Draft Standard for a Common Mezzanine Card – CMC"*, 21 March 2001, [online: http://ess.web.cern.ch/ESS/standards/cmc-d24a.pdf]

[50] Motorola, inc. *"Serial Peripheral Interface (SPI) Block Guide"*, Jul 2004, [online: http://www.freescale.com/files/microcontrollers/doc/ref_manual/S12SPIV4.pdf]

[51] P. Moreira, *"The GOL Test Board Schematics"*, 07 Feb. 2002, [online: https://edms.cern.ch/document/337074/1]

04

Data Reception & Error Detection

Chapter 4. Data Reception & Error Detection

In communication systems, a significant role of the Data Link layer is to convert the potentially unreliable physical link between two systems into an apparently very reliable link. This is usually achieved by including redundant information in each transmitted packet. Depending always on the nature of the link and the data one can include just enough redundancy to make it possible to detect errors and then arrange for the retransmission of damaged packets.

In the case of the BLM system, this scenario is not possible for many reasons. The most obvious is that this configuration implies a bidirectional link, to acknowledge correct receipt. It will also need storage of the data, for retransmission, in a place with high levels of radiation. Thus, for those reasons the system was decided to avoid any kind of error checking at the transmitting part but include only enough redundancy for the receiving part to detect any discrepancies.

Error detection determines if the data received through a medium is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends it to the original data frame. The receiver uses the same calculation methodology, generates a checksum for the received data frame, and compares it against the transmitted checksum. If the two values are the same, the received data frame is correct and no corruption has occurred during transmission or storage. [52]

In this chapter, the reasons that forced this implementation as well as the reliability of the error detection used will be explored.

4.1 Reliability Increase of Link

In the BLMTC design in order to increase even more the reliability of the communication link it was decided to double it. The redundant signal will not be treated as an independent one but was decided to be driven to the same card so that a comparison of the two can be made and in the case of a particular difference to trigger a dump of the beam.

4.1.1 Redundant Signal Assessment

On the other side though, this redundancy is a potential source of errors that must be avoided. The most obvious are the "lock ups" or "dead locks", as they are usually called in the communication context, and the unsynchronised comparisons.

On a lock up, usually node A waits data from B to process but B waits for a data request. On an unsynchronised comparison, comparison of data taken from two different acquisitions is taking place. The probability of this to happen can increase the further behind the redundancy of the system is chosen to exist. That is, the doubling of the system can start somewhere just after the CFC (current-to-frequency converter) till the optical fibre's laser. The more processes included the more possibilities for a mistake of this kind.

In both cases, the only solution lies in the awareness of such problems and the construction of a solid firmware that will not allow those to happen.

4.1.2 Evaluation of Comparison Scenarios

There are many possible designs that could be used to treat the redundant signal when it arrives to the BLMTC system. In all of them at some point a comparison has to be made in order to discover the erroneous signal. The comparison can be made in various points of the system. An evaluation of possible designs is given:

- The system could be implemented with redundancy and compare at the output level, i.e. the *Th* (Threshold) & *W* (Warning) outputs. The implementation would be reasonably easy, as it would only need XOR logic at the outputs to recognize a difference. Although this would give the advantage that both processes would run in parallel avoiding problems like lock-ups and slower cycles it is still not a reliable implementation and would require twice the resources. Nevertheless, its worst weakness is that it will mask any differences below the *Th* & *W* levels making them accessible only by a post-mortem.

- Another idea is to perform comparisons of the complete incoming packets over fixed intervals instead of doing that for every reception of data. Again, even though that strategy gives the gain of speeding up the working cycle of the system it is still not a very reliable implementation, as it will impose the uncertainty of the non-checked data packets.
- To compare at the Sum-Registers level is a very attractive option only in the case where a specific implementation of the TC (Threshold Comparator) is chosen. In all other cases, the number of registers and the amount of data held in them makes this kind of comparison unattractive because of the computational power needed. (Note that, the implementation of the *TC* chosen will use approximately 200 registers with some of them holding data of up to 40-bit long values. That implementation is discussed in full detail in Chapter 6.)
- The cyclic redundancy check or CRC is a widely used parity bit based error detection scheme in serial data transmission applications. This code is based on polynomial arithmetic. Two blocks of data can be rapidly compared by seeing if their CRCs are equal, saving a great deal of calculation time in most cases. In the case of the *BLMTC* it is also foreseen to be used for error detection so it will be already appended to the transmitted packets. A block diagram of this procedure is shown in Figure 4-1 and more about the CRC algorithm to be used is explained in Section 4.3.

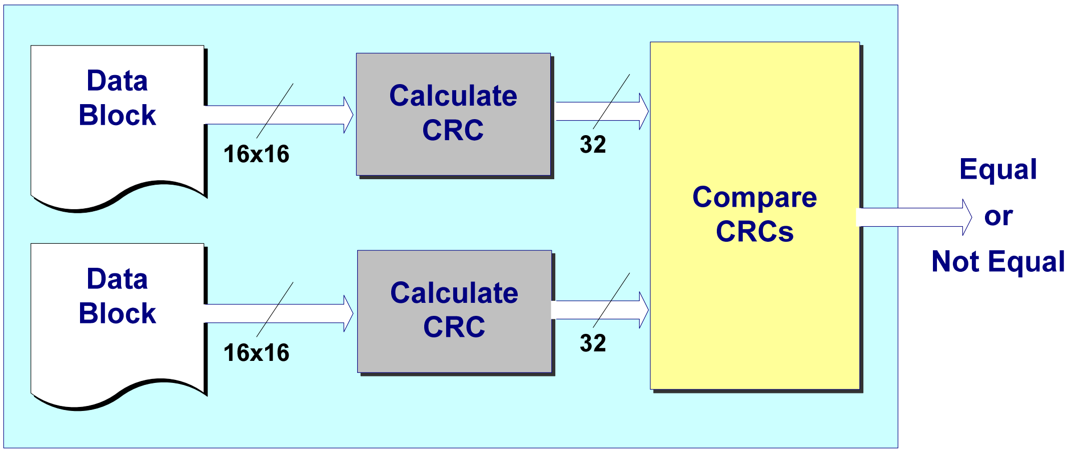


Figure 4-1: Data Block Comparison Using the CRC Algorithm.

4.2 THE RCC (RECEIVE, CHECK AND COMPARE) PROCESS

The BLMTC system receives, de-serialises and decodes the packets transmitted over the doubled optical link. The system should be able to extract one valid packet of detector data to provide on the later stages of the processing. This assignment was given to the RCC process, which performs the following functions.

The RCC will check for errors on both transmissions, i.e. calculate their CRCs at every packet reception. It will compare the CRC's of the two packets (4bytes each) in order to check if the two packets are identical and coming from the same source. By using all available information it will be able to select a valid signal to use (either the primary or the redundant) or request the dump of the circulating beam. The RCC process is one of the few processes that have the ability to trigger a dump of the beam. For that reason, a great effort will be given to make it as reliable and error free as possible.

Its additional tasks will be to check the status of the tunnel installation by checking the status information received at each packet. It will collect and report the errors seen. Actually, it will provide a significant part of the Status and Error Logging information by collecting the errors seen from the various checks that will be used by the expert applications to track potential problems.

Finally, by removing the extra/redundant bits, i.e. leaving only the counter and the ADC data and realigning these data will provide the 20bit data from each of the eight detectors to their later parallel processing stages.

A block diagram of those proposed functions is illustrated in Figure 4-2 and on the following sections of this chapter a more in depth explanation will be done for them.

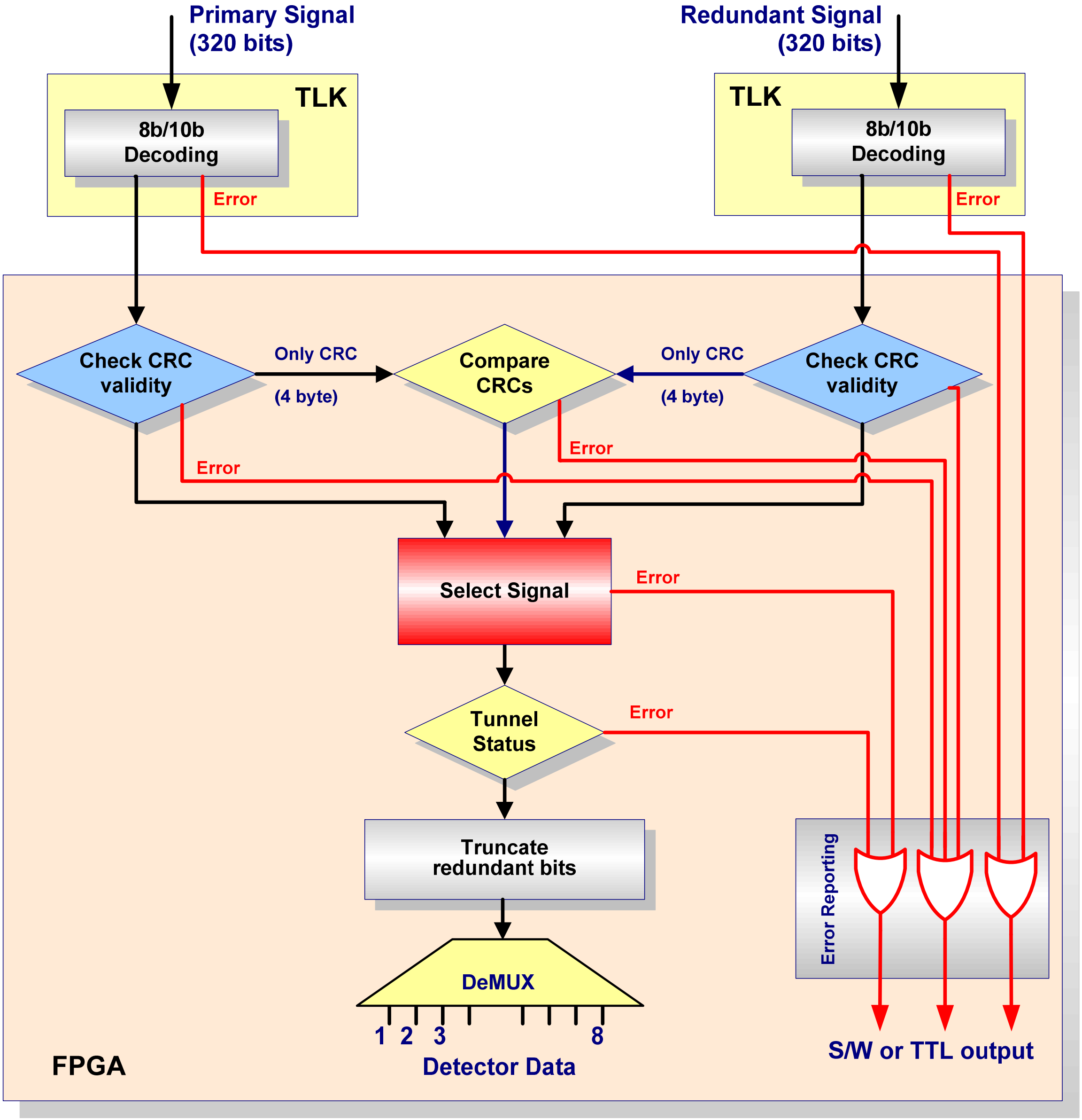


Figure 4-2: Block Diagram of the RCC (Receive, Check and Compare) Process.

4.3 CRC CODING

Cyclic Redundancy Check (CRC) is an error-checking code that is widely used in data communication systems and other serial data transmission systems. CRC is based on polynomial manipulations using modulo-2 arithmetic. Some of the common Cyclic Redundancy Check standards are the CRC-8, CRC-12, CRC-16, CRC-32, and CRC-CCIT.

The CRC function [53] validates data streams via redundant encoding. CRCs are a preferred type of redundant encoding, where redundant bits are spread over more bits than the original data stream. Similar to parity checking, CRC encoding is a method of generating a code to verify the integrity of the data stream. However, while parity checking uses one bit to indicate even or odd parity, CRC encoding uses multiple bits, and therefore catches more errors in the data stream. CRCs are particularly effective for two reasons:

- CRCs provide excellent protection against common errors such as burst errors, in which consecutive bits in a data stream are corrupted during transmission.
- The original data are the first part of the transmission, which makes systems that use CRCs easy to understand and implement.

4.3.1 CRC CALCULATION

In general, an n-bit CRC is calculated by representing the data stream as a polynomial *M(x)*, multiplying *M(x)* by x^n (where n is the degree of the polynomial *G(x)*), and dividing the result by a generator polynomial *G(x)*. The resulting remainder is appended to the polynomial *M(x)* and transmitted. The complete transmitted polynomial is then divided by the same generator polynomial at the receiver end. If the result of this division has no remainder, there are no transmission errors.[57]

Mathematically, this can be represented as: $CRC = remainder\left[M(x) * \frac{x^n}{G(x)}\right]$ [4-1]

In general *G(x)* should have a degree greater than zero and less than that of the polynomial *M(x)*. Another requirement for *G(x)* is a non-zero coefficient in the x^0 term.

The CRC-32 uses the generating polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x \ +1 \qquad [4\text{-}2]$$

The polynomial for other commonly used CRC codes can be seen in Table 4-1.

4.3.2 PROBABILITY OF CRC ERRORS NON-DETECTION

The method, though, is non-infallible. However, if the *N*-bit checksum is large, the probability that some errors will not be detected can be made very small.

More specifically, the errors that can be detected by the CRC are:

1) all single-bit errors

2) all double-bit errors (as long as G(x) has 3 terms).

3) any odd number of error as long as G(x) has(x + 1) as a factor.

4) any burst error with a length less than the length of CRC,

5) most longer bursts, e.g. for G length of N+1 bits and burst length of B the probability of undetected error Pr is equal to

$$\mathrm{Pr} = \begin{cases} 2^{1-N}, & \text{for } B = N+1 \\ 2^{-N}, & \text{for } B > N+1 \end{cases} \qquad [4\text{-}3]$$

Solving Equation 4-3 for

B>17 and N=16 gives $\mathrm{Pr} = 2^{-16} = 1.5259 * 10^{-5}$,

B>33 and N=32 gives $\mathrm{Pr} = 2^{-32} = 2.3283 * 10^{-10}$,

Where *Pr* is the probability of not detecting the error, *N* is the bits of the checksum, and *B* is the length of the burst error.

Thus, **CRC-16:** ***Pr = 1.5259*10*** $^{-5}$ and

CRC-32: ***Pr = 2.3283*10*** $^{-10}$

for burst errors longer than **17** and **33** bits respectively.

In the BLM system, as already have been said, the intention is to transmit the message twice through different channels. If, and only if, both messages pass the CRC check and contain identical information the message is considered valid. In all other cases, either software or hardware triggers will be initiated. Thus, the probability of not detecting the error becomes even lower. In support to the above argument, is the use of optical fibre systems, which have the property of not producing burst errors.

4.3.3 CRC Parallel Implementation

A parallel implementation operates on multiple bits of the data stream per clock cycle. An algorithm for generating the next-state equations for parallel implementation of CRC-32 is given in *"A Symbol Based Algorithm for Hardware Implementation of Cyclic Redundancy Check (CRC)"* [58] and a block diagram of its implementation is illustrated in Figure 4-3 [57]. The algorithm involves looping to simulate the shifting, and concatenating strings to build the equations after "n" shifts.

Table 4-1: Parameters for Various Standard CRC Algorithms.

NAME	WIDTH	POLY	INIT	REFIN	REFOUT	XOROUT	CHECK
		(Hexadecimal)				(Hexadecimal)	
CRC-16	16	8005	0000	"YES"	"YES"	0000	BB3D
CITT	16	1021	FFFF	"NO"	"NO"	0000	29B1
Kermit	16	8408	0000	"YES"	"YES"	0000	0C73
CRC-32	32	04C11DB7	FFFFFFFF	"YES"	"YES"	FFFFFFFF	CBF43926
JamCRC	32	04C11DB7	FFFFFFFF	"YES"	"YES"	00000000	340BC6D9
ZMODEM	16	1021	0000	"NO"	"NO"	0000	31C3

Notes:

The CHECK is a simple way to verify that the algorithm is working properly. The CHECK word is the CRC output value when the ASCII string "123456789" (equivalent to the decimal string "49 50 51 52 53 54 55 56 57") is input to the CRC algorithm.

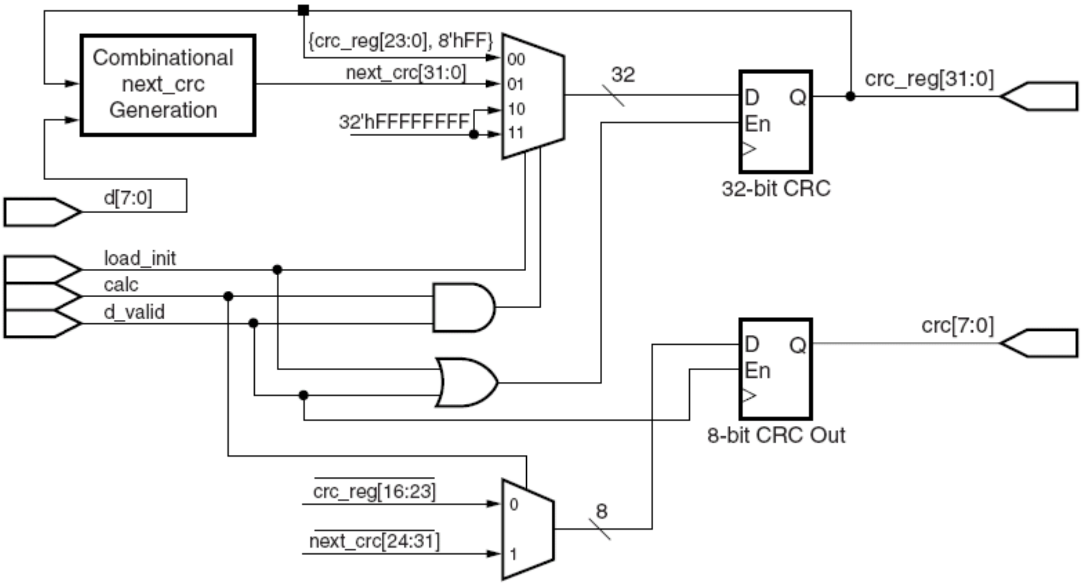


Figure 4-3: Block Diagram of CRC-32 Parallel Implementation. [57]

4.3.4 CRC VHDL IMPLEMENTATION

The CRC production and check blocks have been implemented having a 16-bit parallel calculation in order to minimize the latency imposed by the consecutive cycles needed for the system to produce or check the checksum. A higher bit parallel implementation would be possible, of course with the expense of more FPGA resources, but this system would not gain more in speed. The data bus provided by the physical link, shown in the preceding chapter, limits the input to a maximum of 16-bit.

The packet will be consisting of 256 bits in total, thus 14 cycles are needed to calculate, and 2 to compare the CRC product with the one transmitted. Having in mind that the same procedure is done twice once before being transmitted and once after being received this gives a minimum of 32 cycles of latency added by this process. Both of the systems use 40MHz system clocks adding up to a total of 800ns of system latency without adding any transmission delay.

The result of implementing the CRC block (written in VHDL and imported to Quartus II software) can be seen at Figure 4-4. The code was written in non-vendor specific VHDL so that it can be reused easily by the CFC card's FPGA. In that way, the two systems compatibility would be assured. The actual VHDL code can be seen in the appendix's Section A.1.

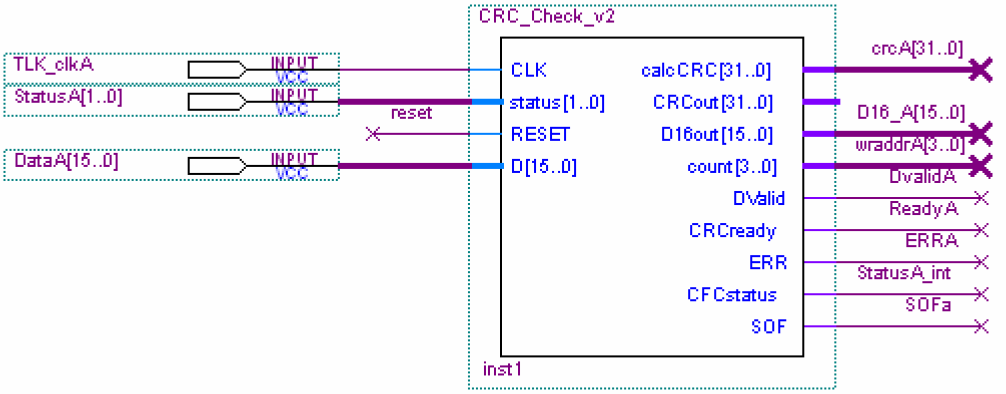


Figure 4-4: Block Symbol of CRC-32 Parallel Implementation created with Quartus II.

The block receives 16-bit input of data *(input: D[15..0])* in parallel at every raising edge of the clock *(input: CLK)*. There is also an asynchronous reset pin *(input: RESET)* and two flags for the incoming data status *(input: status[1..0])* which are given by the communication link.

The outputs of the block comprise of the received data, the received CRC (both registered) and the calculated CRC code *(outputs: D16out[15..0], CRCout[31..0]* and *calcCRC[31..0] respectively)* with all these data outputs giving in parallel either 16 or 32-bits. There are two control signals for the later stages, one signifies when the calculated value is available *(output: CRCready)* and the other the address to write the received data *(output: count[3..0])*. Finally, there are two error flags, one signalling an error at the tunnel *(output: CFCstatus)* and the other for indicating an error in the check *(output: ERR)*.

4.4 8B/10B CODING

The 8B/10B coding was introduced in 1982 by A.X.Widmer and P.A.Franaszek [59] from IBM. This way of coding is used today in transmission protocols – synchronous, like Fibre Channel, and asynchronous, like Gigabit Ethernet [40]. This codec is also used for the 800Mbps extensions to the IEEE 1394 / Firewire standard [60], and 8B/10B is the basis of coding for the electrical signals of the PCI Express standard [61].

Nowadays, there are dedicated chips that could be used or even IP cores if it is needed to be implemented in the FPGA that can also handle the synchronization and the (de)serialization. Altera provides, as well as other companies do, an 8B/10B IP Core [63] and in the Stratix GX family [64] [65] of devices exists as an embedded hardware option. In the BLM system case, the GOL chip is being used as a transmitter and the TLK as a receiver, and both of them are making use of this coding.

With the 8B/10B encoding before transmission, the complete frame is split into 8-bit blocks and each of them is encoded into a 10-bit block. On receipt, the reverse procedure (i.e. decoding) is used. It serves two purposes. First, it makes sure there are enough transitions in the serial data stream so the phase of the transmitted clock can be recovered easily from the embedded data. Second, because it thrives in transmitting the same number of ones as zeros, it maintains a DC balance in the transmission line.

The 8B/10B code is an example of the more general mBnB code, in which m binary source bits are mapped into n binary bits for transmission. Redundancy is built into the code to provide the desired transmission features by making n > m. The 8B/10B code actually combines two other codes, a 5B/6B code and a 3B/4B code. A mapping is defined that maps each of the possible 8-bit source blocks into a 10-bit code block.

There is also a function called disparity control. In essence, this function keeps track of the excess of zeros over ones or ones over zeros. An excess in either direction is referred to as a disparity. If there is a disparity, and if the current code block would add to that disparity, then the disparity control block complements the 10-bit code block. This complement has the effect of either eliminating the disparity or at least moving it in the opposite direction of the current disparity.

4.4.1 ENCODING ALGORITHM

The 8B/10B coding relies on conversion of 8 input bits to 10 output bits sent via any transmission channel. In order to improve the coding algorithm, the input bits are divided to two separate sub-groups: 5-bit and 3-bit. The bits and groups are signed as ABCDE and FGH. Each of the groups is coded separately with the algorithm 5B/6B and 3B/4B. These functions convert the input bits to *abcde* and *fgh* and add one additional bit *i* and *j* to each group. These groups are joined together to form a single 10-bit output word.

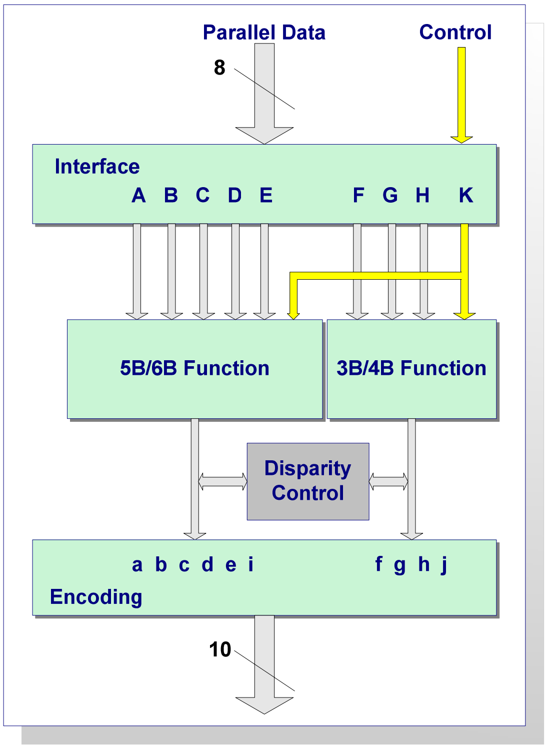


Figure 4-5: Block Diagram of the 8B/10B Encoding Process.

The way of coding depends on the value of binary disparity signal. The disparity signal is zero or unity depending on the number of 0s and 1s in the previously sent 10-bit output word. For example, if the number of 0s is bigger than 1s, the value of disparity is changed to the opposite.

The 8B/10B coding assures the separation of data bytes from control bytes. When the control bytes are transmitted, the state of the control signal at the input of the coder is high. The method of coding is changed in this case. The coder is able to distinguish the kind of received byte.

4.4.2 DECODING ALGORITHM

The decoder converts 10 bits back to 8 bits of the input data. Similarly as in the coder, here the bits re divided to two parts: 6-bit and 4-bit. Next, the decoding processes restore 5-bit and 3-bit of original parts. They are combined to the output 8-bit word. The decoder recognizes the type of control byte by investigation of particular bits of the received word. A control word is shown that has been received by the setting of the control line to the high state.

4.4.3 ERROR DETECTION

Error discovery in the sent words is done on the level of 6-bit and 4-bit parts as well as on the level of the whole word. The checking embraces the investigation of conformance with allowed sequences of coding. When an erroneous word is received, which fulfils the condition of the control word, but could not have been sent from the coder, the transmission error signal is also reported. The discovery of error in the disparity signal is done independently for the 6-bit group and 4-bit group. The groups are recovered and compared to the expected values. When the error is discovered in the disparity signal in a group, a general error is reported.

The first aspect to be considered is the exploitation of the code redundancy for error checking. The second area to be explored is how errors in the line code interact and affect error detection when the CRC encoding, shown in Section 4.3, is applied beforehand to the digits.

4.4.4 ERROR CHECKING USING REDUNDANCY OF 8B/10B CODE

When investigating the validity of entire packets and not the correctness of individual characters it is seen that each packet starts and ends with a delimiter. For packets defined in this way, each of the start and end characters contain at least one nonzero-disparity subblock, which prevents disparity violations arising from errors to be carried forward into another packet.

In addition, the developers of this code have claimed and later proved that, with the receiver in character synchronisation, no single digit error can generate the delimiter (that is, the 0011111000 or its complement) from encoded data. As well as that, the opposite is also true and no single error can transform the delimiter into a data code point.

The simplest error patterns, which may escape detection by the code, are a single erroneous 1 complemented by a single erroneous 0. Such complementary errors, when confined to a single subblock, may simply change it into another valid code point. Single errors in subblocks change the disparity. Thus, it is possible that a complementary pair of digit errors can change the disparity of two subblocks in conformance with the alternating polarity rule, such that the errors are not detectable by the code. [59]

4.5 Combined Error Detection of CRC and 8B/10B code

Single errors (or short error bursts) in the encoded line digits of a block code can generate a longer error burst in the decoded message. For the 8B/10B code proposed here, the effects of line digit errors are always confined to the 6B or 4B subblocks in which they occur and generate error bursts no longer than 5 or 3 respectively (from a single line digit error). [59]

This derives from the fact that each 6B or 4B subblock is uniquely decodable on the basis of just the digit values belonging to that subblock and without any reference to disparity or other extraneous parameters. The only exceptions are the special characters K.28.1, K.28.2, K28.5, and K.28.6 (their purpose is shown in Table 3-3), for which the decoding of the *fghj* bits is dependent on the *abcdei* bits. However, adverse effects from this are limited because special characters usually appear only at specified slots with respect to the packet boundaries and are not covered by the CRC**.** [59]

From the preceding paragraph one can conclude that, the CRC used with this code can detect at least any combination of double errors in the line digits, thus making a significant impact on the combined guaranteed level of error detection. A double error in the line digits generates, in the worst case, two error bursts of 5 each, after decoding. For this case, specific generator polynomial for the CRC codes have been shown that are used with the capability of detecting two error bursts. In general, with 16 check bits, two bursts of combined length 10 or less can be detected in packets as long as 142 bytes; 24 check bits can accomplish the same thing for packets as long as 36 862 bytes.

Finally, it has already been shown in Section 4.3.2 that a CRC can generally detect any single error burst of a length which does not exceed the number of check bits. With the 8B/10B code described here, any single error burst of length 15 or less in the encoded digits cannot grow to more than 16 bits after decoding. Similarly, error bursts of length 25 and 35 in the encoded bits translate into error bursts no longer than 24 and 32 bits, respectively, after decoding. Thus, the 32 check bits that will be used in the BLM system together with the 8B/10B coding will detect any errors in its 32-byte long packets.

4.6 Selection of Signal

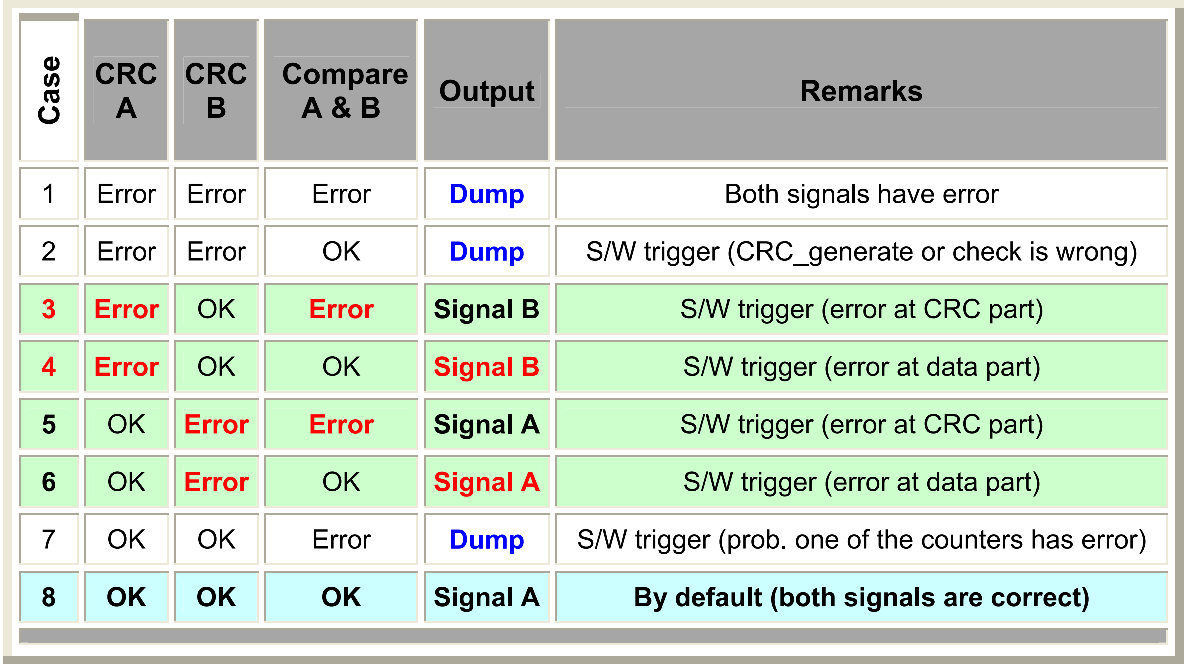
The *Signal Select* function receives the outputs of the CRC checks and their comparison and gets the responsibility of selecting the error free signal to convey on the stages below. In Table 4-2 can be seen the output for any of the possible inputs.

Additionally, its outputs will be collected and used by error reporting processes to indicate problematic areas and failing components in the tunnel installation. Some of the problems it can discover are indicated on the remarks of Table 4-2.

4.6.1 System Availability Enhancement

This function was foreseen for two reasons. The first is to discard one of the two signals if they are identical and the second is in order to have a valid signal in the case where one of the transmissions failed but the other went through with no errors. Thus, by using efficiently all information it is expected to increase the availability of the system.

Table 4-2: Outputs of the Valid Signal Selection Function.



Case	CRC A	CRC B	Compare A & B	Output	Remarks
1	Error	Error	Error	**Dump**	Both signals have error
2	Error	Error	OK	**Dump**	S/W trigger (CRC_generate or check is wrong)
3	**Error**	OK	**Error**	**Signal B**	S/W trigger (error at CRC part)
4	**Error**	OK	OK	**Signal B**	S/W trigger (error at data part)
5	OK	**Error**	**Error**	**Signal A**	S/W trigger (error at CRC part)
6	OK	**Error**	OK	**Signal A**	S/W trigger (error at data part)
7	OK	OK	Error	**Dump**	S/W trigger (prob. one of the counters has error)
8	**OK**	**OK**	**OK**	**Signal A**	**By default (both signals are correct)**

The errors seen in cases 3 to 6 (see Table 4-2) will be allowed to occur without the system giving an immediate signal to dump the circulating beam. Instead, the system will continue uninterrupted using, for the processing of the data, the signal that had no CRC error.

Moreover, a software trigger will be sent to the error logging that can use this to detect and program replacement of the failing parts immediately or in the next scheduled maintenance depending on their repetition.

4.6.2 IMPLEMENTATION OF THE SIGNAL SELECT FUNCTION

The function's truth table can be constructed, using the information for its expected behaviour, as seen in Table 4-3.

Table 4-3: Truth Table of Signal Select Function.

Inputs			Outputs	
ERR A	ERR B	ERR C	Valid Signal	Dump
0	0	0	0	0
0	0	1	d	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	d	1
1	1	1	d	1
Valid Signal: 0 = Signal A, 1 = Signal B, d = don't care Dump: 0 = Allow, 1 = Dump				

Extracting now the equations from the truth table:

$$Valid\ Signal = \overline{A}\,B\,\overline{C} + \overline{A}\,B\,C = \overline{A}\,B \qquad [4\text{-}4]$$

$$Dump = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + A\,B\,\overline{C} = \overline{A}\,\overline{B} + A\,B\,\overline{C} \qquad [4\text{-}5]$$

A way of implementing those equations using logic gates is illustrated in Figure 4-6.

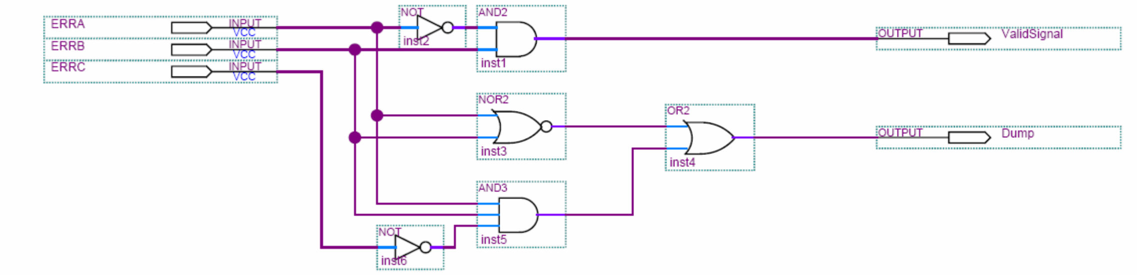


Figure 4-6: Implementation of the Signal Select Function using Logic Gates.

4.7 METASTABILITY FROM CLOCK DOMAIN CROSSING

The binary data from each link arrive with their own clock recovered from the CDR (clock data recovery) circuitry and must be crossed to the clock domain of the system. Any asynchronous input from the outside world to a clocked circuit represents a source of unreliability, since there is always some probability that the clocked circuit will sample the asynchronous signal just at the time that it is changing. A signal that crosses clock domains is also considered asynchronous into the new clock domain since no constant phase and time relationship exists between the two domains.

The proper operation of a clocked flip-flop depends on the input being stable for a certain period of time before and after the clock edge. If the setup and hold-time requirements are met, the correct output will appear at a valid output level (either Low or High) at the flip-flop output after a maximum delay of tCO (the clock-to-output delay). However, if these setup and hold-time requirements are not met, the output of the flip-flop may take much longer than tCO to reach a valid logic level. This unstable behaviour is called metastability.[68]

In the digital design practises, usually a multistage synchronising circuit [69] [70] is used to improve the MTBF (Mean-Time-Between-Failure) [68] of the design. Here, a different approach to this very common problem has been taken. The data need to be preserved for some time that is enough to calculate their CRCs and decide which signal to convey in the later stages. A true-dual-clocked RAM is used for each of them that combine both features. It buffers the data for the necessary time by writing the data into the memory using the recovered clock and when the process decides which of them should be used the memory is read using the system clock.

Finally, the design employs different input to output widths of the dual-clock RAMs to minimise the delay that will be added from reading back the correct data with the expense of increased memory block usage.

4.8 IMPLEMENTATION OF THE RCC PROCESS

The RCC process, shown in Figure 4-7, is situated in the input stage of the FPGA. It receives the data from the two links (inputs: *DataA* and *DataB*) via the PMC Connectors and the Mezzanine, shown in Sections 2.4.1 and 2.4.2 respectively, together with the link control pins (inputs: *StatusA* and *StatusB*) and their recovered clock signals (inputs: *TLK_clkA* and *TLK_clkB*).

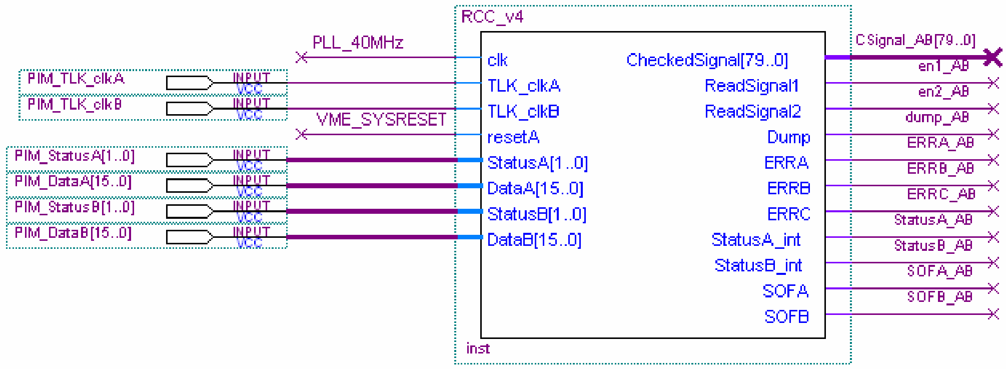


Figure 4-7: RCC Block Symbol created with Quartus II.

After processing those inputs, only the valid detector data (160bits) are outputted (output: *CheckedSignal[79..0]*) in two consecutive clock cycles controlled by their respective output enable signals (outputs: *ReadSignal1* and *Read Signal2*).

Finally, a plethora of other information is also outputted that is intended for the Error and Status Reporting (ESR) process. This process is discussed in more detail on Chapter 8. Those outputs include the errors seen from the CRC checks (outputs: *ERRA* and *ERRB*), the CRC comparisons (output: *ERRC*), and the new packet received flags (outputs: *SOFA* and *SOFB*).

The schematic implementation of the RCC is illustrated in Figure 4-8.

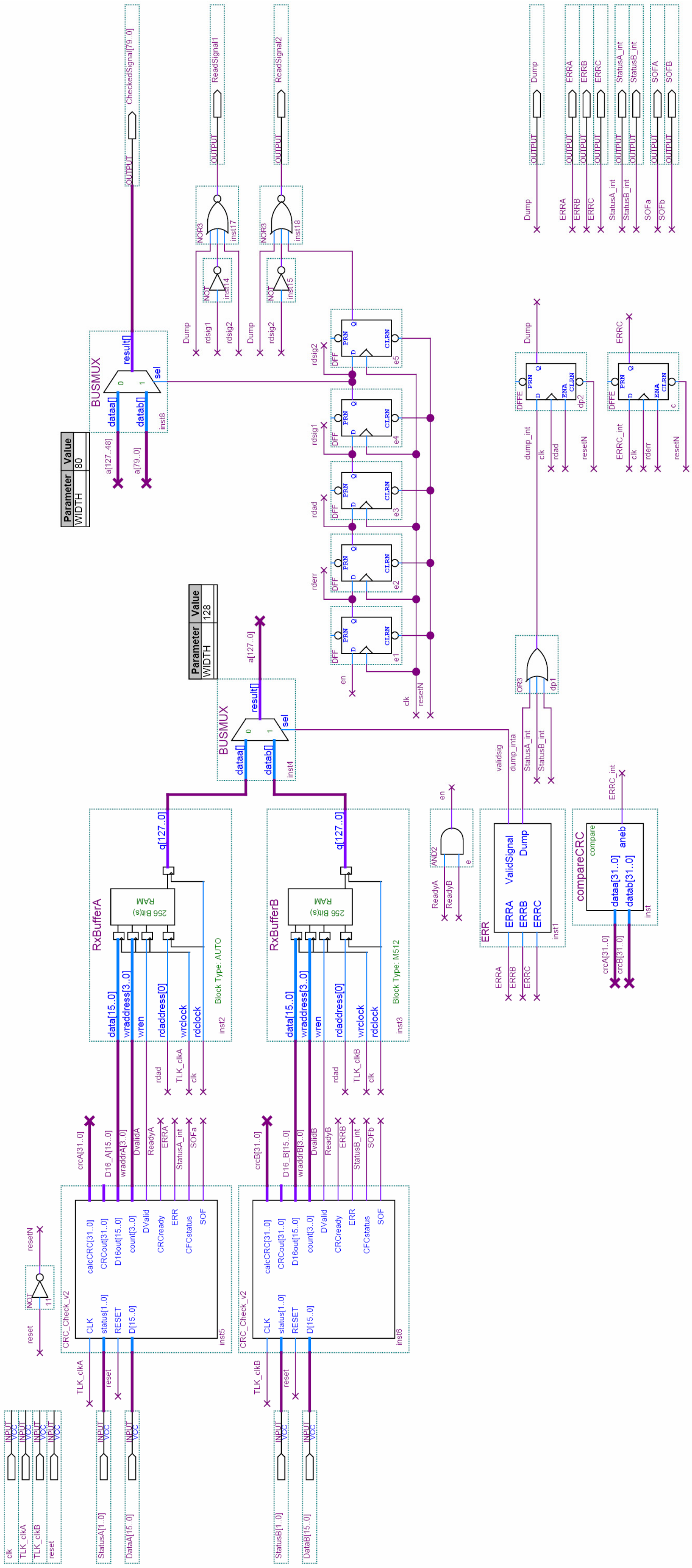
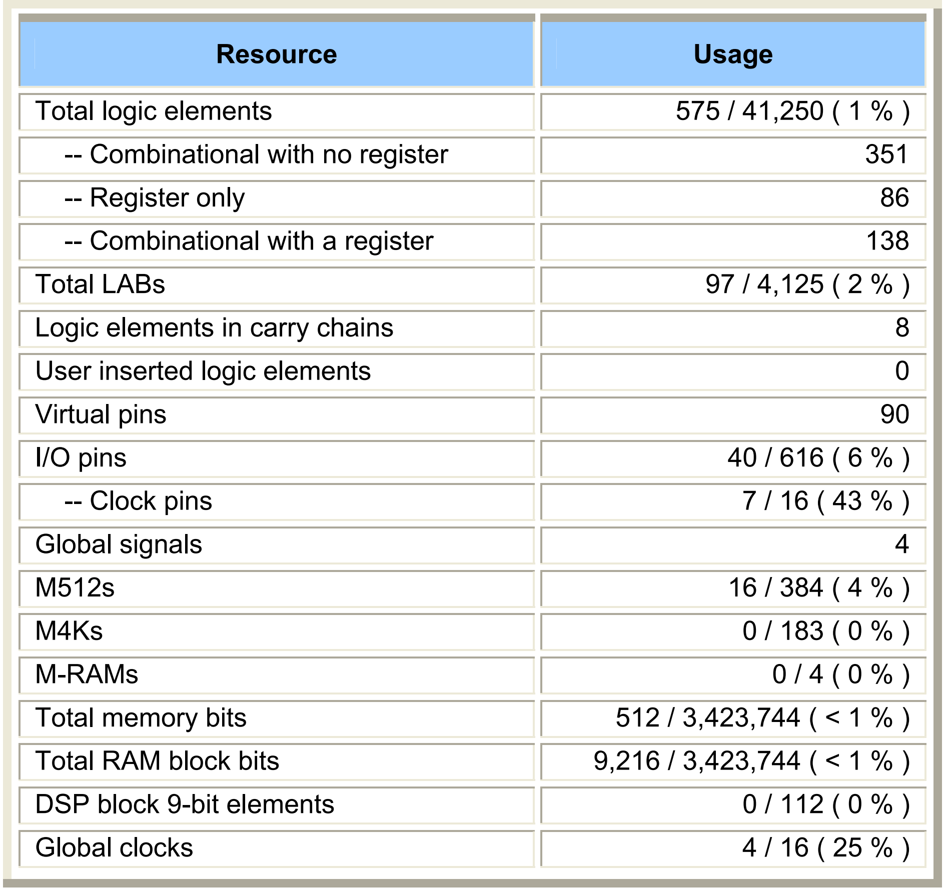


Figure 4-8: RCC Schematic Implementation with Quartus II.

4.8.1 RESOURCE UTILISATION BY THE RCC PROCESS

Table 4-4 summarises the resource usage of the proposed RCC process by the Quartus II Fitter. It can be derived that eight of the embedded M512 memory blocks [71] are necessary to implement each of the *RxBuffer*s instead of one (if implemented in a different configuration) in order to achieve the minimum latency. If found necessary, it is possible this usage to be exchanged, either with four of the M4K memory [71] blocks, or with direct implementation in Logic Elements.

Table 4-4: RCC Fitter Resource Utilisation Summary.



Resource	Usage
Total logic elements	575 / 41,250 (1 %)
-- Combinational with no register	351
-- Register only	86
-- Combinational with a register	138
Total LABs	97 / 4,125 (2 %)
Logic elements in carry chains	8
User inserted logic elements	0
Virtual pins	90
I/O pins	40 / 616 (6 %)
-- Clock pins	7 / 16 (43 %)
Global signals	4
M512s	16 / 384 (4 %)
M4Ks	0 / 183 (0 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	512 / 3,423,744 (< 1 %)
Total RAM block bits	9,216 / 3,423,744 (< 1 %)
DSP block 9-bit elements	0 / 112 (0 %)
Global clocks	4 / 16 (25 %)

4.9 STATIC TIMING SIMULATION OF THE RCC PROCESS

The first step for the verification of the correct implementation of the RCC process is a time-driven back-annotated simulation using the Quartus II Simulator. More information about this verification tool and the setting used can be found in Section 2.8.

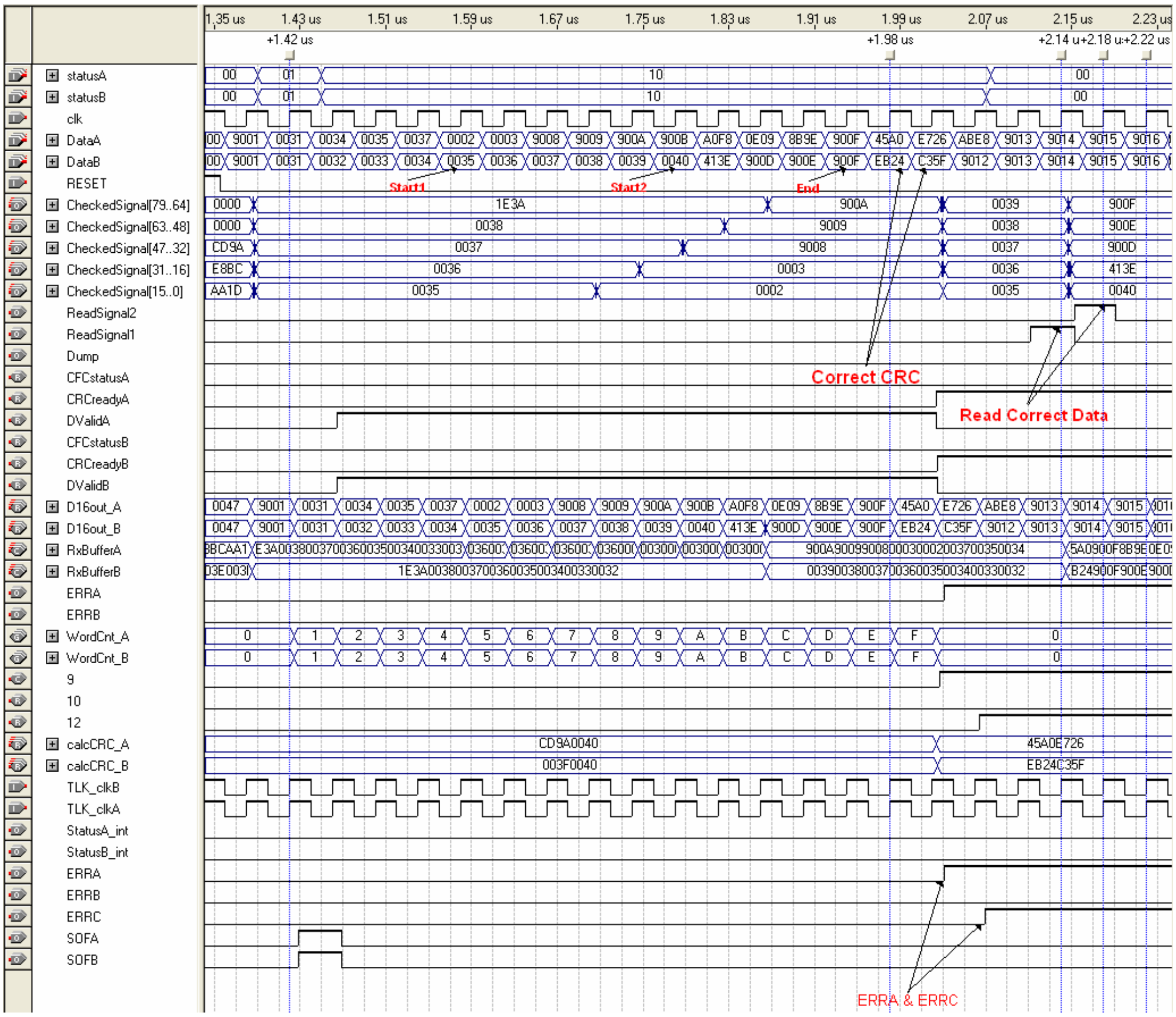


Figure 4-9: RCC Simulation / Only One Signal Correct.

In Figure 4-9 is shown a part of the simulation output waveforms where only one signal had correct data, i.e. data that corresponded to the CRC it carried. The ERRA and ERRC flags were asserted but not the Dump and the correct frames were available to be read when the output read enable signals ReadSignal1 and ReadSignal2 were asserted.

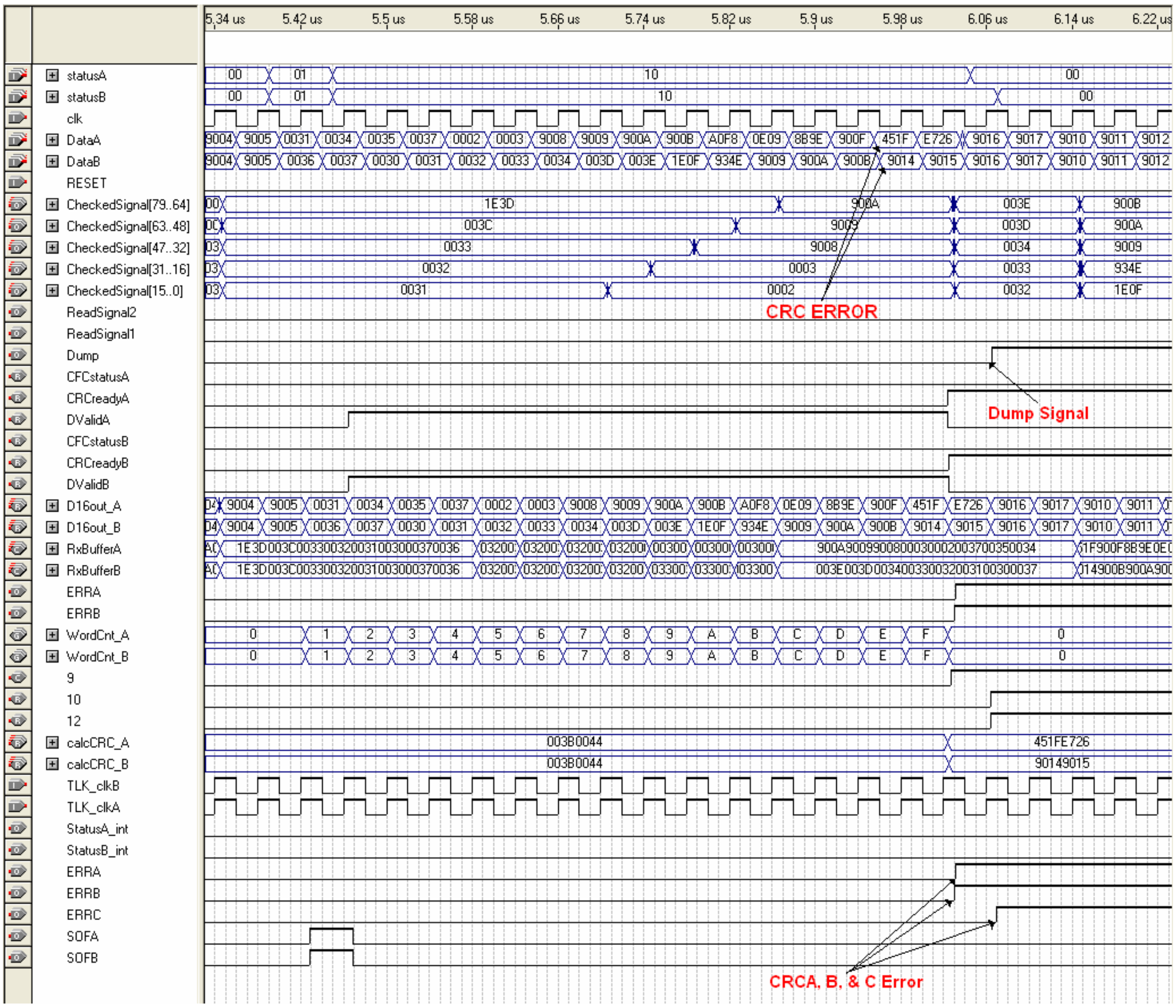


Figure 4-10: RCC Simulation / Both CRCs have Error.

In Figure 4-10 another part of the simulation output waveforms is shown where both signals had errors, i.e. data that did not correspond to the CRC they carried. Here all the error flags were asserted, i.e. ERRA, ERRB, ERRC together with the Dump. Correctly, in this situation the output read enable signals were not asserted.

4.10 HARDWARE VERIFICATION TEST OF THE RCC PROCESS

One more great tool provided for the verification in the Quartus II environment is the SignalTap II embedded logic analyser. This tool and the general settings used is shown in more detail in Section 2.8.

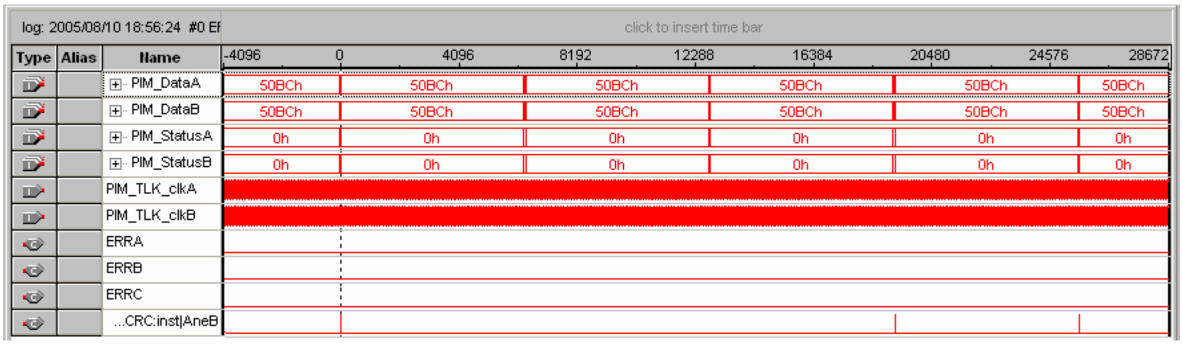


Figure 4-11: SignalTap II / Maximum Length of Acquisition.

It uses the spare FPGA internal memory to store the acquired data. Thus, the acquisition clock and the number of signals asked to observe are reverse proportional to the length of the observation. In Figure 4-11 is shown a complete acquisition by the SignalTap with a clock of 80MHz. The signal nodes chosen allow the acquisition to reach to 32KSamples meaning that this logic analyser is using all four M4K memory blocks [71] available in the EP1S30 FPGA device.

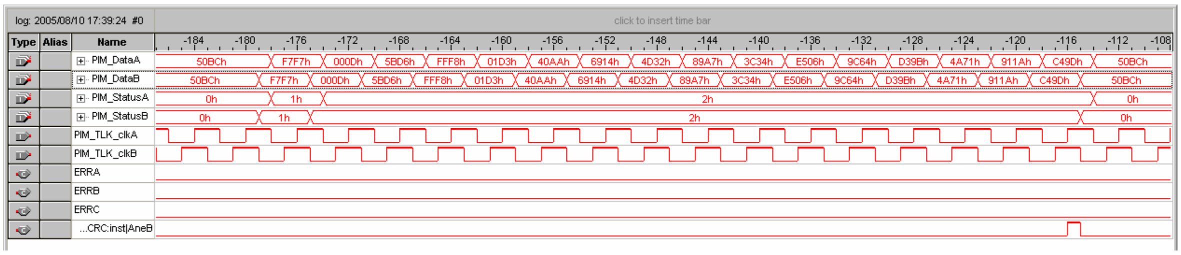


Figure 4-12: SignalTap II / Non-Synchronised TLK clocks.

In Figure 4-12 is shown a case similar to the one measured using an oscilloscope, seen in Section 2.4.2, during the verification of the BLM Mezzanine card. Since the two recovered clock signals from the optical link are not synchronised, the packets can also arrive having a time difference. Here, it is also seen that correctly the comparison of their CRCs does not give an error assertion of the ERRC output because of this.

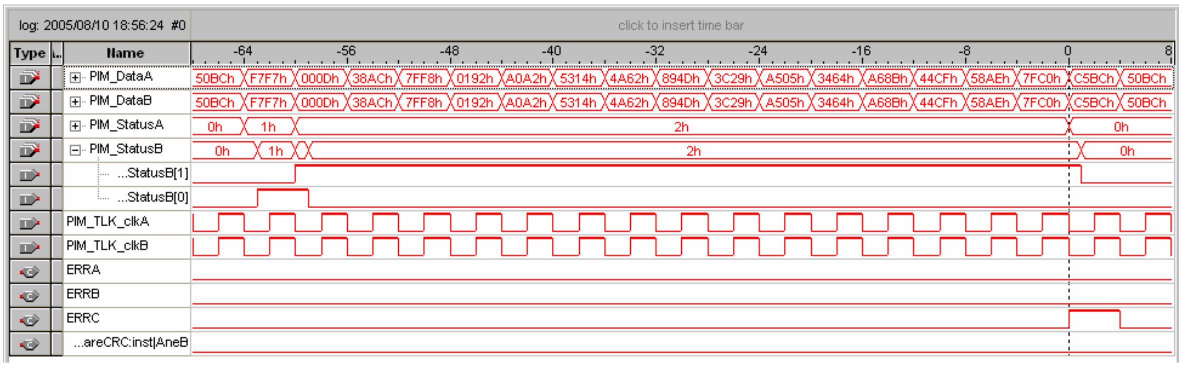

log: 2005/08/10 18:56:24 #0
click to insert time bar
Type
Name
-64
-56
-48
-40
-32
-24
-16
-8
0
8
PIM_DataA
PIM_DataB
PIM_StatusA
PIM_StatusB
...StatusB[1]
...StatusB[0]
PIM_TLK_clkA
PIM_TLK_clkB
ERRA
ERRB
ERRC
...areCRC:inst|AneB
50BCh F7F7h 000Dh 38ACh 7FF8h 0192h A0A2h 5314h 4A62h 894Dh 3C29h A505h 3464h A68Bh 44CFh 58AEh 7FC0h C5BCh 50BCh
50BCh F7F7h 000Dh 38ACh 7FF8h 0192h A0A2h 5314h 4A62h 894Dh 3C29h A505h 3464h A68Bh 44CFh 58AEh 7FC0h C5BCh 50BCh
0h 1h 2h 0h
0h 1h 2h 0h

Figure 4-13: SignalTap II / SOF Problem.

Figure 4-13 is showing the opposite result for the same situation of the non-synchronised data that recorded during the development of the RCC process and before achieving the required functionality. Even though the data from both channels are identical because of the clocks difference in phase the ERRC flag was asserted. That situation was easily solved by introducing a clock delay to the comparison of the two signals.

Finally, during the hardware test of this process, many other situations that the system might face were created on purpose to cover likely possibilities. Some of them were sudden disconnection of one of the signals, errors in the packets with identical CRCs, stop of the packet transmission for periods longer that 40μs, and freezing of the CFC card. In all of those cases, the process exhibited the wanted operation.

4.11 SUMMARY

In the BLM system, for reliability reasons, many parts were decided to have redundancy. The communication link was one of them. A comparison of the two could trigger a dump of the beam in the case of a difference. By including, some digital techniques, like the 8B/10B coding and the CRC (Cyclic Redundancy Check), a more sophisticated receiver part was produced that achieves even higher data reliability and system availability.

The CRC, a polynomial arithmetic based algorithm, is widely used as error detection scheme. In this system, the redundant bits it introduces are not only used to detect transmission errors but also to find differences between the two signals.

Some of the advantages the 8B/10B code gave to the system were: the simplicity and the low cost needed for it to be added to the system, DC balancing, with minimal deviation from the occurrence of an equal number of 1 and 0 bits across any sequence, good transition density for easier clock recovery and the additional error detection capability.

Especially for the error detection functions for the transmission and because of their importance in the reliability of the system, a more thorough study has been conducted. The first aspect to be considered was the exploitation of the code redundancy for error checking and the probability of errors passing undetected. The second area to be explored was how errors in the 8B/10B code interact and affect the error detection by the CRC. Both have given very promising results and a very low probability of non-detection.

The design incorporated an efficient method to cross the received data through the different clock domains avoiding any possible metastability. It provides the correct data to the next stage with minimum clock cycles, i.e. minimum latency, and compensates possible delays from the clock recovery circuitry. Moreover, very important information will be passed through its error-reporting feature of the status and the future failures of the whole system.

Finally, this part of the design has exhibited the intended functionality in both simulation and in-system hardware tests. Actually, many iterations in the design cycle have been initiated by the hardware tests which is usually the case when off-chip connections need to be made.

References

[52] Altera Corp. *"Error Detection Using CRC in Altera FPGA Devices"*, [online: http://www.altera.com/literature/an/an357.pdf]

[53] Altera Corp. *"CRC MegaCore Function, Parameterized CRC Generator/Checker"*, [online: http://www.altera.com/literature/ds/dscrc.pdf]

[54] Altera Corp. *"Implementing CRCCs in Altera Devices"*, [online: http://www.altera.com/literature/an/an357.pdf]

[55] Gorry Fairhurst, *"Digital Communications Course Notes"*, [online: http://www.erg.abdn.ac.uk/users/gorry/course/phy-pages/man.html]

[56] J. Stone, M. Greenwald, C. Partridge, and J. Hughes, *"Performance of Checksums and CRCs over Real Data"*, Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, Aug. 2000.

[57] Chris Borrelli, *"Xilinx Application Note 209, IEEE 802.3 Cyclic Redundancy Check"*, [online: http://www.origin.xilinx.com/bvdocs/appnotes/xapp209.pdf]

[58] Rajesh Nair, Gerry Ryan and Farivar Farzaneh, *"A Symbol Based Algorithm for Hardware Implementation of Cyclic Redundancy Check (CRC)"*, 1997 VHDL International User's Forum (VIUF '97), p. 82

[59] A.X.Widmer, P.A.Franaszek: *"A DC-balanced, partitioned-block, 8B/10B transmission code"*, IBM Journal of Research and Development, vol 27, no 5, 1983, pp. 440-451

[60] IEEE, *"IEEE 1394b High-Performance Serial Bus."*, 2002, standard.

[61] E. Solari and B. Congdon, *"The Complete PCI Express Reference"*, Hillsboro, OR: Intel Press, 2003.

[62] L. B. James, A. W. Moore, and M. Glick, *"Structured Errors in Optical Gigabit Ethernet",* Passive and Active Measurement Workshop (PAM 2004), Apr. 2004.

[63] Altera Corp., *"8B/10B IP Core"*, [online: http://www.altera.com/products/ip/communications/codec/m-alt-ed8b10b.html]

[64] Altera Corp. *"Stratix GX Device Family overview"*,
[online: http://www.altera.com/products/devices/stratixgx/overview/sgx-overview.html]

[65] Altera Corp. *"Stratix GX transceiver user guide"*, March 2004,
[online: http://www.altera.com/literature/ug/ug_sgx.pdf]

[66] P.Moreira et al, *"G-Link and Gigabit Ethernet Compliant Serializer for LHC Data Transmission"*, (presented at the IEEE Nuclear Science Symposium and Medical Imaging Conference, 2000),
[online: http://proj-gol.web.cern.ch/proj-gol/publications/paperNSS155.pdf]

[67] G.Anelli, M.Campbell, M.Delmastro, F.Faccio, S.Florian, E.Heijne, P.Jarron, K.Kloukinas, A.Marchioro, P.Moreira, W.Snoeys *"Radiation tolerant VLSI Circuits in Standard deep submicron CMOS technologies for the LHC experiments: Practical design aspects"*, Proceedings of NSREC 1999,IEEE Trans. on Nucl.Sci., vol. 46, no.6, Dec. 1999.

[68] Cadence Design Systems, Inc. *"Clock Domain Crossing, Closing the loop on clock domain functional implementation problems"*, Technical Paper, Dec 2004,
[online: http://www.cadence.com/whitepapers/cdc_wp.pdf]

[69] Altera Corp. *"AN042: Metastability in Altera Devices"*, version 4, May 1999,
[online: http://www.altera.com/literature/an/an042.pdf]

[70] Chris Wellheuser, Texas Instruments Incorp. *"Metastability Performance of Clocked FIFOs"*, [online: http://focus.ti.com/lit/an/scza004a/scza004a.pdf]

[71] Altera Corp. *"TriMatrix Memory in Stratix Devices"*,
[online: http://altera.com/products/devices/stratix/features/stx-trimatrix.html]

05

Data Acquisition & Merging Algorithm

Chapter 5. Data Acquisition & Merging Algorithm

In this chapter, a strategy for the data acquisition from each detector by the two systems involved, the CFC and the ADC, will be proposed.

Moreover, based on the fact that the two types of data acquired for each detector are different, a pre-processing is needed for them to be combined seamlessly. An implementation is given that tackles this need as well as the noise coming from the ADC measurement.

Finally, the aim will be to achieve the functionality of the above processes in an efficient way that uses the least possible time and area for their calculations, i.e. clock cycles and logic elements needed.

5.1 Data Acquisition

The data from the two acquisition systems discussed in Chapter 2, the CFC and the ADC, available at the tunnel electronics have to be acquired in such way that correspond as closely as possible to the same moment.

One strategy to achieve this could be to read the counter data when the ADC conversion starts. Then the Counter data will need to be held in a register, for ~200ns, the same time needed for the ADC data to become available.

Moreover, at the counter a reset can be applied immediately after its value has been read and stored to the count register. The pulses (coming from the monostable) that feed the counter are ~100ns long and a reset can be made within one clock cycle (i.e. 25ns). Thus, there is no fear of losing any counts coming from the CFC circuitry. In that way, the counter data will represent the number of counts accumulated in the last 40μs without any extra data processing.

Figure 5-1 illustrates the acquisition systems involved as well as the process needed for each detector to control them in the CFC card's FPGA.

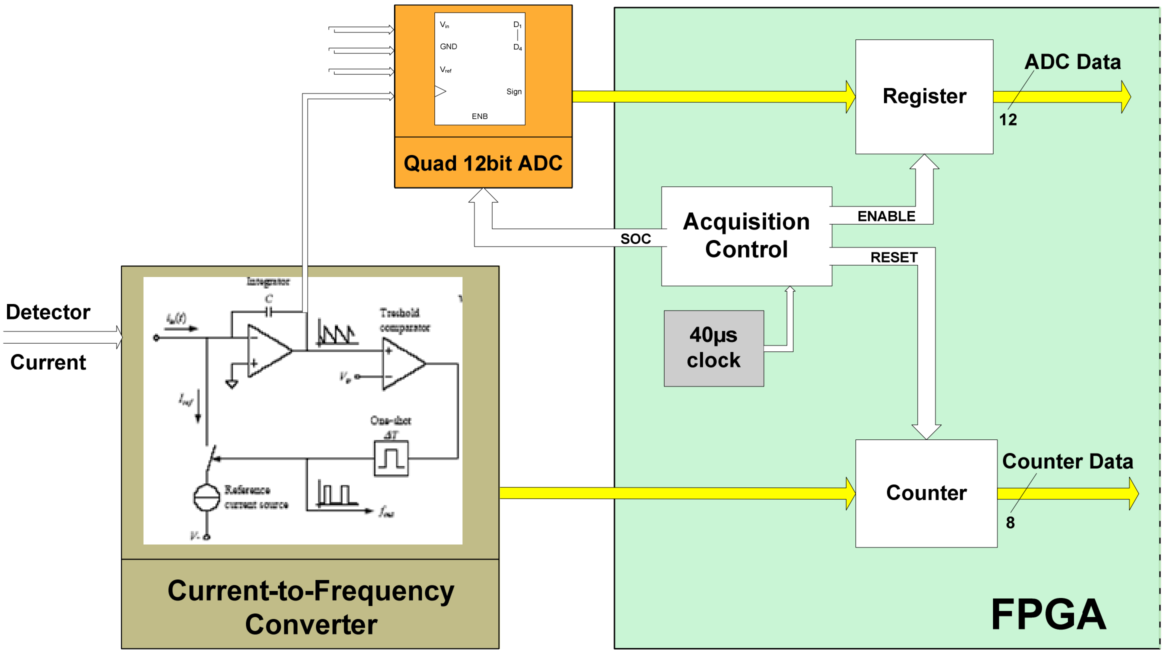


Figure 5-1: Systems Used for the Data Acquisition.

In Figure 5-2 is shown the timing waveform of a read cycle with the signal assertions needed for initiating such an action. The SOC (start of conversion) output pin of the FPGA will initiate the analogue to digital conversion. The SOC will give one pulse every 40µs and all the rest of the processes will only need to be synchronised to that pulse and the 25ns period system clock.

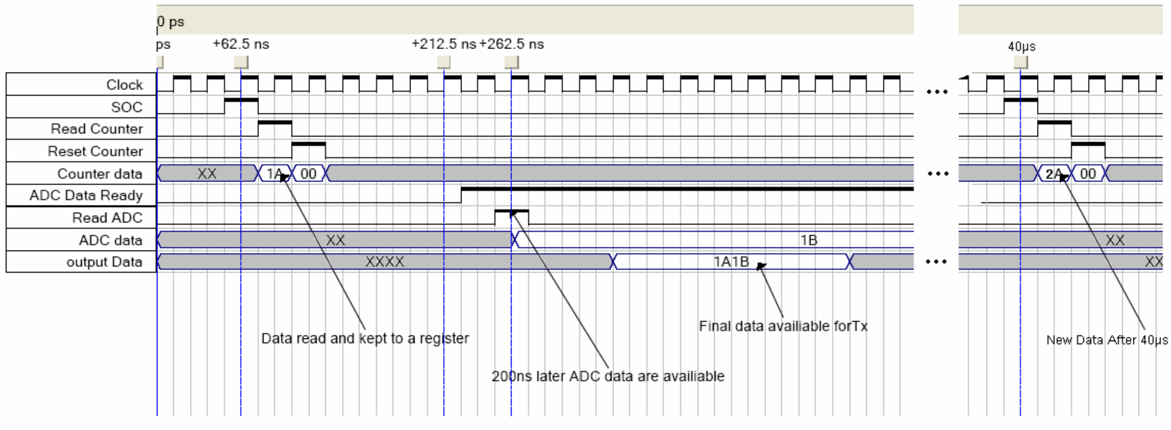


Figure 5-2: Timing waveform of read cycle.

5.2 Data Merging Algorithm

Based on the fact that the two types of data acquired for each detector are different pre-processing is needed in order for those to be merged. The measurement of the frequency, produced by the Current-to-Frequency Converter, with a counter relates to the average current induced from the last acquisition. On the other hand, the voltage measured by the ADC is the fraction of the charge in the capacitor.

In order to merge those data the difference of the last two ADC measurements is needed. It will then correspond to the counter fraction of the last 40μs and thus could be added to the counter value. This could be described in an equation as:

$$\text{Value}_{\text{DETECTOR}}(n) = \text{Value}_{\text{COUNTER}}(n) + [\text{Value}_{\text{ADC}}(n-1) - \text{Value}_{\text{ADC}}(n)]/2^{N} \qquad [5\text{-}1]$$

Where, $\text{Value}_{\text{COUNTER}}$ and $\text{Value}_{\text{ADC}}$ are the recorded values from the counter and the ADC respectively, and N is the number of bits used from the ADC. The difference is divided by its full scale in order to be normalised as a fraction.

5.2.1 Positive Difference Result

For ease of explanation, let us assume that the CFC card has recorded the situation shown in Figure 5-3. There were 2 acquisitions occurred, *acq(n)* and *acq(n+1)*, with a difference in time of 40μs. From the graph it can also be observed that the data send at *acq(n)* had an ADC value of 80% of its scale and similarly the data send at the second acquisition *acq(n+1)* had 2 counts as the counter data and 35% of the ADC scale as the ADC data.

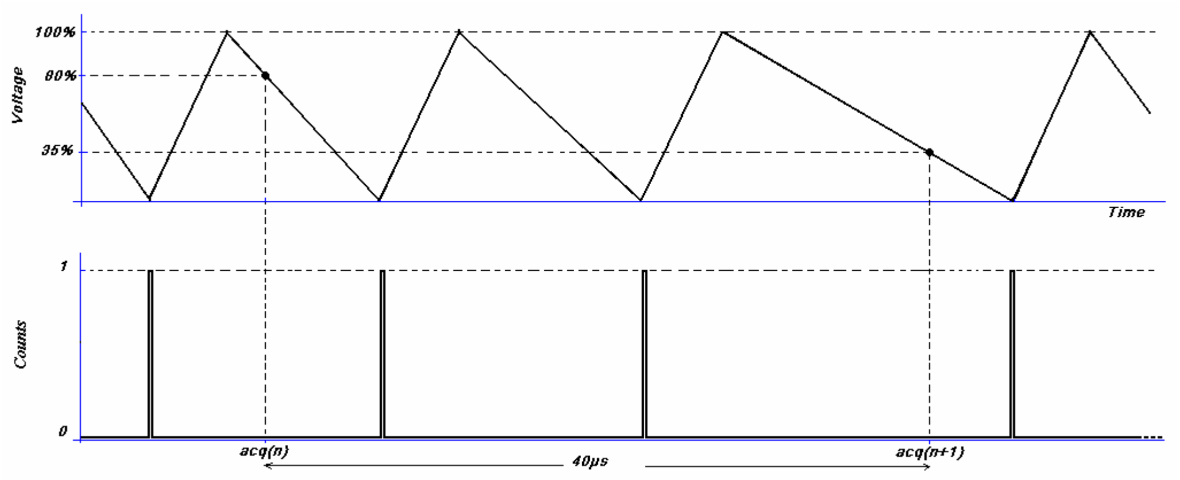


Figure 5-3: ADC and CFC Output (1st case).

It is possible to derive, using those data, what the system has accumulated in the 40μs between the two acquisitions for this detector. It was 2 counts and 80 – 35 = 45% of a count or differently 2.45 counts.

5.2.2 NEGATIVE DIFFERENCE RESULT

Nevertheless, in order to cover all possible cases, it is necessary to check what happens when there was a negative result in the subtraction of the two ADC values. Figure 5-4 shows an example of this case. In the first acquisition, the ADC data was measured as 35% of the scale and in the second acquisition, the counter has accumulated two counts and the ADC has given 80% of its scale.

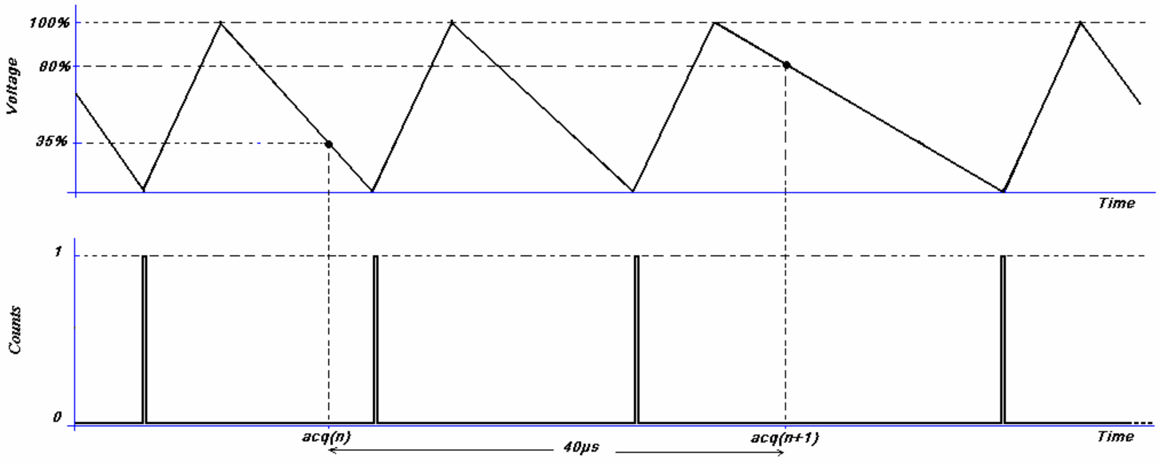


Figure 5-4: ADC and CFC Output (2nd case).

Doing the same calculation for this set of inputs it will give: 2 counts and 35 – 80 = – 45% of a count which basically means that in the observation window between the two acquisitions the detector has accumulated 1 count and 55% or 1.55 counts.

5.3 Noise on the ADC Output

Apparently, the situation described in Section 5.2 is an ideal situation where the charge in the capacitor is discharged proportionally depending only on the current given to the input of the CFC by the detector. In reality, various noise sources are introduced in all stages of this circuit from the power supply, the CFC itself, and the ADC.

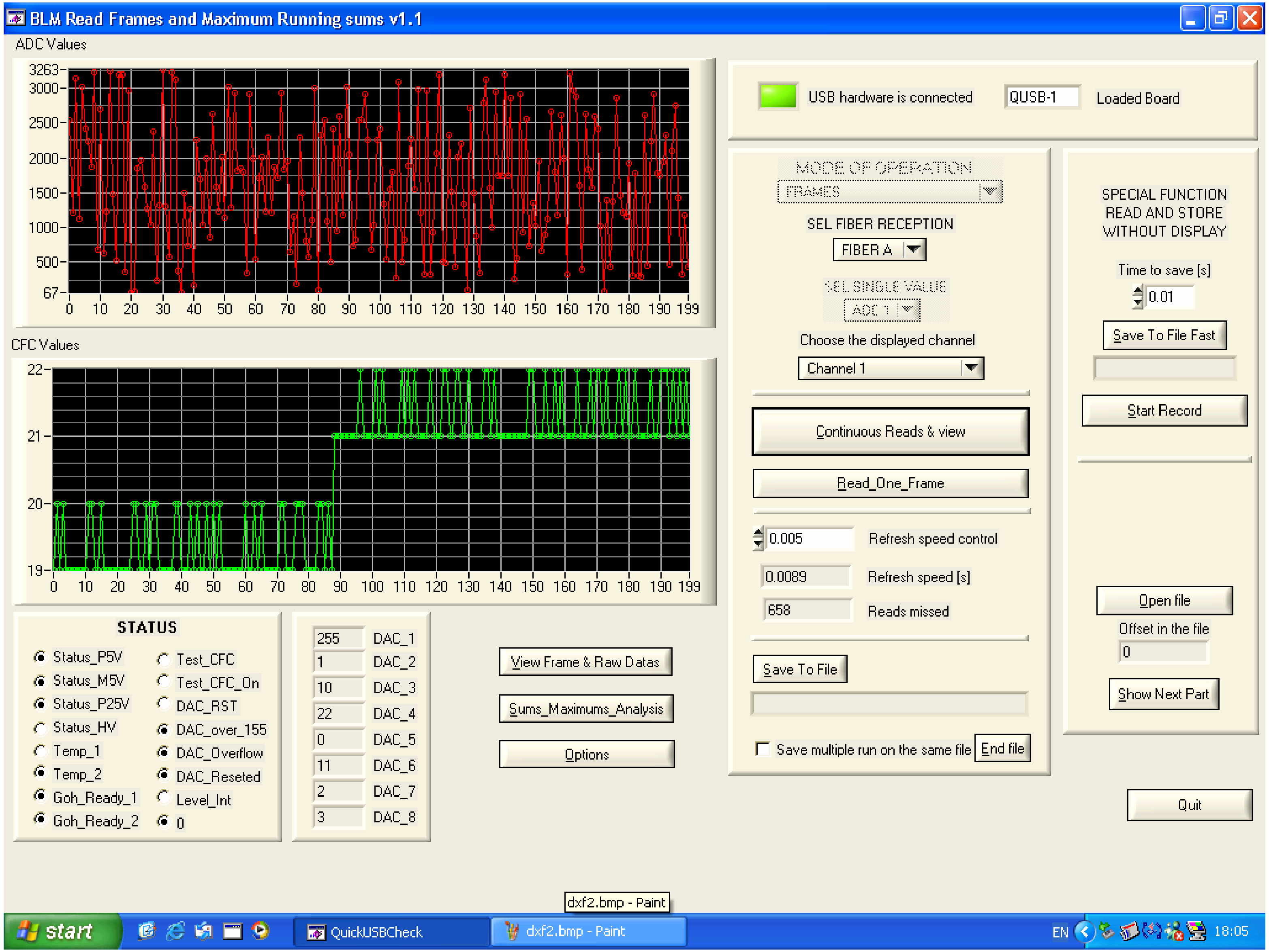
Even though, the purpose of this thesis was not to find those sources of noise, nor ways to improve this part, it was necessary to evaluate what kind of data the Data Analysis system would get. Then, by using this information, it would be possible to choose what sort of filtering could be applied to make the data merging strategy possible as well as efficient.

5.3.1 Visualisation of the Noise Pattern

One of the first systems used to visualise the data was assembled by a BLM Mezzanine, an FPGA development board from Microtronix [73] and a high-speed USB2.0 [74] connectivity solution called QuickUSB module [75] from Bitwise.

Using this configuration, the data transmitted from the CFC card was then available in the PC environment. An application was consecutively being developed [76] using the National Instruments LabVIEW [77] software that could sort and store those data for online display or post-processing. The LabVIEW interface produced is shown in Figure 5-5. A zoom in the online display panel can be seen in Figure 5-6. This plot is providing the ADC values, using a decimal base, that have been measured over the last 500 acquisitions.

Using this configuration, sample files with 10,000 acquisitions held on each of those files was possible to be stored and will be used in the following sections to observe and provide solutions for each situation.



USB hardware is connected

QUSB-1 Loaded Board

MODE OF OPERATION: FRAMES

SEL FIBER RECEPTION: FIBER A

SEL SINGLE VALUE: ADC 1

Choose the displayed channel: Channel 1

0.005 Refresh speed control

0.0089 Refresh speed [s]

658 Reads missed

□ Save multiple run on the same file

SPECIAL FUNCTION
READ AND STORE
WITHOUT DISPLAY

Time to save [s]: 0.01

Offset in the file: 0

STATUS

● Status_P5V	○ Test_CFC
● Status_M5V	○ Test_CFC_On
● Status_P25V	○ DAC_RST
○ Status_HV	● DAC_over_155
○ Temp_1	● DAC_Overflow
● Temp_2	● DAC_Reseted
● Goh_Ready_1	○ Level_Int
● Goh_Ready_2	● 0

255	DAC_1
1	DAC_2
10	DAC_3
22	DAC_4
0	DAC_5
11	DAC_6
2	DAC_7
3	DAC_8

Figure 5-5: LabVIEW Online Display Application GUI.

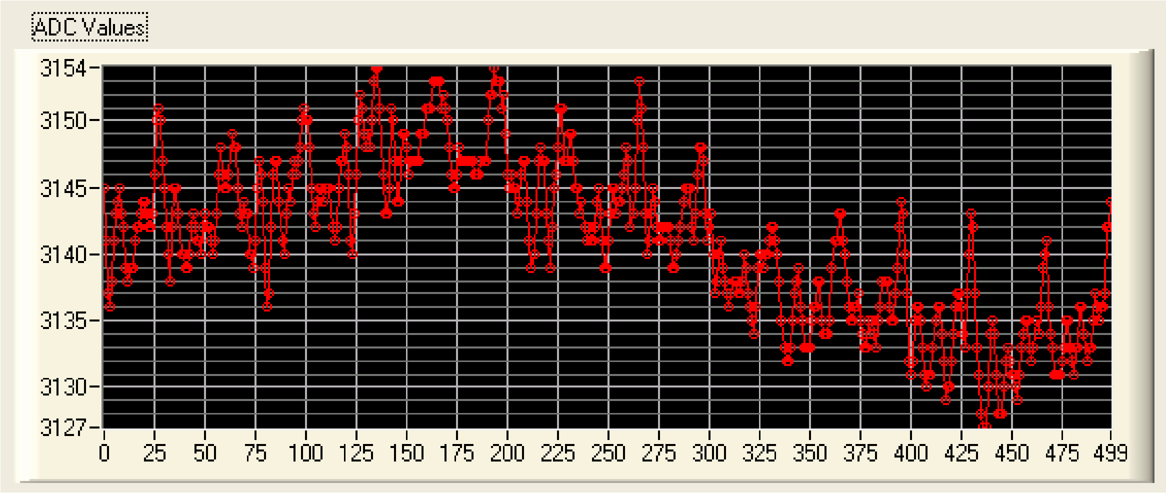


Figure 5-6: Recording and display of the raw ADC Data using a LabVIEW application.

Plotting any of those stored files, similarly to what has been produced in Figure 5-7, shows clearly that there is not a linear discharge of the capacitor when a constant current is applied to the input. Instead, there are positive jumps recorded at the ADC output from one acquisition to the next that can span up to the fifth LSB. Additionally, it allows the observation of the presence of a sinusoidal element on the ADC output (series name: ADC_out). Figure 5-8 is a zoom-in on the same acquisition file showing only the first 1,000 samples.

5.3.2 MINIMUM-VALUE-HOLD FUNCTION

Moreover, on both of the figures (Figure 5-7 & Figure 5-8) it is included one more series of values, called ADC_min, which is the calculated output of a Minimum-Value-Hold (MVH) function. This function operates in the way that it keeps the minimum value seen from the ADC. At every new acquisition it receives, it compares the ADC value with one held in a register. If it is found to be smaller then it updates the value of the register with that new minimum value. If the new value is larger than the value of the register, i.e. it is larger than the minimum, the register simply retains its value and waits for a new acquisition.

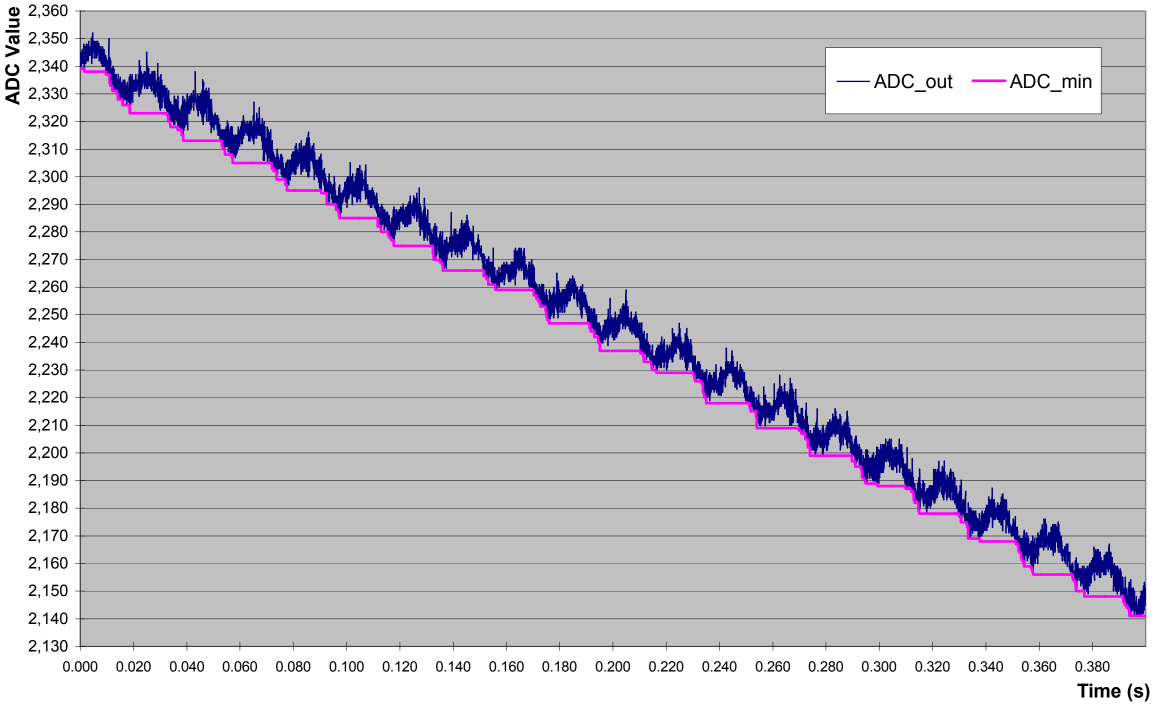


Figure 5-7: Plot (10,000 samples recording) of the ADC data output (ADC_out) and after the Minimum-Hold-Value function (ADC_min).

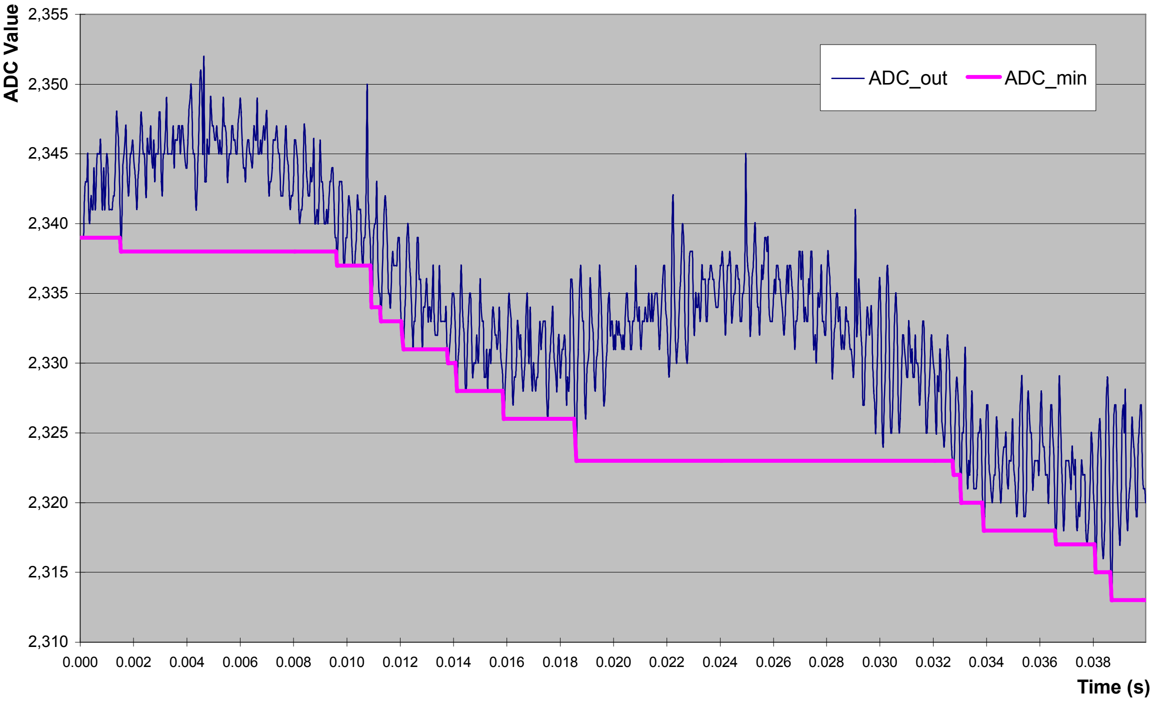


Figure 5-8: Plot (1,000 samples recording) of the ADC data output (ADC_out) and after the Minimum-Hold-Value function (ADC_min).

5.3.3 DATA-COMBINE PROCESS EXPECTED OUTPUTS

Concurrently, the Data-Combine process is checking that the loss rate is less than one count per acquisition, i.e. very slow losses, and enters to an operation mode specific for this. When in this mode, it outputs the ADC value change by calculating the difference between the new value and the minimum value if this new value is smaller than the minimum. Otherwise, the output is equal to zero.

In Figure 5-9 is shown the ADC fraction output using the first 1,000 samples from the data file. It is easily extracted from the figure that on the beginning of the observation the ADC had a value of 2,339 and 40ms later a value of 2,313. Subtracting those, gives a decrease of 26 points, which is also equal to the sum of the ADC fractions outputted at the same time period.

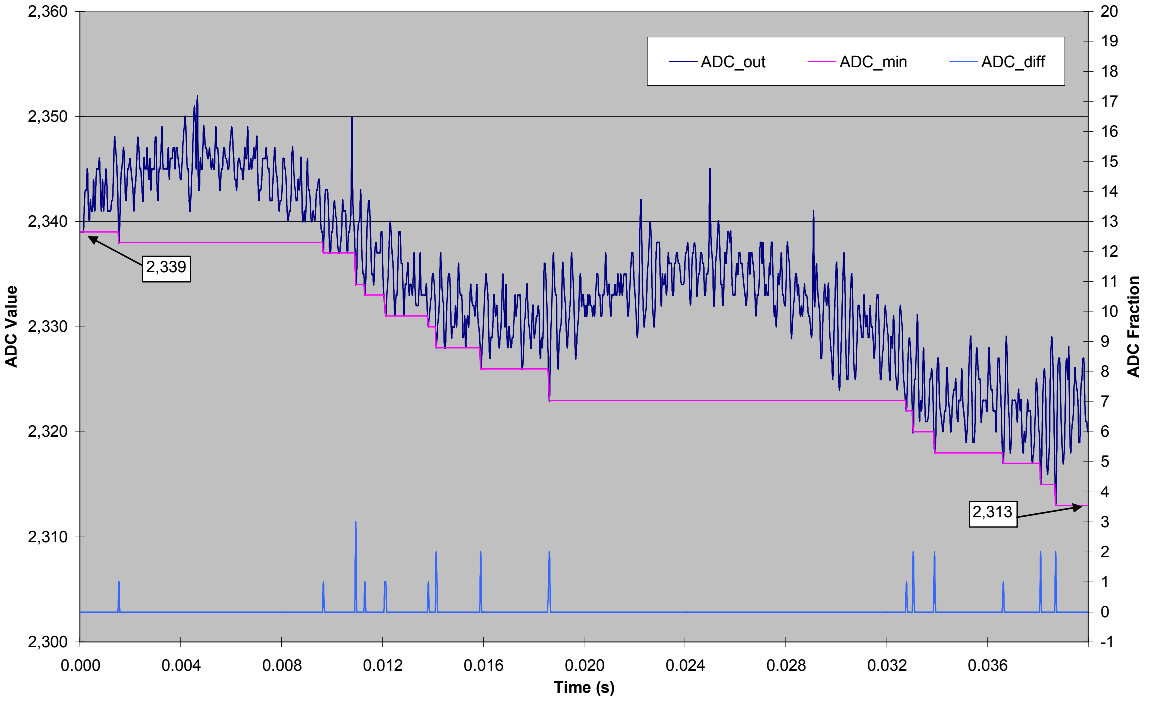


Figure 5-9: ADC Fraction Output when using the Minimum-Value-Hold Function.

During the very slow losses, the charge of the CFC capacitor will eventually become zero since it is constantly discharging and charged back to its full scale followed by a pulse to the counter input. The Data-Combine process will receive this count but it must be made sure that it does not consider this as a real count since it is merely a change of state. More, it has to reset the Minimum Value register for its correct operation.

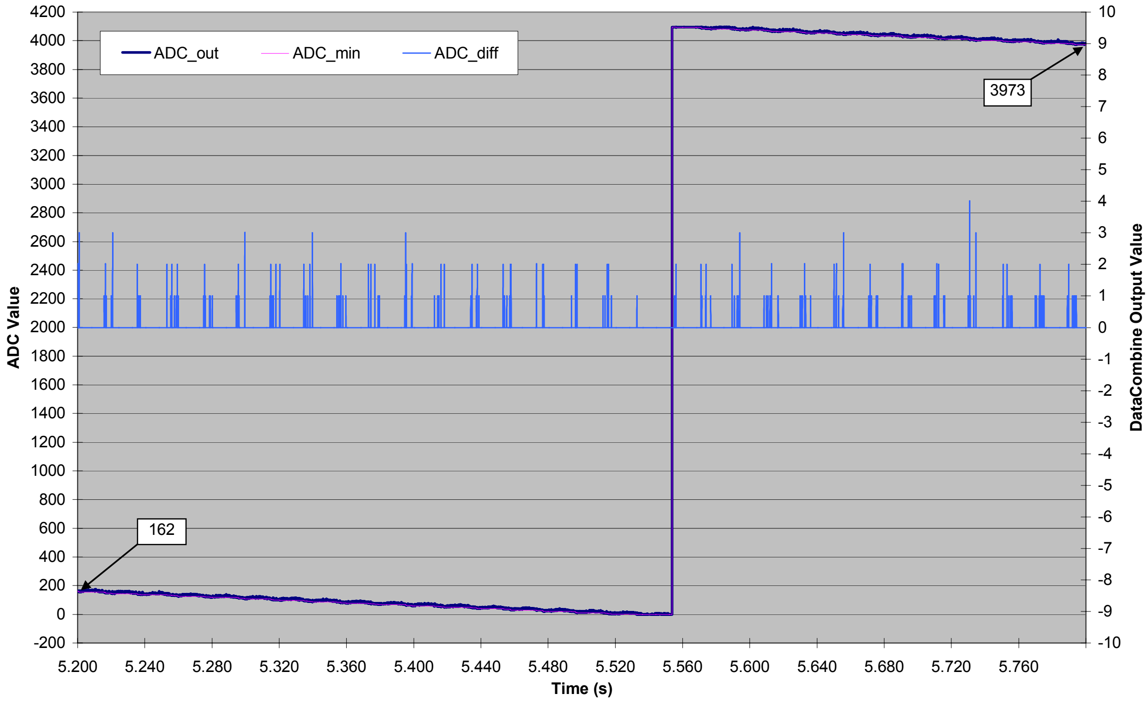


Figure 5-10: Data-Combine Output when using the Minimum-Value-Hold Function.

In Figure 5-10 it is shown the situation where the capacitor charge eventually reached to zero and a count was transmitted followed by a recharge. The wanted Data-Combine output values can also be observed in this figure. Similarly to the previous example, it can be extracted from the figure that the value of the ADC at the start of the observation window is 162 points and at the end is equal to 3973 points. The sum of the Data-Combine output for the same period of time is 284 points which can also be calculated as 162 + (4095 - 3973) = 284.

5.4 Systematic Error Due to the Data-Combine Process.

In the previous section, it has been proven that the proposed strategy for the Data-Combine process will output correctly the change of the input over a period of time in all possible situations. Thus, the processing stages later in the BLMTC design, which will use this value to provide the integration over different time periods, will receive the valid data.

Nevertheless, there is a systematic error imposed by the chosen processing. It is expected to have an impact, at least to the short integration times, when there are very slow losses. This is a result of the MVH function. The resolution of its output is inverse proportional to the swing of the ADC output due to the noise.

This can be visualised in Figure 5-11 where it is seen that the fraction output instead of being uniformly distributed in time is concentrated into periodic regions with higher concentrations. Those regions are expected to have larger spacing between them with higher values if higher amplitude noise is propagated in the system.

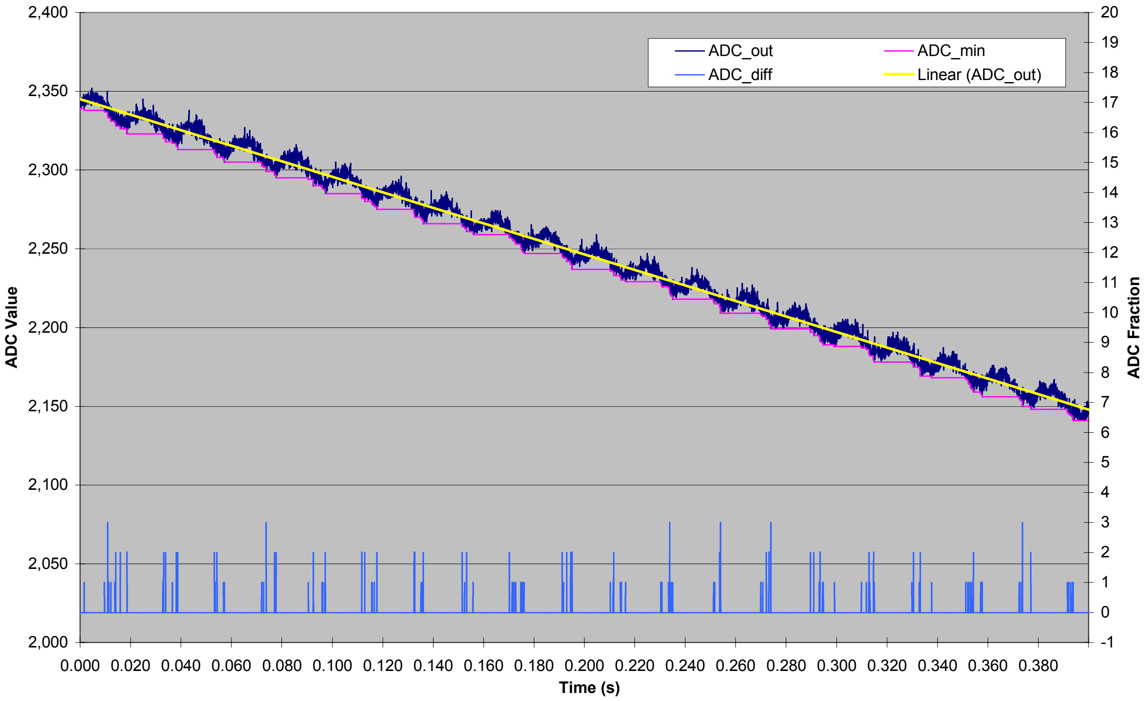


Figure 5-11: Over-estimation when using the Minimum-Value-Hold (MVH) Function.

This effect has been observed to be higher than the error from the digitisation by the ADC circuitry and leads to the use of only the ten most significant bits from the 12-bit ADC output. Nevertheless, the dynamic range covered by the 18-bits, that is 8 from the counter and 10 from the ADC, is still enough to cover the requirements.

5.5 Implementation of the Data-Combine Process

Taking into consideration the above and without forgetting that the above data are all conveyed throughout in binary form then it is wise to use 2's complement signed numbers representation in most of the system. Moreover, since this process makes use of mathematical functions and bit manipulations it is simpler to be described using the VHDL language.

Figure 5-12 illustrates the needed functionality of the Data-Combine block and its realisation in VHDL code can be found in the Appendix Section A.2.

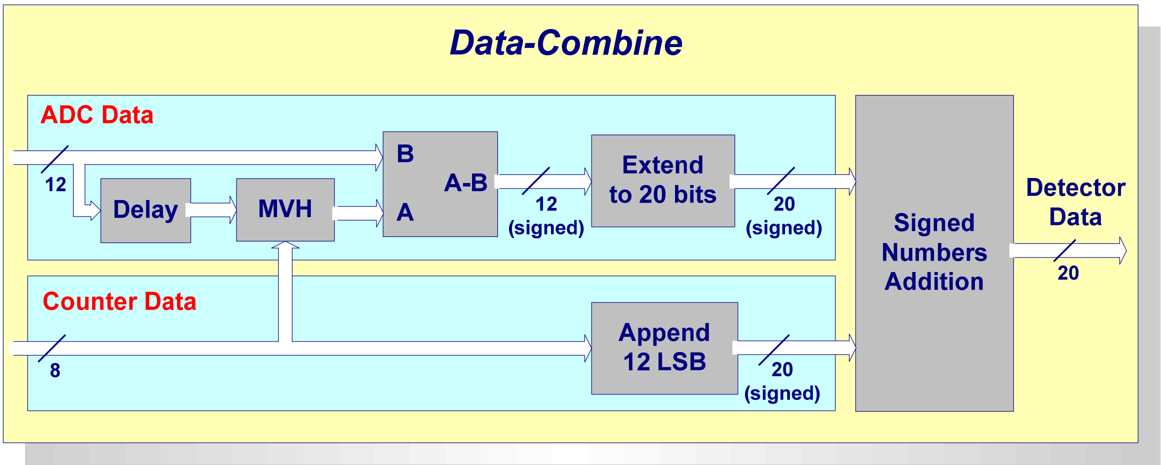


Figure 5-12: Overview of the Data-Combine Process.

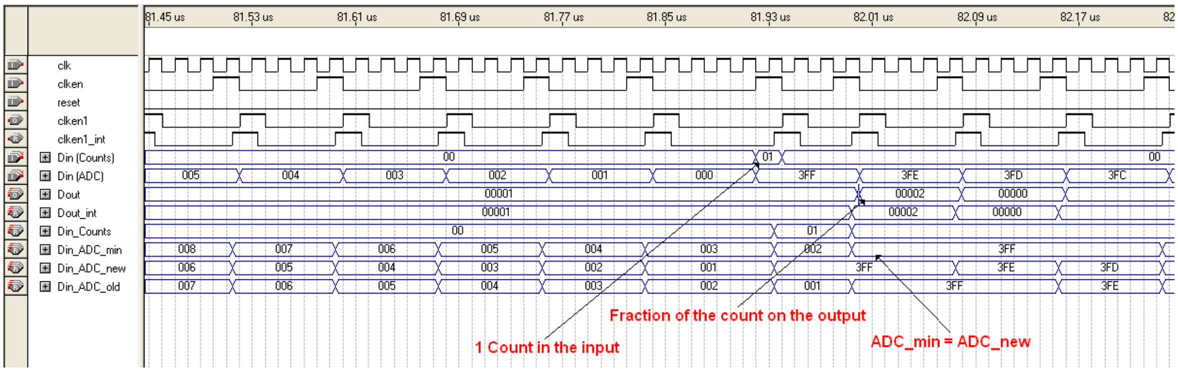
Table 5-1 summarises the resource utilisation of the Data-Combine process when compiled by the Quartus II Fitter. The achieved final number of Logic Elements used, i.e. 115, is sufficiently low enough since this process will be necessary to be placed in the BLMTC design 16 times, once for each detector.

Table 5-1: Data-Combine Resource Utilisation Summary

Resource	Usage
Total logic elements	115 / 32,470 (< 1 %)
-- Combinational with no register	58
-- Register only	16
-- Combinational with a register	41
Total LABs	43 / 3,247 (1 %)
Logic elements in carry chains	58

5.6 Static Timing Simulation of the Data-Combine Process

The first step for the verification of the correct implementation of the Data-Combine process is a time-driven back-annotated simulation using the Quartus II Simulator. More information about this verification tool and the setting used can be found in Section 2.8.



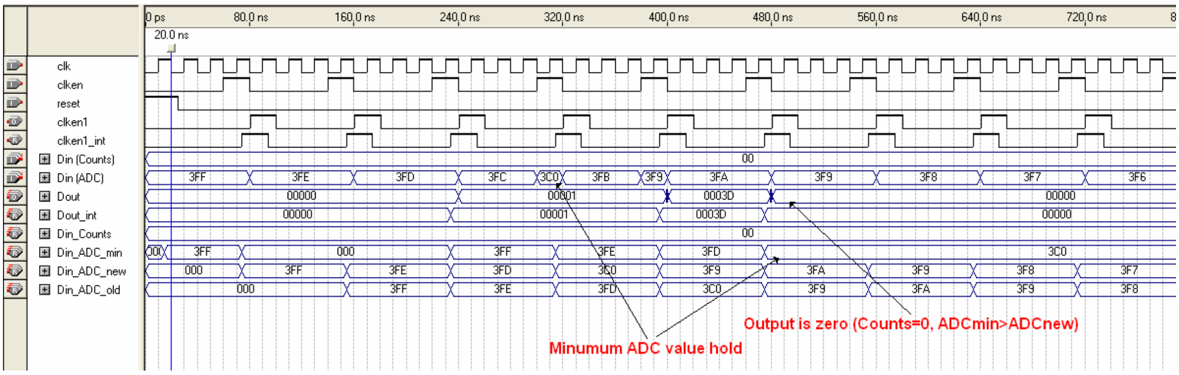
Fraction of the count on the output

1 Count in the input

ADC_min = ADC_new

Figure 5-13: Timing Simulation / Case of very slow losses

The needed functionality of the Data-Combine process during the very slow losses is shown in Figure 5-13. During the CFC change of state, the one count transmitted is not accounted but correctly only the correct fraction is outputted.



Output is zero (Counts=0, ADCmin>ADCnew)

Minumum ADC value hold

Figure 5-14: Timing Simulation / Minimum Value is retained.

The simulation waveform output part that shows the correct operation of the Minimum-Value-Hold function is shown in Figure 5-14. The minimum value is retained and upward changes in the ADC values, while in the slow losses mode, are not giving any fraction change to the output, i.e. the *Dout*, remains zero.

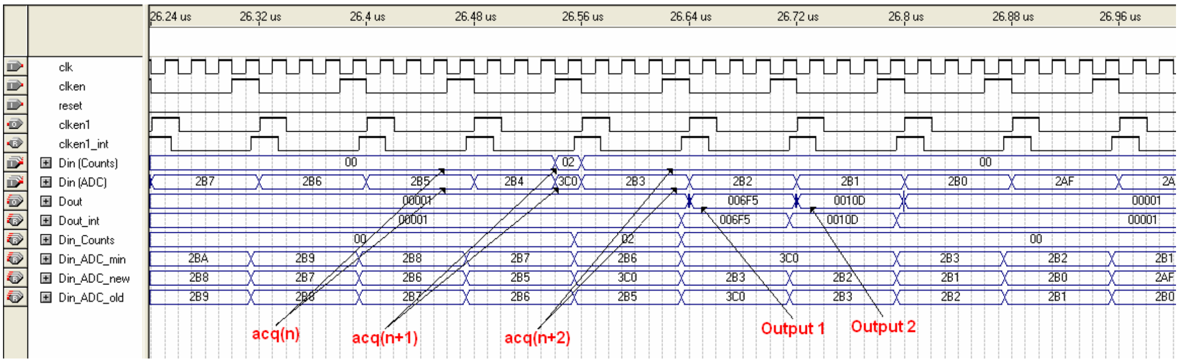


Figure 5-15: Timing Simulation / Data-Combine Function Validation.

Moreover, in Figure 5-15 the Data-Combine process is shown to be able to change correctly between its different operating modes and provide the wanted output. It is verified by looking closer at the acquisitions that took place, that the outputs this process gives match the expected values. In Table 5-2 is shown, for this comparison, the received data and the process outputs together with the calculated/expected outputs.

Table 5-2: Comparison of Acquisition Outputs with Expected Values.

Acquisition	Counts	ADC	Output	Expected Value
acq(n-1)	0	0x2B6	0x00001	-
acq(n)	0	0x2B5	0x00001	0x000 + (0x2B6 - 0x2B5) = 0x001
acq(n+1)	2 (= 0x800)	0x3C0	0x006F5	0x800 + (0x2B5 - 0x3C0) = 0x6F5
acq(n+2)	0	0x2B3	0x0010D	0x000 + (0x3C0 - 0x2B3) = 0x10D

5.7 Hardware Verification Test of the Data-Combine Process

Using SignalTap II embedded logic analyser, a tool provided for the verification in the Quartus II environment, the process can be tested on hardware and in real-time. This tool and the general settings used are shown in more detail in Section 2.8.

For testing this process some of the internal nodes were decided to be included to help the observation. Those were the Minimum-Value-Hold function output (= *Din_ADC_min*), the newly received ADC value (= *Din_ADC_new*), the delayed ADC value (= *Din_ADC_old*), and the newly received Counter value (= *Din_Counts*). Selecting those nodes for observation with the 40MHz system clock and using all four M4K memory blocks [71] in the EP1S30 chip has resulted in 32KSamples maximum acquisition length.

Figure 5-16 shows one of those observation windows. The system in this situation is setup to very slow losses and thus the counter values are constantly zero except the one count it receives when there is a change in the state of the ADC. The waveforms in this figure are comparable to those seen during the simulation and the output shows the expected values.

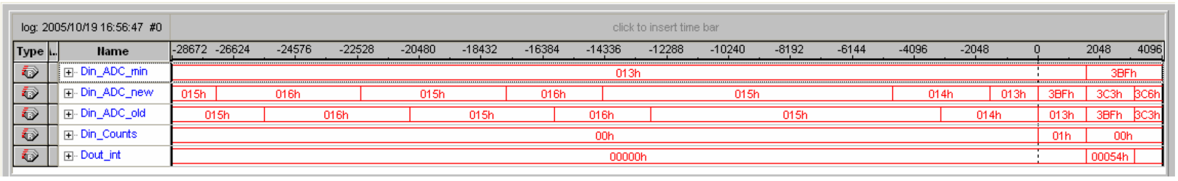


Figure 5-16: SignalTap II / Very Slow Losses - Correct Output.

On the contrary, in Figure 5-17, which is a similar situation and recorded a few minutes later, the output gives an unexpected value. With a closer observation, it is found that the source of the malfunctioning in the processing is that the counter value signifying the change of state comes on the following acquisition or simply delayed compared to the ADC value.

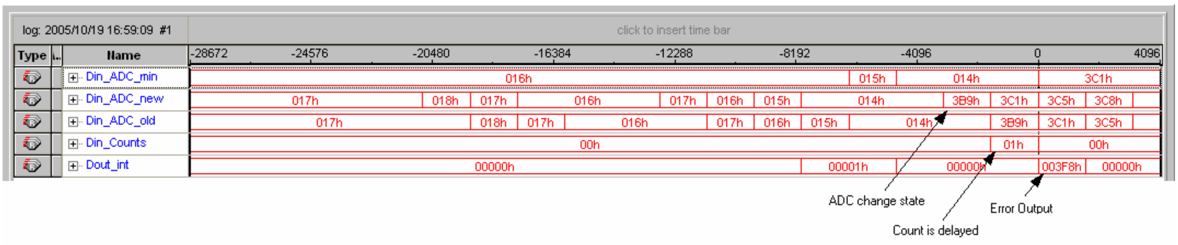


Figure 5-17: SignalTap II / Very Slow Losses - Error Output.

After a few more measurements and with the trigger set to the one count output, it was discovered that the CFC card was transmitting the transition most of the times synchronised but some times the counter value was delayed and was appearing in the following transmission. The Data-Combine process was revised to include and expect this behaviour so that it would provide a correct output in all cases. Figure 5-18 shows a recording of this case with the revised version.

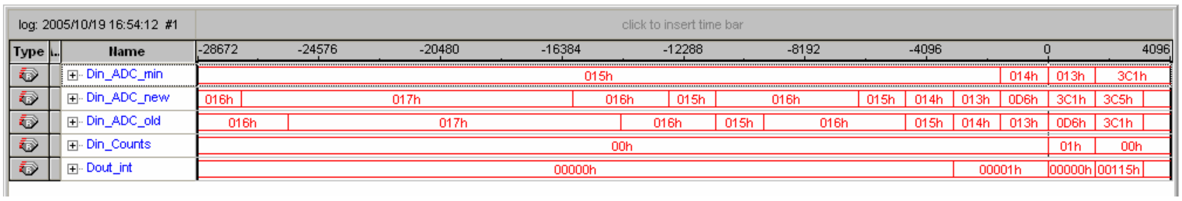


Figure 5-18: SignalTap II / Slow Losses – Revised Version to include delayed arrival of Count.

Finally, the process was tested for the case where fast losses were detected and the high number of counts was received correctly. In this situation, similarly to any other, where the number of counts received is higher than one, the MHV function does not give any contribution.

5.8 SUMMARY

In the BLM system, from each detector two different systems acquire data in parallel. Thus, two types of data are transmitted for each of them. An acquisition strategy and a pre-processing was necessary for them to be combined seamlessly.

The measurement with a counter of the pulses produced by the Current-to-Frequency Converter relates to the average current between the two last acquisitions. On the other hand, the voltage measured by the ADC is the voltage of the CFC integrator at the time of the counter readout. It represents then the fraction remained from the last count and the start for the next acquisition.

In order to combine those data the difference of the last two ADC measurements is calculated. It corresponds to the counter fraction of the last 40µs and thus could be added to the counter value. Of course, since the difference could be a negative number, signed number arithmetic was used for the addition.

Moreover, the system has to cope with the noise propagating through the ADC data. For this, the Minimum-Value-Hold function was implemented to be used when the system operates in the region of very slow losses detection. In order to understand the noise pattern, a test system was built that recorded and displayed on a PC the received data.

The data combining process had to compensate additionally the change of the state of the ADC and the counts transmitted because of this as well as to overcome the spuriously unsynchronised transmission of the two types of data.

Finally, this part of the design has exhibited the intended functionality in both the simulation and the in system hardware tests. Using this scheme, the usage of resources and computation effort for the whole system is minimised since it is concentrated in this beginning stage. All later stages are just exploiting those combined values.

References

[72] G.Minderico et al, *"A CMOS low power, quad channel, 12 bit, 40MS/s pipelined ADC for applications in particle physics calorimetry"*, presented at the 9th Workshop on electronics for LHC, 2003, [online: http://lhc-electronics-workshop.web.cern.ch/LHC-electronics-workshop/2003/sessionsPDF/Eleccal/MINDERIC.pdf]

[73] Microtronix, *"Microtronix Stratix Development Kit"*,
[online: http://www.microtronix.com/product_stratix.htm]

[74] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, *"Universal Serial Bus Specification Rev. 2.0"*

[75] Bitwise Systems, *"QuickUSB Module"*, [online: https://www.quickusb.com/index.htm]

[76] Application developed by Jonathan Emery, unpublished.

[77] National Instruments Corp. *"LabVIEW 5.1 Function & VI Reference Manual"*, 1998 [online: http://www.ni.com/labview/]

06

Real-Time Data Processing

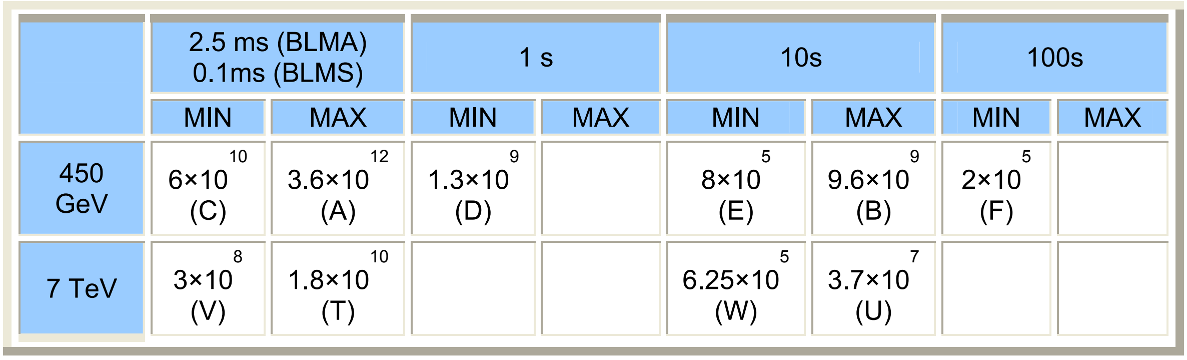
Chapter 6. Real-Time Data Processing

On systems that perform real-time processing of data, performance is often limited by the processing capability of the system. Processing data in real-time requires dedicated hardware to meet demanding time or space requirements. Modern field programmable gate arrays (FPGAs) include the resources needed to design complex processing structures with hundreds of I/Os, millions of logic gates, configurable interconnections, and embedded memory blocks. Furthermore, most of the manufacturers now include complete and fully parameterisable cores optimised for the device architecture that may offer more efficient logic synthesis and device implementation. This mix of powerful hardware and software can provide systems with the advantages (compared to an ASIC) of low development costs and short development cycle. In addition, the device can be reprogrammed making it ideal for future upgrades or system specification changes.

The strategy for machine protection and quench prevention is presently based on the BLM system. At each turn, there will be several thousands of data to record, process and transmit to the interlock system and display. The processing involves a proper analysis of the loss pattern in time (transient losses) and a proper account of the energy of the beam. This complexity must be minimized by all means to maximize the reliability of the BLM system as a whole.[78]

The dynamic range is the domain of variation of the beam losses inside which the calibration goal must be reached. The criteria used to define the dynamic range are based on the expected uses, the estimated loss levels and the strategy of machine protection.[78]

Table 6-1: Dynamic range for the BLMAs and BLMSs in p/m/s. [78]



	2.5 ms (BLMA) 0.1ms (BLMS)		1 s		10s		100s	
	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
450 GeV	6×10^{10} (C)	3.6×10^{12} (A)	1.3×10^{9} (D)		8×10^{5} (E)	9.6×10^{9} (B)	2×10^{5} (F)	
7 TeV	3×10^{8} (V)	1.8×10^{10} (T)			6.25×10^{5} (W)	3.7×10^{7} (U)		

Figure 6-1 shows the dynamic range needed for the two most widely used detector families of the LHC system. The loss rates marked by letters appear in Table 6-1, with their corresponding numerical values. The detector families are distinguished based on their topology and they are given in Table 7-2.

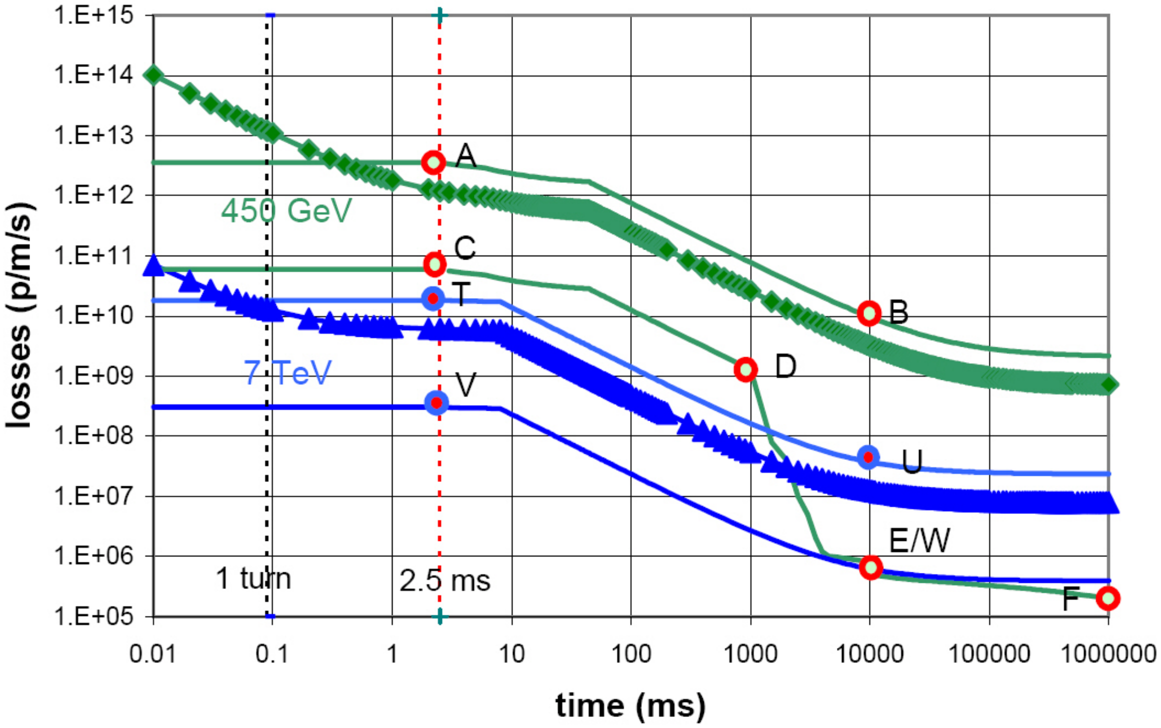


Figure 6-1: Dynamic range for the BLMAs and BLMSs. [78]

6.1 Choice of Data Processing Method

The procedure which in general will be followed is based on the idea that a constantly updated moving window can be kept by adding to a register the incoming newest value and subtracting from it the oldest. The number of values that are kept under the sum defines also the length of the moving window.

The ideal strategy to be followed, with the minimum achievable error, would be an infinite number of constantly updated moving windows with various lengths to cover the whole time region from 40μs to 100s. Such an implementation is not feasible, since it would need infinite amount of resources. Instead, it can be used as a reference in the comparison with any other processing method.

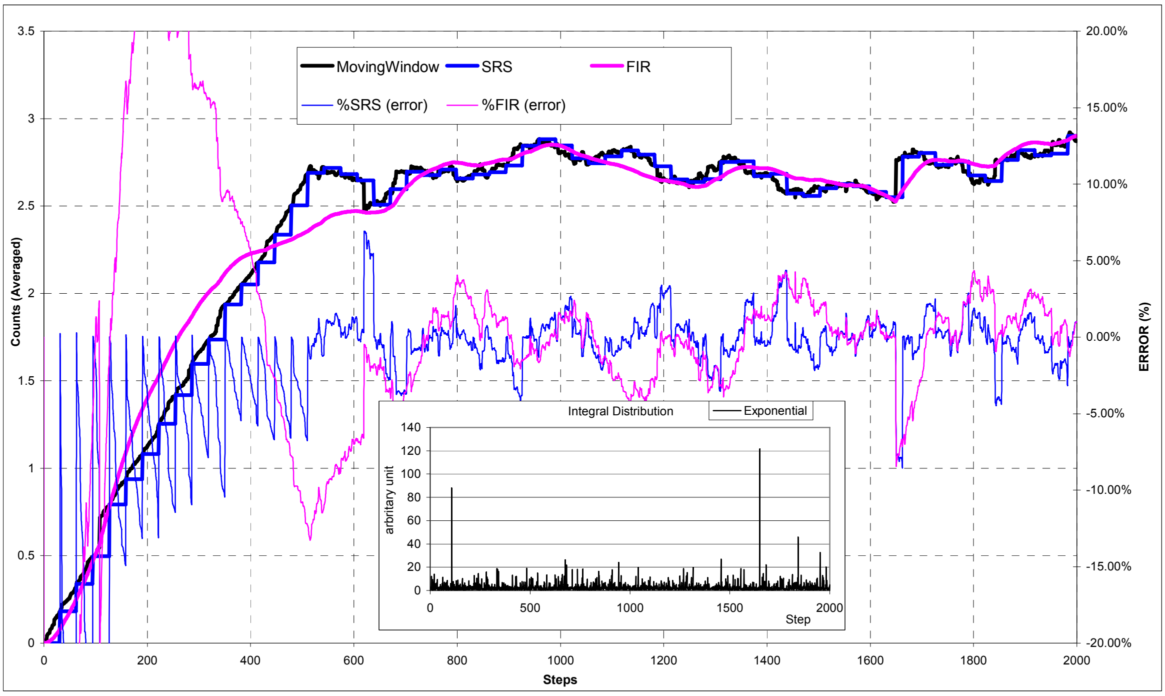


Figure 6-2: Data Processing Techniques Comparison (Case I).

Figure 6-2 and Figure 6-3 are examples of such comparison with different input signals. The Moving Window is the ideal response and two other data processing strategies are plotted for comparison in the same graph. The SRS, denoting a Successive Running Sum implementation, and the FIR, a 2nd order filter. Their relative error is also calculated and plotted.

In Figure 6-2 the input signal has a relative constant fluctuation with a small amplitude. The exception is given by two samples with high amplitude. This case is similar to the situation of the detectors recording fast losses. A second case is shown in Figure 6-3 where the input has a constant fluctuation and to that is added a linear increase. This situation is similar to when a loss gradually builds up in the accelerator.

It is clearly observable, in both figures, that the SRS is following closely the ideal with a delay of 16 samples. Its error line is recovering with the same speed to zero. On the contrary the 2nd order filter exhibits long periods of over or under-estimation and its error line crosses more than once the maximum 20% relative error allowed by the specifications. Additionally the filter has a very slow recovery period making it even more unattractive.

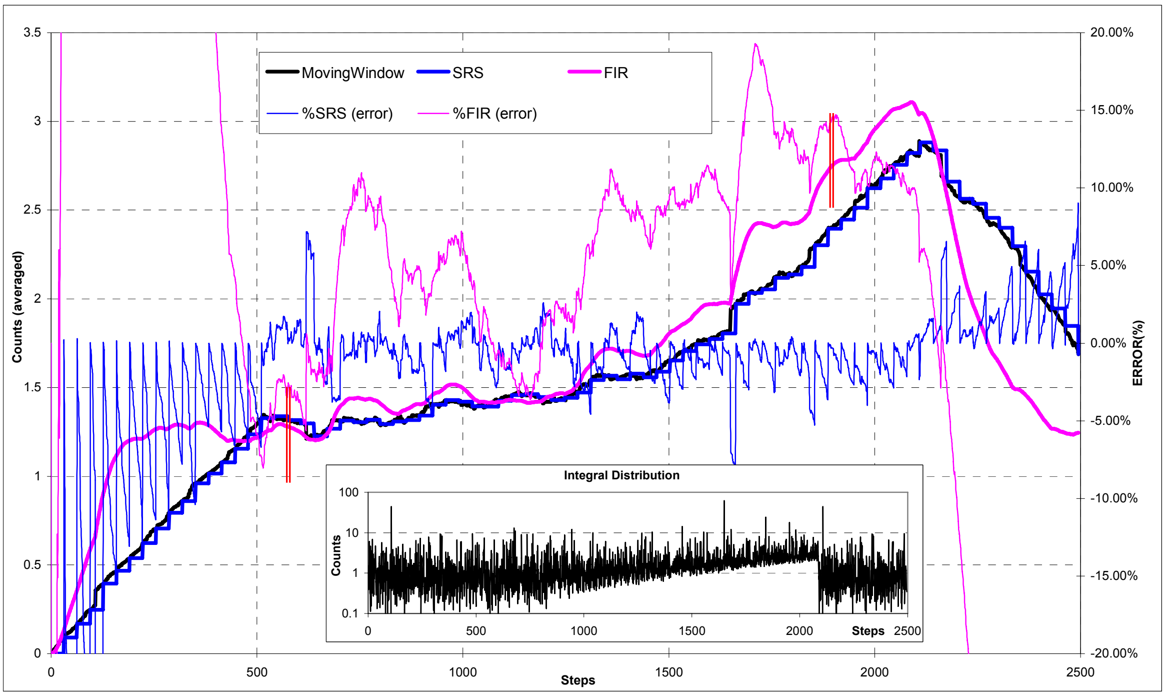


Figure 6-3: Data Processing Techniques Comparison (Case II).

The same conclusions are further supported by a more comprehensive study [80] on the loss patterns expected and the response of those systems. In this study, it is also included conventional inputs like the unit step, unit impulse, linear increase, as well as, Gaussian distribution in order to cover more possibilities and characterise more accurately each system.

6.2 Successive Running Sum (SRS) Data Processing

6.2.1 Running Sums

The Running Sums can be produced simply by adding the newly acquired value to a register and subsequently subtracting from it a value coming delayed by a number of cycles in a shift register. A similar configuration is used in the BLMTC system where, in order to increase the efficiency in resources, it is making use of the parameterised multipoint shift registers available for the Stratix devices. It is simply adding the difference of those two values, the new and the delayed, to an Accumulator.

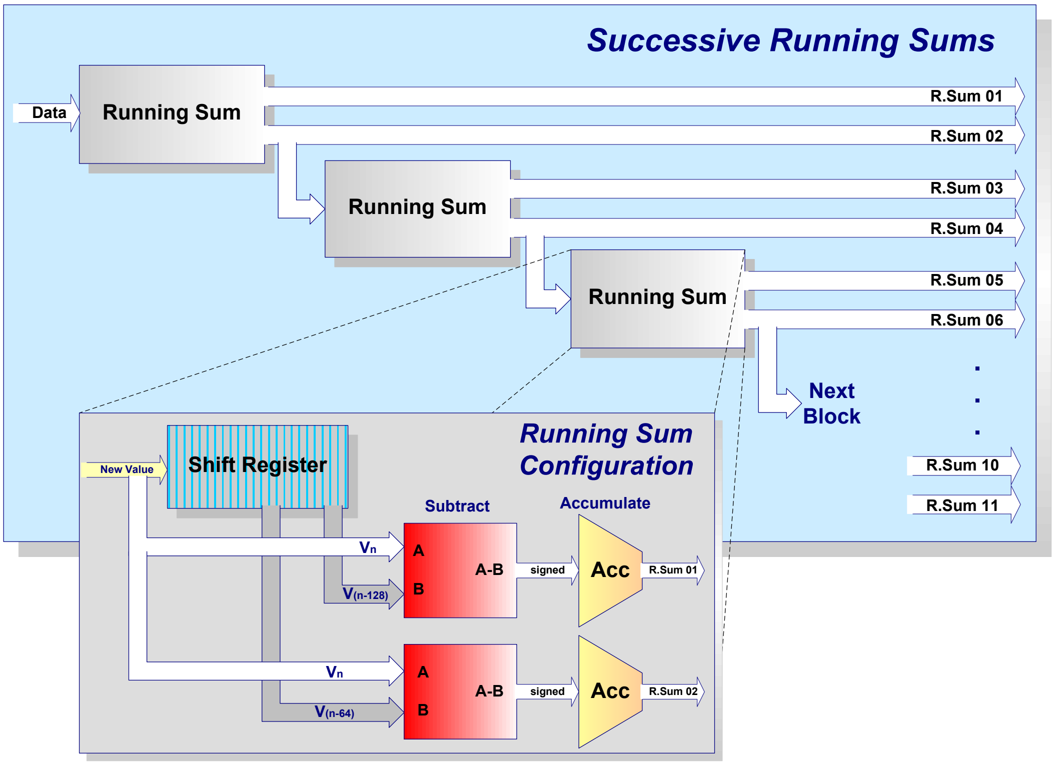


Figure 6-4: Production of Successive Running Sums.

This system will be used to create a moving window whose width relates to the number of values stored in the Shift Register. A block diagram of the Running Sum configuration can be seen in Figure 6-4. The "new value" corresponds to that received from the RCC process discussed in a previous Chapter and the "oldest value" is taken from the Shift Register where it was stored.

6.2.2 Successive Running Sums

One more scheme of resource sharing is employed by the Successive Running Sum technique. It is able to reach long integration periods with relatively small in length shift registers. In this technique, the storage problem of long histories of the acquired data, which are needed for the construction of long moving windows, is tackled by consecutive storage of sums of the received values.

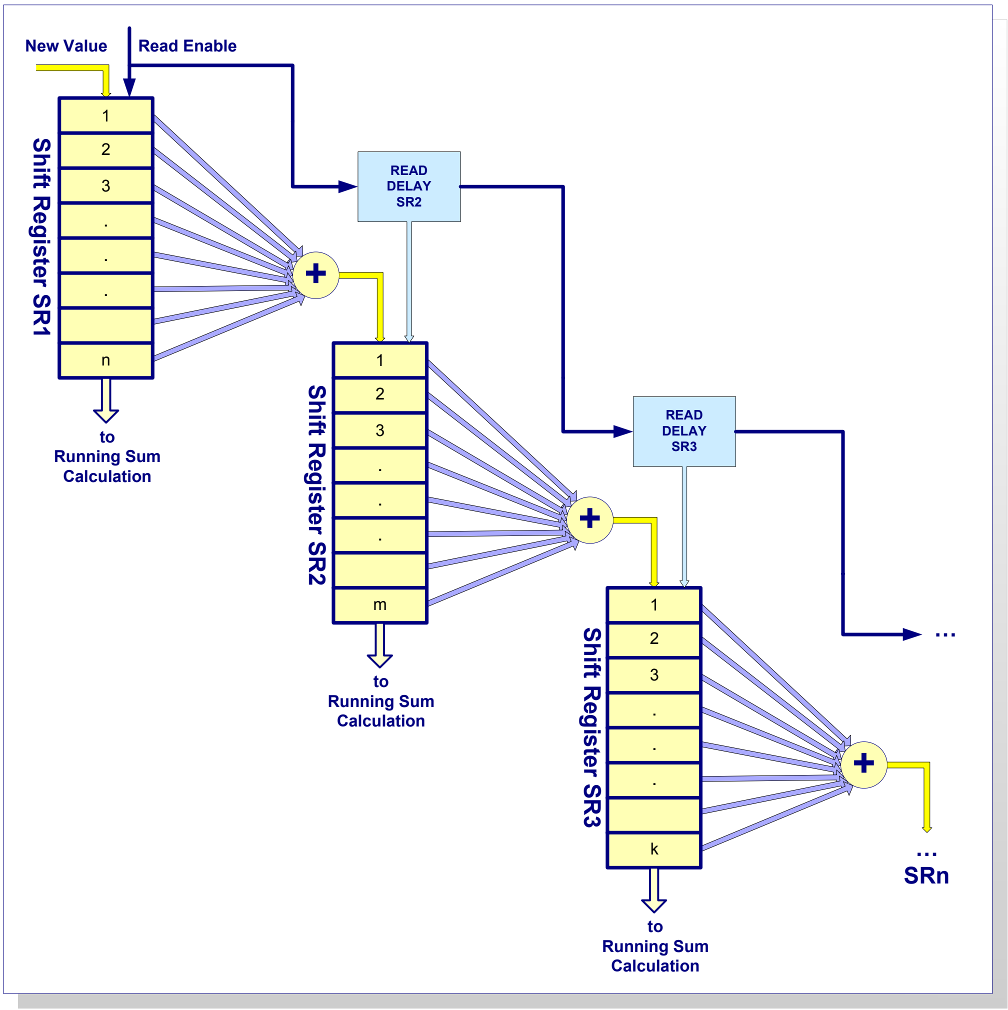


Figure 6-5: Cascading of the Shift Registers in the SRS technique.

Figure 6-5 illustrates this concept. In general, it works by feeding the sum of one Shift Register's contents, every time its contents become completely updated, to the input of another Shift Register. By cascading more of these elements, very long Moving Windows could be constructed with no difficulty, that use significantly smaller memory space.

Furthermore, in the BLMTC system design proposed, the sum of the Shift Register contents is continuously kept and updated in the Running Sums. Thus, the RS output could be also used directly to feed the following stage's input, similarly to the configuration illustrated in Figure 6-4, therefore minimising more the resource utilisation. More about the Shift Registers and their implementation options will be discussed in the following Section 6.4.2.

The optimal achievable latency in the response of each stage in such a system is equal to the refreshing time of the preceding Shift Register. That is, the time needed to completely update its contents. In Figure 6-5, the supervisor circuit, denoted as "Read Delay", making sure that the sum is calculated every time with new values, holds a delay equal to this latency to guarantee the correct operation. Thus, the delay is every time equal to the preceding Shift Register's input clock period multiplied by the elements planned to be used in the sum.

For example (and using the notation of Figure 6-5):

$$SR2_{DELAY} = f_{NewValue} * n \qquad [6\text{-}1]$$

$$SR3_{DELAY} = SR2_{DELAY} * m \qquad [6\text{-}2]$$

Where, $SR2_{DELAY}$, $SR3_{DELAY}$ are the read delays needed for the first and the second Shift Register respectively, $f_{NewValue}$ is the frequency of the input, and the n, m are the number of elements held in each of the Shift Registers.

By cascading just five of these elements, holding only 64 or 128 values each, it is enough to reach the 100-second upper integration limit requested by the specifications. This gained efficiency was necessary for this system to be applicable in a configuration with relatively very low memory availability. In a different configuration of this system, where only Running Sums would be used, the Shift Registers would need to hold approximately 3 million values for each of the 16 detectors to achieve the same approximation error. That means a total of approximately 150MBytes. Instead, by using the Successive Running Sum technique the system is using only some of the FPGA internal memory since it does not need more than 100KBytes.

6.3 Optimal Configuration of the SRS

Given the tolerance acceptable for quench prevention given by the specifications, the quench threshold versus loss time curve has been approximated with a minimum number of steps fulfilling the tolerance. In this way, the number of sliding integration windows has been reduced to twelve.

Furthermore, the latency introduced has little effect to the optimal approximation accuracy though since it varies between them. More specifically, the running sums that span to the low range (fast losses) have zero or very small additional latency. The latency gradually increases as the integration time increases, reaching up to 1.3s for the 21 and 83s time range. In Table 6-2 it is shown the proposed implementation for the BLMTC.

Table 6-2: Successive Running Sums configuration used in BLMTC.

	Moving Windows		**Refreshing**		**Shift Register Name**	**Signal Name**	**bits needed** for each detector	**bits used** for each detector
	40 µs steps	**ms**	**40 µs steps**	**ms**				
Sub-System A	1	0.04	1	0.04		RS00	**18**	**20**
	2	0.08	1	0.04		RS01	**19**	**22**
	8	0.32	1	0.04	SR1	RS02	**21**	**22**
	16	0.64	1	0.04		RS03	**22**	**22**
	64	2.56	2	0.08	SR2	RS04	**24**	**26**
	256	10.24	2	0.08		RS05	**26**	**26**
Sub-System B	2048	81.92	64	2.56	SR3	RS06	**29**	**32**
	8192	327.68	64	2.56		RS07	**31**	**32**
	32768	1310.72	2048	81.92	SR4	RS08	**33**	**36**
	131072	5242.88	2048	81.92		RS09	**35**	**36**
	524288	20971.5	32768	1310.72	SR5	RS10	**37**	**40**
	2097152	83886.1	32768	1310.72		RS11	**39**	**40**

Legend for Table 6-2:

The blue coloured RS (running sum) outputs, i.e. RS01, RS04, RS06, and RS08, represent their additional utilisation as inputs for the adjacent SRs (shift registers), i.e. SR2, SR3, SR4, and SR5 respectively.

6.4 Implementation of the SRS

The procedure for the data processing, which was chosen to be followed, is based on the idea that a constantly updated moving window can be kept by adding to a register the incoming newest value and subtracting its oldest value. The number of values that are kept under the window, or differently, the difference in time between the newest and the oldest value used defines the integration time it represents.

Additionally the implementation will make use of the cascading of multiple moving windows to create longer integration periods in order to minimise the resources utilisation.

6.4.1 Running Sum Calculation

A technique similar in functionality to the one discussed previously, but more efficient in the number of processes that it needs, is shown in Figure 6-6 for the Running Sum calculation. The old value is subtracted from the new value initially and then the output difference is fed to an accumulator. The difference might be, of course, negative and for this reason, the accumulator has a signed input. In addition, the accumulator is built with an enable input, for controlling the function. Its output can be read with a latency of one clock cycle.

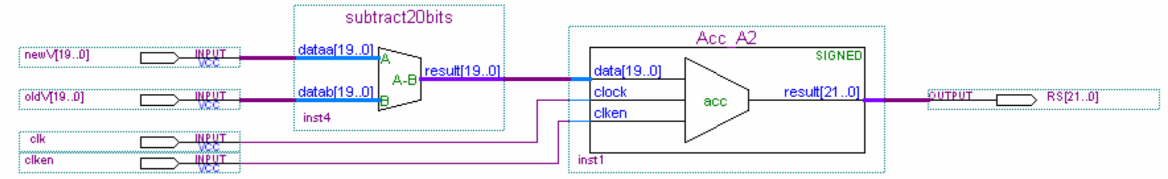


Figure 6-6: Running Sum Calculation.

The width of the accumulator, i.e. the width of its register, is also important. It must be able to hold all possible data without the fear of overflowing and losing data but still should be implemented without any waste of resources. The input of the Running Sum value (shown in Table 6-2) grows in width the later it is in the successive production stage to achieve its maximum accuracy. Thus, accumulators with different number of bits were constructed, for each stage, equally to the maximum width they might have. Those widths can be 20, 22, 26, 32, 36, or 40-bit long.

6.4.2 Production and Maintenance of a Running Sum

As design complexities increase, use of vendor-specific IP blocks has become a common design methodology. Altera provides parameterisable cores, referring to them as Megafunctions. They are optimised for its device architectures that may offer more efficient logic synthesis and device implementation.

The “old value”, that is, the oldest value in the moving window needed to be subtracted from the sum, can be provided with various ways. However, since the target device will be an Altera FPGA then the “altshift_tabs” Megafunction [79] can be used to produce an optimised Shift Register to provide the value delayed by the necessary cycles. It uses the device’s embedded memory blocks for implementing the Shift Register and gives the option to choose any of the available memory block types, that is, in this case either the M4Ks or the M512s. This option will be exploited in order to fit all of the Running Sums in the device by spreading them between all the available resources.

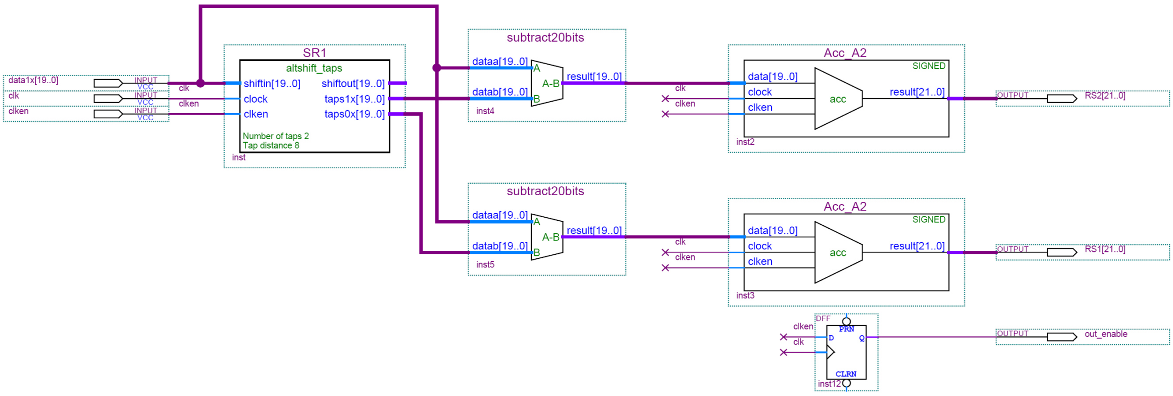
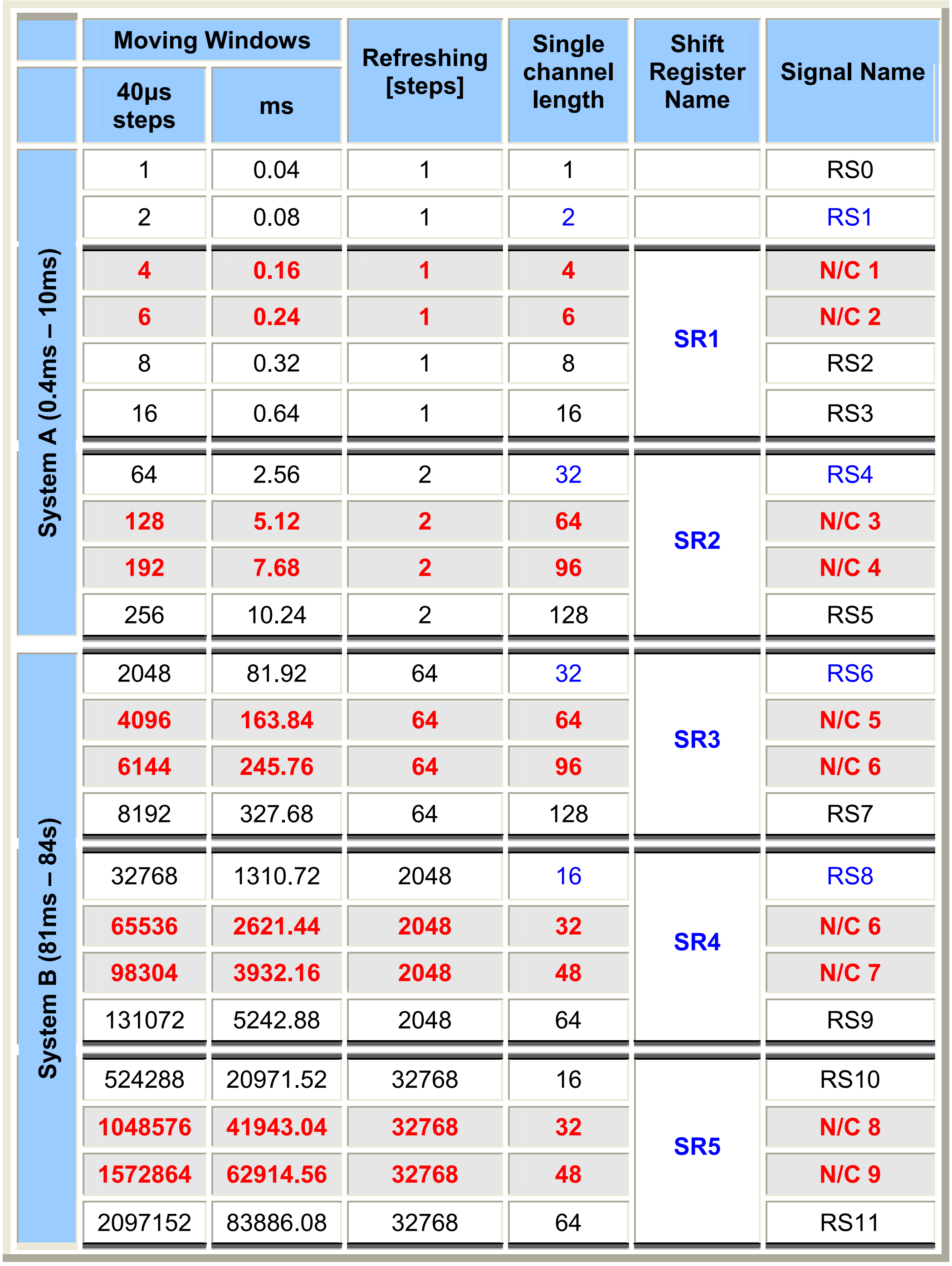


Figure 6-7: Using a Shift Register with Taps for keeping a Running Sum.

This parameterised core can be also configured to give intermediate outputs, usually referred to as taps. The taps provide data outputs from the shift register at certain points in the shift register chain. This feature can be effectively used to combine overlapping memory contents which will provide the most of the reductions. Figure 6-7, illustrates this feature. The Shift Register provides two taps with eight samples distance. That is, two moving windows are created. The first will hold 8 samples and the second 16. Of course, the integration times that those will correspond will be defined by the sample input frequency.

The multipoint output of the Shift Registers combined with the implementation proposed will give as well a possible future advantage in case a different length Running Sum is needed. In Table 6-3 is shown the current configuration of the system including the unused outputs of the Shift Registers (denoted with N/C). If in the future it is found necessary, the nine not used Moving Windows can be exchanged with any of the used ones.

Table 6-3: Successive Running Sum Configuration (including unused outputs).



	Moving Windows		Refreshing [steps]	Single channel length	Shift Register Name	Signal Name
	40µs steps	ms				
System A (0.4ms – 10ms)	1	0.04	1	1		RS0
	2	0.08	1	2		RS1
	4	**0.16**	**1**	**4**	**SR1**	**N/C 1**
	6	**0.24**	**1**	**6**		**N/C 2**
	8	0.32	1	8		RS2
	16	0.64	1	16		RS3
	64	2.56	2	32	**SR2**	RS4
	128	**5.12**	**2**	**64**		**N/C 3**
	192	**7.68**	**2**	**96**		**N/C 4**
	256	10.24	2	128		RS5
System B (81ms – 84s)	2048	81.92	64	32	**SR3**	RS6
	4096	**163.84**	**64**	**64**		**N/C 5**
	6144	**245.76**	**64**	**96**		**N/C 6**
	8192	327.68	64	128		RS7
	32768	1310.72	2048	16	**SR4**	RS8
	65536	**2621.44**	**2048**	**32**		**N/C 6**
	98304	**3932.16**	**2048**	**48**		**N/C 7**
	131072	5242.88	2048	64		RS9
	524288	20971.52	32768	16	**SR5**	RS10
	1048576	**41943.04**	**32768**	**32**		**N/C 8**
	1572864	**62914.56**	**32768**	**48**		**N/C 9**
	2097152	83886.08	32768	64		RS11

6.4.3 OPTIMISATIONS OF THE RUNNING SUMS

The altshift_taps Megafunction is implemented in the device memory blocks. The width and depth of the memory block depends on its parameters. If a longer or wider Shift Register is needed then two or more memory blocks will be combined but no other process can use the memory bits left unused by each Shift Register implemented. Figure 6-8 illustrates an example where the contents for each detector is 32 x 8-bit values. If each detector is treated independently, its Shift Register will occupy one M512 memory block. For the same case, if the data from two detectors are pre-combined, the resource usage will drop to half.

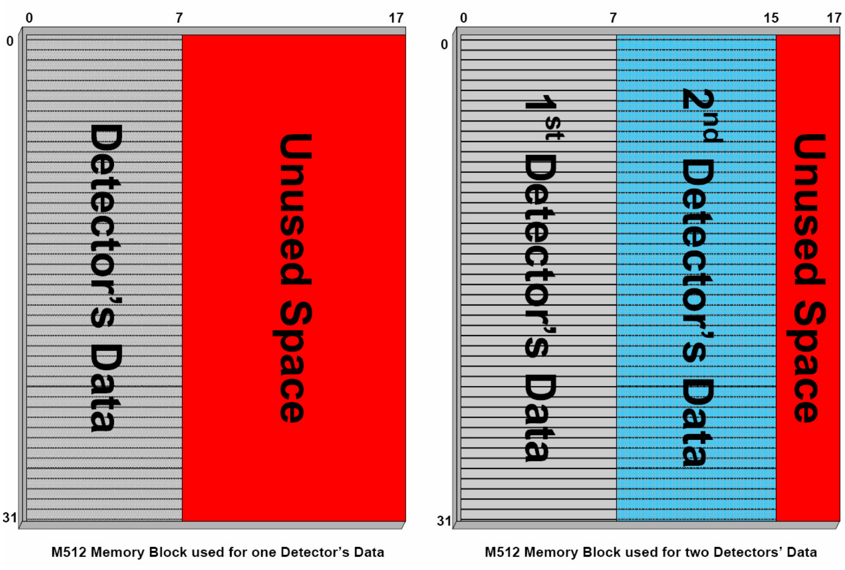
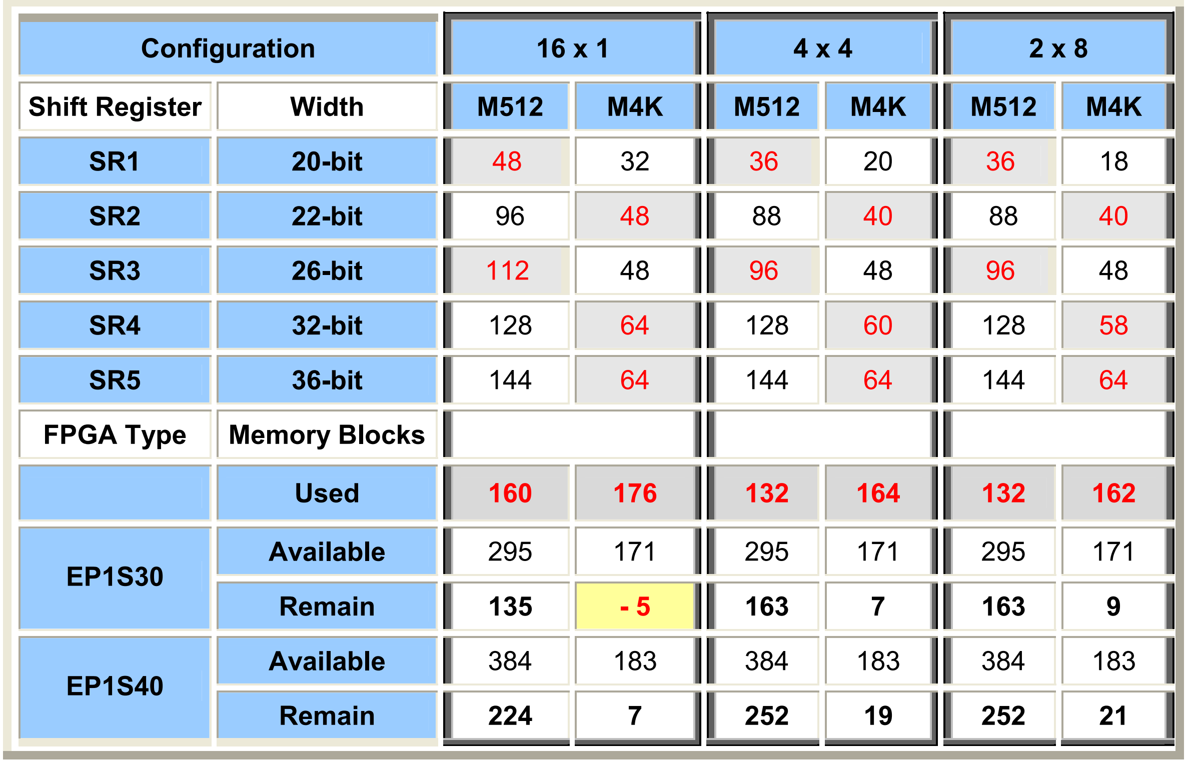


Figure 6-8: Optimisation in Memory Usage by the Shift Register.

Of course, this is not always the case and there is not a generic way to discover such optimisations. Probably this is also the reason why none of the synthesis tools available performs such resource sharing. Thus, it was found necessary an investigation to be made to find the optimal configuration and then it could be forced to be used by the synthesis tool.

In Table 6-4 are shown the results on the investigation of the memory use over different configurations of the five Shift Registers needed to be used for constructing the Successive Running Sums. Three configurations of the 16 detectors were tested; each detector independent, four sets of four detectors combined to share memory resources, and similarly two sets of eight detectors combined.

Table 6-4: Memory Resources Utilisation for different configurations of the Shift Registers.



Configuration		16 x 1		4 x 4		2 x 8	
Shift Register	Width	M512	M4K	M512	M4K	M512	M4K
SR1	20-bit	48	32	36	20	36	18
SR2	22-bit	96	48	88	40	88	40
SR3	26-bit	112	48	96	48	96	48
SR4	32-bit	128	64	128	60	128	58
SR5	36-bit	144	64	144	64	144	64
FPGA Type	Memory Blocks						
	Used	**160**	**176**	**132**	**164**	**132**	**162**
EP1S30	Available	295	171	295	171	295	171
	Remain	**135**	**- 5**	**163**	**7**	**163**	**9**
EP1S40	Available	384	183	384	183	384	183
	Remain	**224**	**7**	**252**	**19**	**252**	**21**

The results are not so dramatic as the previous example since the first optimisation has already been done by carefully choosing the Running Sums' lengths to match the allowed configurations of the memories. In Table 6-3, it was seen that the channel lengths do not exceed 128, are always multiples of 8, and that all output points in a Shift Register have an even spacing. That is, all the parameters for achieving the most efficient results have been respected.

Nevertheless, the gain in the resources, by using any of the two combined forms instead of treating each detector independently, is of great importance since it allows the system to fit in all used devices. It can be seen in Table 6-4 that the first configuration falls short by five M4K memory blocks, if the EP1S30 chip from the Stratix family is used.

Including the above considerations and choosing to combine four detectors' data throughout the system, the Running Sum calculations have been transformed similar to the one shown in Figure 6-9. This configuration calculates and outputs two Running Sums for each of the four detector data inputs (inputs: *Data_xx[19..0]*). The output data (outputs: *RS_xxx[25.0]*) are available every time its output enable flag (output: *out_enable*) is asserted.

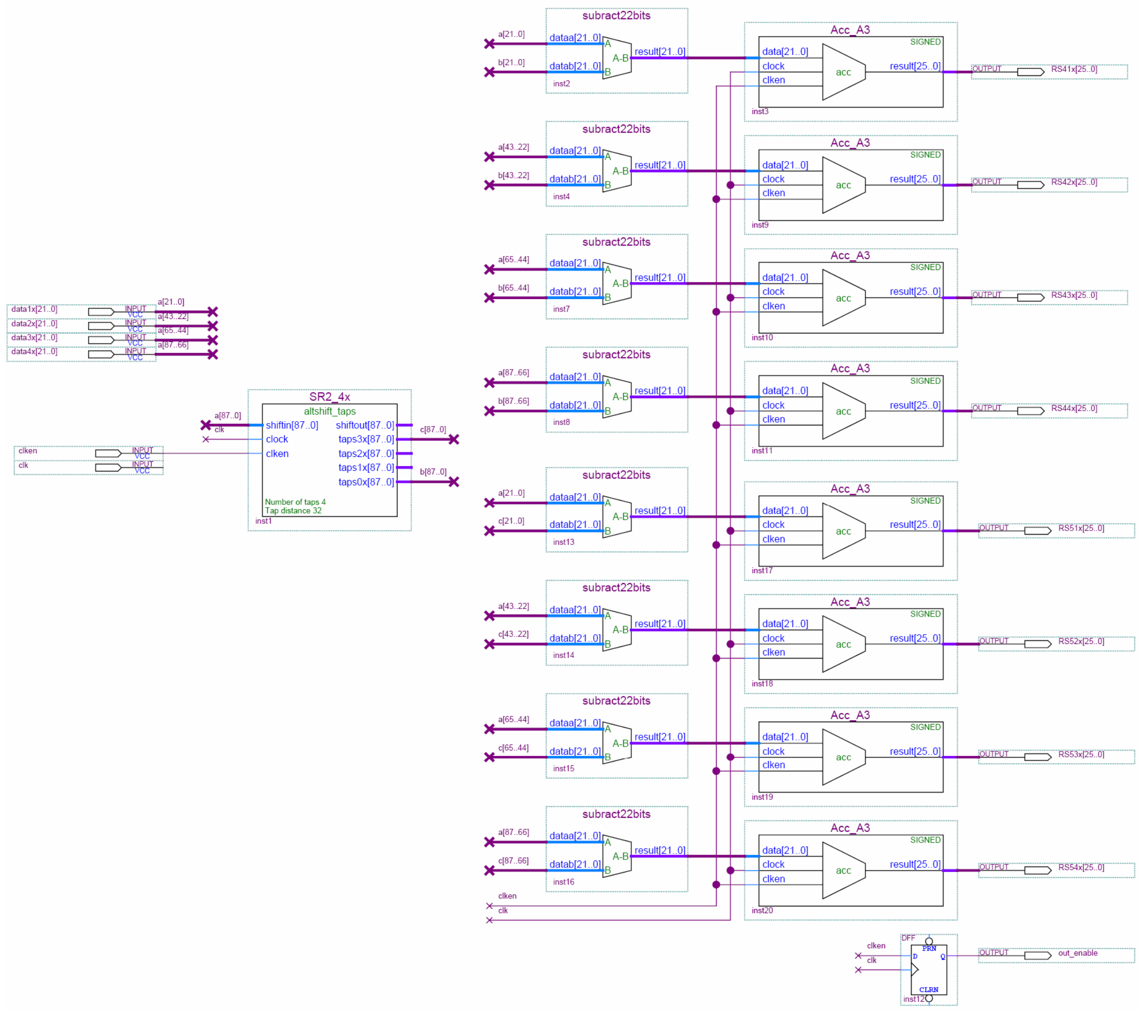


Figure 6-9: Production of two Running Sums for four Detectors with Memory Sharing.

6.4.4 SUCCESSIVE RUNNING SUMS IN THE BLMTC

This optimised block was consecutively used, with specific Shift Register settings for each of the five needed, to construct the two sub-systems shown in Figure 6-10 and in Figure 6-11. The first provides the Successive Running Sums spanning the time region between 40μs to 10ms and the second between 81ms to 84s.

A chain of delays, that allows the correct values to be loaded at each stage, is constructed using counters that are triggered by the output enables of each stage. The whole process is triggered each time new data arrives from the RCC process as shown previously in Chapter 4.

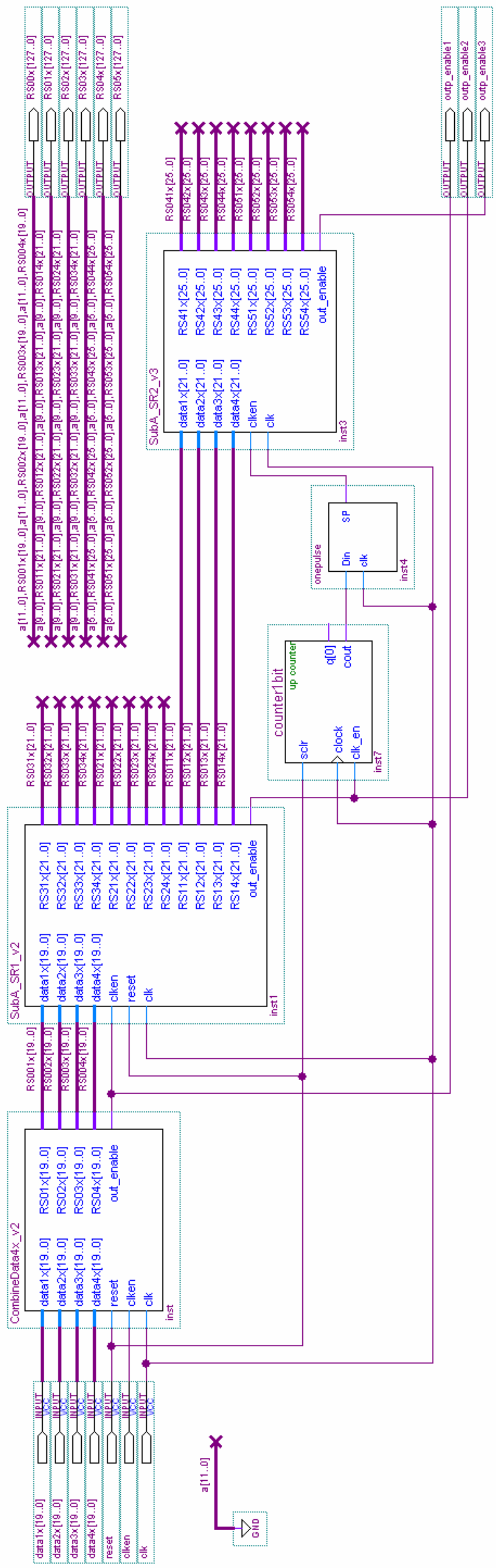


Figure 6-10: Successive Running Sums from 40µs to 10ms (for four detectors).

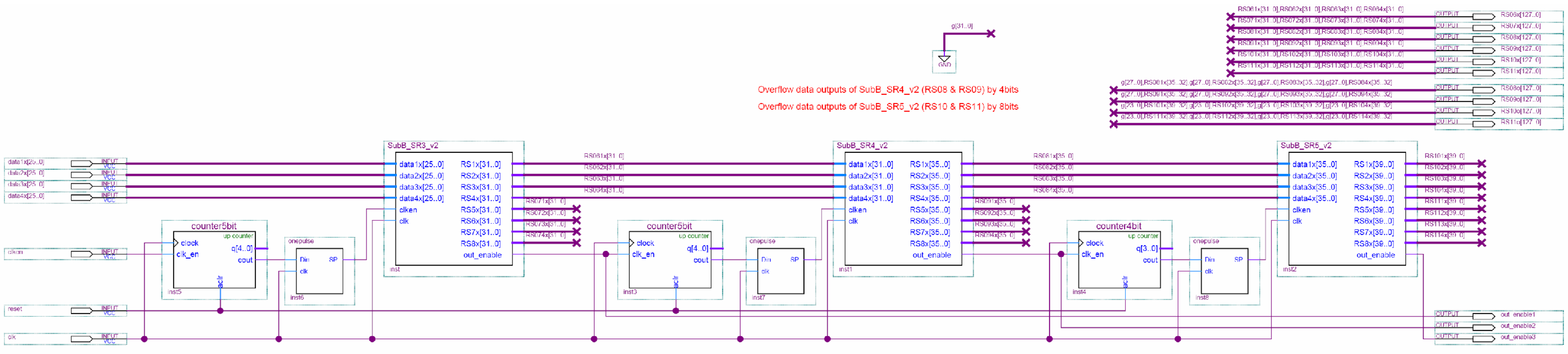


Figure 6-11: Successive Running Sums from 81ms to 84sec (for four detectors).

In the top level, of the BLMTC's hierarchical design, four of each of those sub-systems have been added to provide the data processing needs for the 16 Detectors. This is illustrated in Figure 6-12. The outputs of each block include six read enable flags, one for each data output group. Later stages in the design, like the Logging and the Threshold Comparator that receive the SRS output data, can use those flags not only to receive correctly the data but also to multiplex in time the outputs and thus use resources more efficiently.

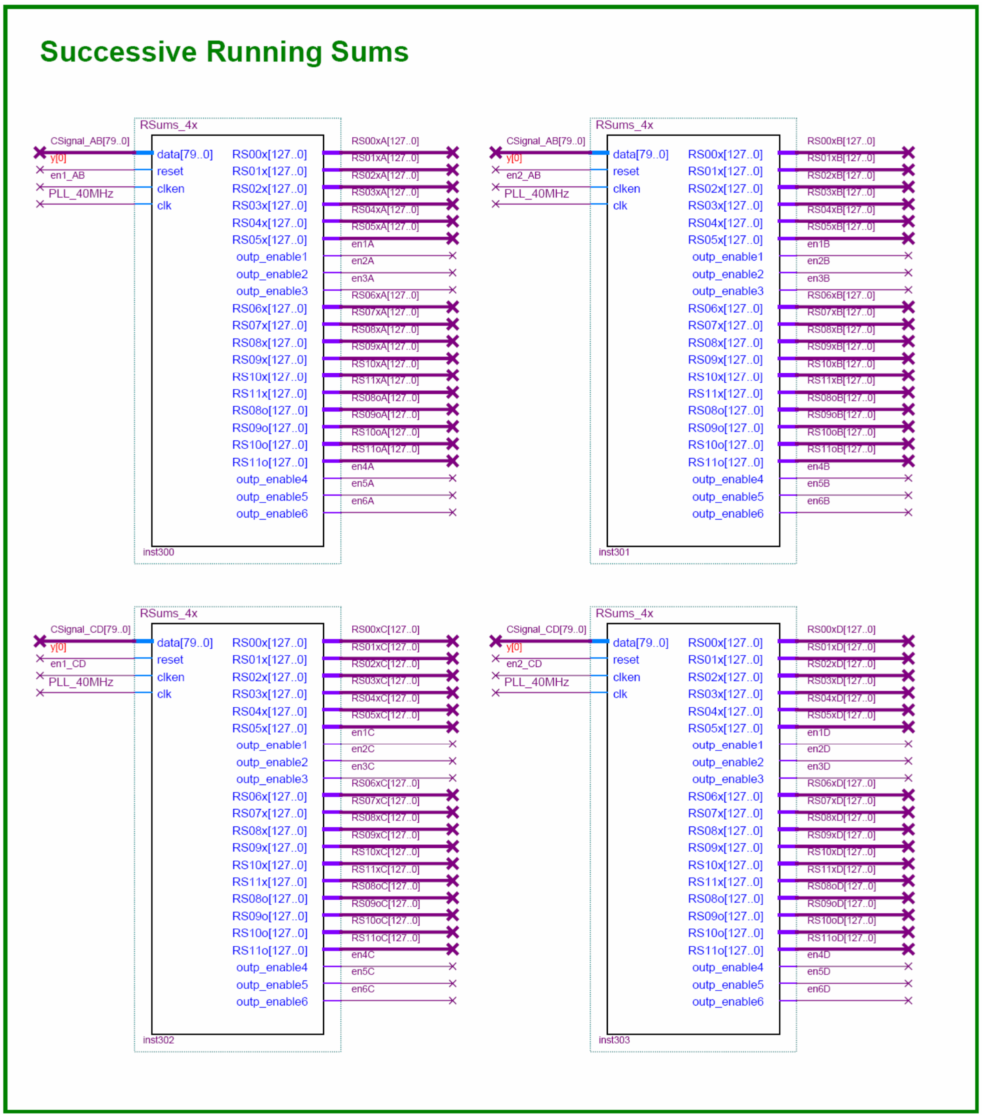
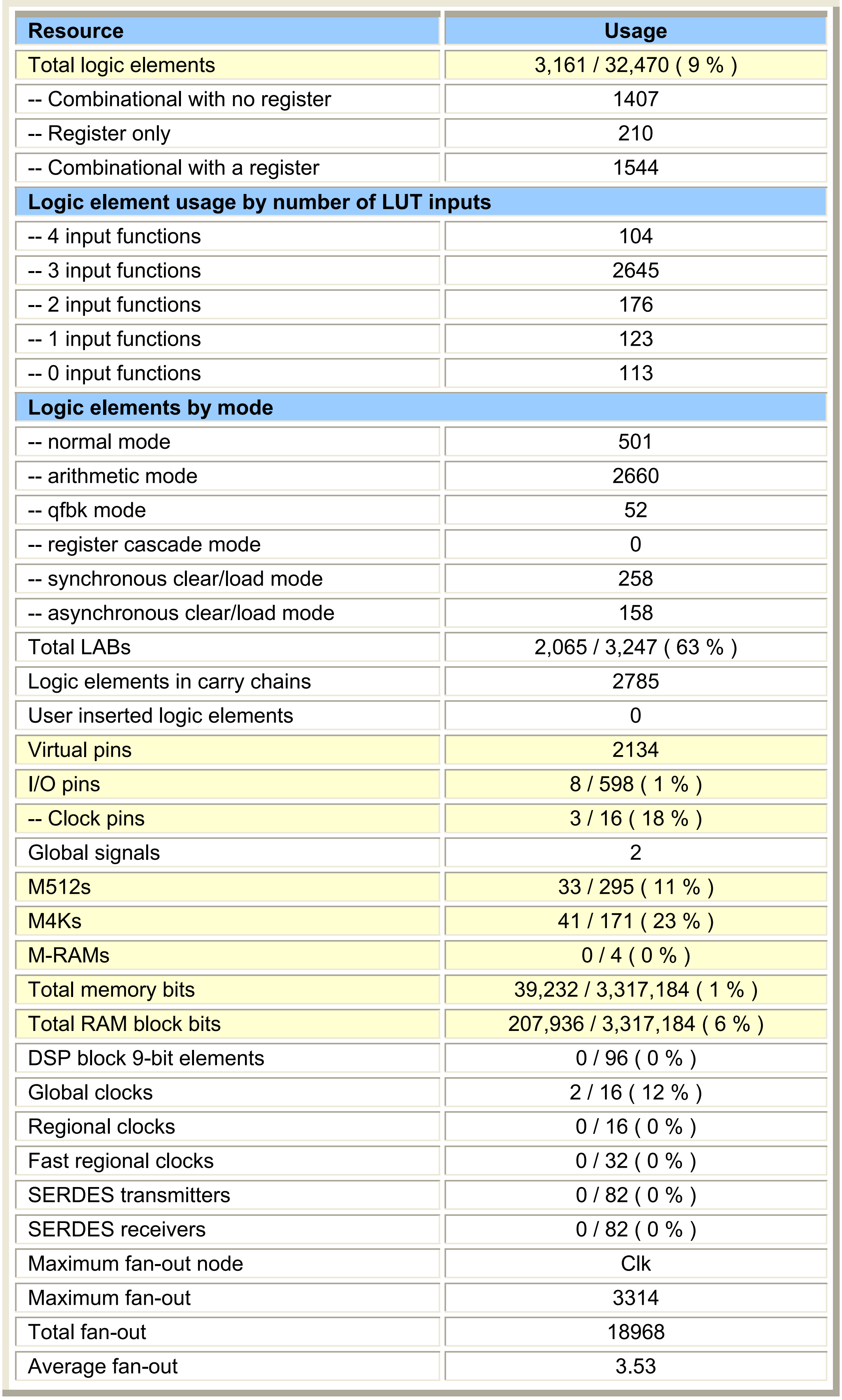


Figure 6-12: Successive Running Sums in the BLMTC.

The resource usage for the data processing technique proposed makes use of 39% of the Logic Elements in the EP1S30 device or 30% of the EP1S40, the second option for the BLMTC system. The complete resource usage, calculated by the Quartus II after fitting, for one of the SRS blocks that treat the data from four detectors can be seen in Table 6-5.

Table 6-5: Resource Usage by the Successive Running Sums for four Detectors.



Resource	Usage
Total logic elements	3,161 / 32,470 (9 %)
-- Combinational with no register	1407
-- Register only	210
-- Combinational with a register	1544
Logic element usage by number of LUT inputs	
-- 4 input functions	104
-- 3 input functions	2645
-- 2 input functions	176
-- 1 input functions	123
-- 0 input functions	113
Logic elements by mode	
-- normal mode	501
-- arithmetic mode	2660
-- qfbk mode	52
-- register cascade mode	0
-- synchronous clear/load mode	258
-- asynchronous clear/load mode	158
Total LABs	2,065 / 3,247 (63 %)
Logic elements in carry chains	2785
User inserted logic elements	0
Virtual pins	2134
I/O pins	8 / 598 (1 %)
-- Clock pins	3 / 16 (18 %)
Global signals	2
M512s	33 / 295 (11 %)
M4Ks	41 / 171 (23 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	39,232 / 3,317,184 (1 %)
Total RAM block bits	207,936 / 3,317,184 (6 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	2 / 16 (12 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	Clk
Maximum fan-out	3314
Total fan-out	18968
Average fan-out	3.53

6.5 Static Timing Simulation of the SRS Process

It was shown that the implementation of the SRS process has been done using IP cores and standard logic blocks in order to achieve its fitting to the available device. Even though this approach gives a resulting code that is bound to a specific manufacturer, in this case for the Altera FPGAs, the immense advantage comes at the verification process. The cores are optimised for speed and area usage, and have a well-defined behaviour, which are preserved during the logic synthesis and place and routing (P&R) compilations by the EDA tools.

The exception comes only when the designer wishes to push the core to work at its maximum operating frequency and beyond. In all other cases, where the circuit will be driven at a fraction of its maximum speed, like in the BLMTC, the simulation is more applied to discover wrong connections or incorrect time interfacing of the components, errors commonly referred as programming bugs, than verifying the timing analysis.

Nevertheless, back-annotated static timing simulation has been done with the help of the Quartus Simulator as the first verification tool, since it was used for all the other implemented processes. More about this tool and the options used can be found in Section 2.8. Parts of the simulation waveforms outputted can be seen in Figure 6-13. The inputs used simulate the behaviour of the process for slow losses. The one count value received is coming from the change of state of the ADC and correctly is not taken into account.

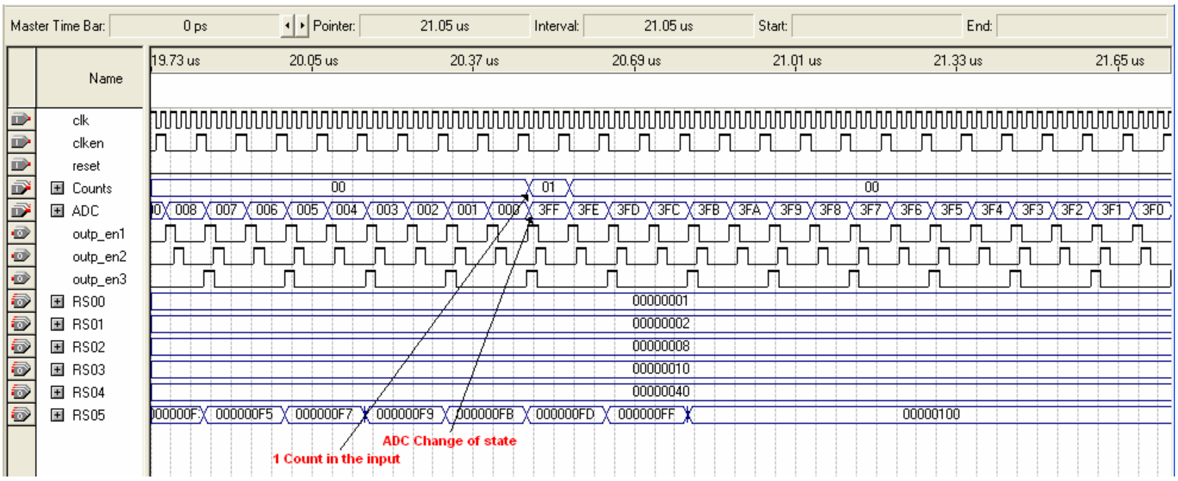

Master Time Bar: 0 ps
Pointer: 21.05 us
Interval: 21.05 us
Start:
End:

Figure 6-13: Simulation / Sub-SystemA’s Running Sum (RS) outputs (slow losses).

One more part of the simulation is shown in Figure 6-14, which in this case exhibits a higher number of input counts. There are six counts received together with a change of ADC state. Thus, the output correctly calculates 5 counts x 1024 fractions/count = 5120 or equally in hexadecimal 1400 fractions.

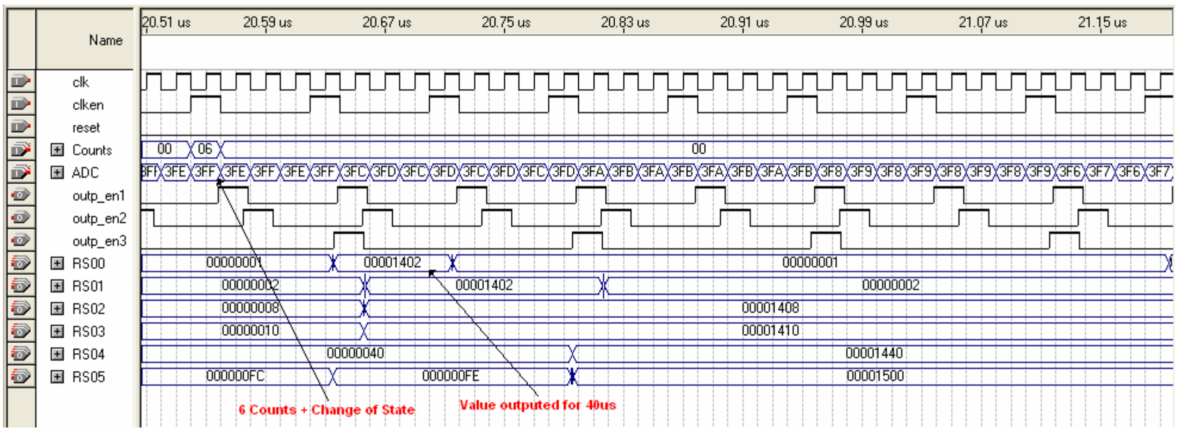


Figure 6-14: Simulation / Sub-SystemA's Running Sum (RS) outputs (fast losses).

6.6 Hardware Verification of the SRS Process

In order to diminish the ambiguity of the comparison of the SRS outputs with the expected values, when real and thus random data were given to the input, two test-benches were created. In order to approximate as close as possible the real system, the CFC card was used in both of them to provide the input to the BLMTC system.

In the first configuration, the difference was that the CFC card's FPGA firmware was not reading the analogue inputs but had fixed values in the counters. Thus, the readout was constant and more importantly known.

The second test-bench initially has been used to check the analogue parts of the CFC card. The detector signal has been replaced by a well-specified current source that could span, with pA accuracy, the complete dynamic range. Thus, by changing the value of the current in the input and having known the conversion ratio the expected outputs could be defined. The same configuration was finally used to check also the linearity of the BLM system produced and some more information on the configuration and the results can be seen in Appendix D.

6.7 SUMMARY

The strategy for machine protection and quench prevention is based on the BLM system. The processing involves a proper analysis of the signal pattern in time and a proper account of the energy of the beam. In this chapter, a highly efficient strategy for the data processing of the BLM system has been proposed to match the available electronics.

The procedure for the data processing, proposed, is based on the Running Sum idea. That is, a constantly updated moving window kept by adding to a register the incoming newest value and subtracting its oldest value. The old value is delayed by a number of cycles with the help of a shift register. The number of values that are kept under the window, or differently, the difference in time between the newest and the oldest value used in this window defines the integration time it represents.

In order to increase the efficiency in resources, it is making use of the parameterised multipoint shift register core available for the Stratix devices and to achieve the best optimisation three configurations for the 16 channels under treatment were tested.

A further optimisation in memory usage needs has been achieved with the Successive Running Sum technique that allowed the data processing to use only the FPGA's embedded memory. It is able to provide 12 moving window for each of the 16 detectors that span from 40μs to 84s. With this technique, the system is able to reach such long integration periods with relatively small length shift registers by making use of the already calculated sums in order to calculate longer running sums. The expense is some additional latency, which in this case found to be negligible since it is every time only a fraction if compared to the integration time of its shift register.

The combination of the multipoint shift register and the SRS scheme used give an additional advantage, if the integration times chosen need to be changed. The procedure is fairly simple and if the not connected outputs from the multipoint shift registers are used, this implementation already provides nine more integration times to choose from.

REFERENCES

[78] CERN Engineering Specification, *"On the Measurement of the Beam Losses in the LHC Rings"*, LHC-BLM-ES-0001 Rev 2.0, CERN, Geneva 13 Jan 2004

[79] Altera Corp. *"altshift_taps Megafunction User Guide"*, Version 1.0, September 2004, [online: http://www.altera.com/literature/ug/ug_alt_shift_taps.pdf]

[80] Gianluca Guaglio, "Comparison of the Running Sums with a 2nd order Filter", BL Section internal note, unpublished.

07

Threshold Comparator & Channel Masking

Chapter 7. Threshold Comparator & Channel Masking

The quench and damage levels are time and energy dependent [81] [82]. For this reason, from each detector's data, as it was shown, this system calculates and provides 12 Running Sums. Every Running Sum, after every new calculation, needs to be compared with its corresponding threshold value that was chosen by the beam energy reading given that moment. If the level is found to be higher, the comparator will initiate the necessary dump request.

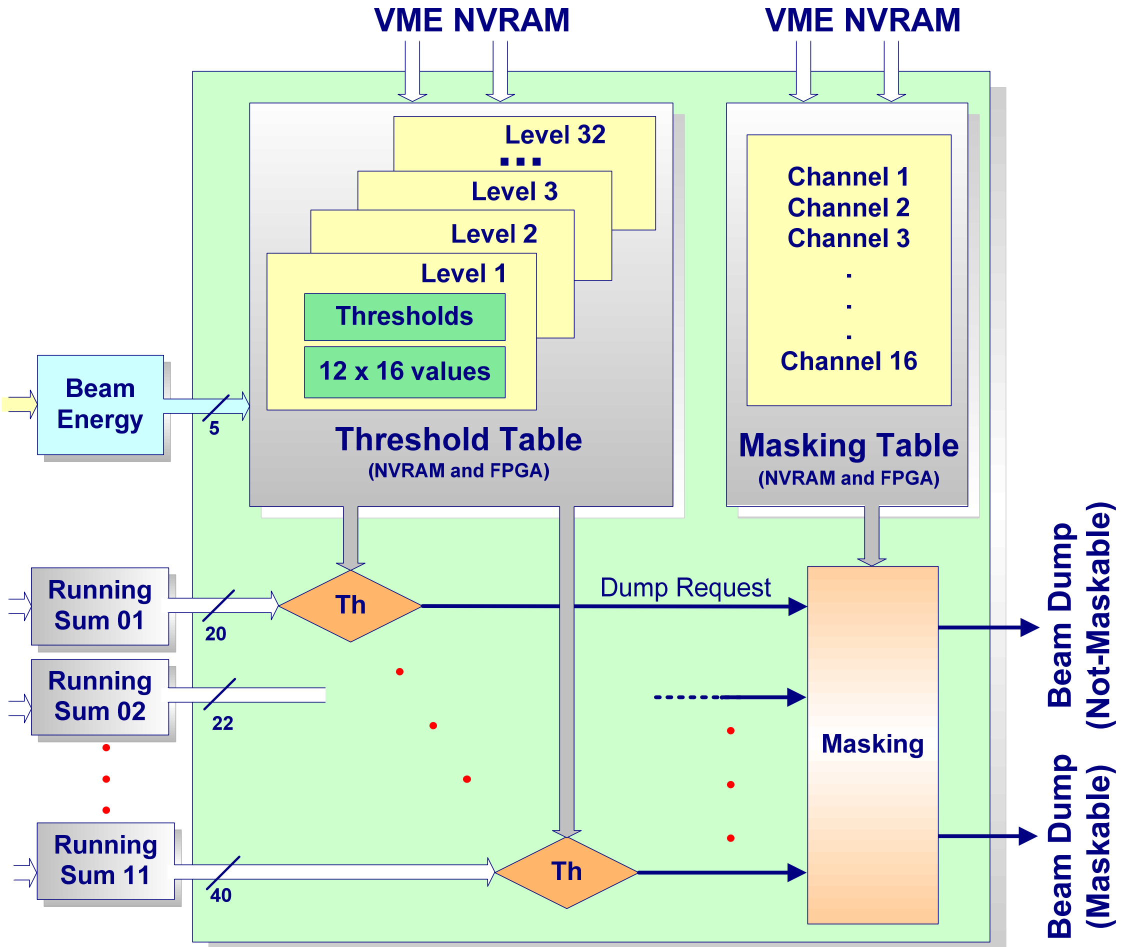


Figure 7-1: Threshold Comparator (TC) and Masking Processes Block Diagram.

For the reasons to be explained, the Threshold Level Comparator and the Channel Masking is proposed to be implemented having unique tables on each card that can be loaded with various ways in the FPGA embedded memory blocks.

All dump requests will initially be gathered by a Masking process with the main purpose of distinguishing between “Maskable” and “Not-Maskable” channels. The Control Room will have the ability to inhibit some of the used channels, i.e. the Maskable, under specific and strict conditions. At the same time, highly critical channels will not be possible under any circumstance to be disabled.

Finally, all Maskable and Not-Maskable outputs will be collected and summed individually by the Combiner card, which consequently will forward them to the Beam Interlock System (BIS) [83].

A block diagram of the proposed functionality is illustrated in Figure 7-1 and in the following parts of this chapter an insight to the proposed implementation of the TC and the Masking processes will be given, as well as an effective strategy for the Threshold and Masking Tables.

7.1 THRESHOLD LEVEL COMPARATOR (TC)

Given the facts that the LHC tunnel is not uniformly constructed, there are Arcs, Dispersion Suppressors, Long Straight sections, Collimation areas, etc, and the detector "sees" more or less particles depending on its position relative to the magnet, different threshold values for each of the approximately 4000 ionisation chambers need to be set. In addition, the threshold value is inversely proportional to the beam energy. The beam energy will not be constant, but it is foreseen to be a ramp from 450GeV to 7TeV lasting approximately 30 minutes.

Thus, the table consisting of those data could be considered as a 3-dimensional table showing the number of protons lost over time, over beam energy, and over ionisation chamber.

7.1.1 THRESHOLD TABLE

The consideration of having a global database of different Threshold and Warning values stored on each card would be highly inefficient. It would actually mean different thresholds for each of the approximately 4000 ionisation chambers, each of them providing 12 Moving Windows, and for all possible 65,536 energy levels, that is, from a 16-bit beam energy reading. This global table would require a significant large memory space reserved on each of the cards to store those values, which exceed the 3 billion values.

In order to minimise that table and the memory needed to be stored, it is proposed to spread the load between them. Thus, a unique block of values will be created for each of the cards of the complete system. The information included will be still for its moving windows, but for less beam energy levels and the specific Ionisation Chambers, each BLMTC card is reading.

More specifically, on each card it was shown that 12 Running Sums for each of the 16 Detectors will be calculated. It is proposed to scale the energy into 32 Beam Energy Levels (0.45 to 7 TeV) and hold information only for those 16 detectors connected. That would give a total of 6,144 Threshold values needed to be held on each card. Table 7-1 shows an example of such Threshold table for one of the 16 detectors by using these parameters.

Table 7-1: Threshold Table Example for one Detector.

Beam Energy	RS00	RS01	…	RS10	RS11
0	Value #1	Value #2	…	Value #11	Value #12
1	Value #13	Value #14	…	Value #23	Value #24
2	Value #25	Value #26	…	Value #35	Value #36
…	…	…	…	…	…
31	Value #373	Value #374	…	Value #383	Value #384

At the same time, the Warning levels could be calculated and transmitted to the Control Room by the PowerPC situated in the VME crate. The Warning level will correspond to a fraction of the Threshold value at that moment, provisionally decided to be 30%. The BLMTC system will provide to the CPU the calculated Running Sums together with their corresponding threshold values through the Logging process to simplify the actions need to be done by the PowerPC. The Logging process is discussed in the following Chapter 8. Moreover, the refresh rate of that system is quite slow compared with the change rate in the BLM system, thus, for the very fast losses no Warnings will be necessary to be checked or transmitted to the Control Room.

Thus, the advantages emerging from this strategy can be seen to be straightforward. The unique table for each of the monitors can thoroughly be prepared and checked before it is uploaded. It can be quickly and easily upgraded for specific systems when the levels need to be changed. It can be used as a calibration and offset tool. It is small enough to be kept internally by the Stratix device, in the embedded memory blocks available. That will decrease the access time and the implementation simplicity, if compared to an external memory device.

For added security not only the table data will have to be loaded manually, and not sent over some network, but in addition when the data are needed to be updated this will be possible to be done only when there is no operation. (i.e. no circulating beam).

7.2 CHANNEL MASKING

The Masking function will serve two main purposes. The first one will be to deactivate all of the unconnected spare channels. It is foreseen, to include two spare channels per CFC card all along the Long Straight Sections (LSS) of the tunnel, for the case more detectors will be necessary to be connected. Additionally, for the rest of the tunnel sections, more or less channels will be used, from the eight available at each card, depending on the topology. The second purpose of the masking function will be to distinguish between the Maskable and the Not-Maskable families of detectors. Table 7-2 shows this distinction for each family.

Table 7-2: Functional families of Beam Loss Monitors.

Type	Area of use	Maskable	Time resolution
BLMC	Collimation sections	No	1 turn
BLMS	Critical aperture limits or critical positions	No	1 turn
BLMA	All along the rings	Yes	2.5 msec
BLMB	Primary collimators	Yes	1 turn / bunch-by-bunch

It should be noted that the whole BLM System is a critical system and must be 100% operational to allow beam injection and that the beam is dumped if it fails. Nevertheless, the distinction is done at this level and the Beam Interlock Controller will have the ability to allow beam dumps coming from the Maskable detectors when a “safe” beam is circulating. That is, beam with low intensity, energy and possibly other parameters, which cannot result in damaging of the equipment.

In general, the Beam Interlock System permits beam operation in the LHC when all User Systems deliver the USER_PERMIT to the Beam Interlock System. To allow for some flexibility while maintaining safety, the LHC systems connected to the Beam Interlock System are classified in two families: MASKABLE and NOT MASKABLE.

MASKABLE signals: The signal from the connected system that has been defined as MASKABLE, could be temporarily ignored if the beam is safe. If the mask has been set by the operator, the USER_PERMIT is not taken into account to produce the BEAM_PERMIT.

NOT MASKABLE signals: if the User System is NOT MASKABLE, the USER_PERMIT will never be ignored, and has a direct influence on the BEAM_PERMIT status. [83]

In the BLM system, detectors which fall into both categories might be treated in the same BLMTC card. Thus, for uniformity all BLMTC cards provide two USER_PERMIT flags, one for the MASKABLE and the other for the NOT MASKABLE.

7.2.1 MASKING TABLE

A similar strategy to the one used for the Threshold Table is proposed also for the Masking table. Even though the amount of information needed for a global table here is much less, it is advised simply for preserving uniformity throughout the design. Thus, unique tables will need also to be produced to hold only the necessary information specific to the channels the card is processing. Two pieces of information will be available, for each of the 16 channels the card is processing. Those will be the detector connection status and in which type of families it belongs, i.e. if it is Maskable or not. Table 7-3 shows an example of a Masking table needed to be stored in each of the BLMTC cards.

Table 7-3: Example of information stored in the BLMTC card's Masking Table.

Detector	Connected	Maskable
1	No	Yes
2	No	Yes
3	Yes	No
…	…	…
15	Yes	No
16	Yes	Yes

7.3 IMPLEMENTATION OF THE TC PROCESS

The TC process will have the main task of comparing the calculated Running Sum values with their corresponding Threshold values for the given Beam Energy. Nevertheless, since there will be needed 192 comparisons to take place at each acquisition, some of them 32 and others 64-bit long, its implementation needs to use a highly efficient strategy.

7.3.1 OPTIMISATION OF THE TC PROCESS

The resource usage can be significantly reduced by resource sharing if the specific implementation of the SRS process, shown in Chapter 6, and the time difference in the arrival of the Running Sum values is taken advantage. It was shown, that the Running Sums are processed by quartets in order to optimise the memory use. Moreover, those combined Running Sum values for the same integration time arrive, from the four SRS processes to the TC process, one after the other with a clock cycle difference.

This behaviour can be used by the TC process to allow its implementation to have only the resources needed for one of them and share those resources between them, since never two of them will need those at the same time. This time-multiplexing scheme will need to be repeated once for each of the 12 integration times chosen for this system to calculate for each detector.

The TC process was partitioned for simplicity in six parts and each compares two Running Sums from each of the 16 detectors with their threshold values, time-multiplexed by sets of four. This partitioning has additionally been done in 6 and not 12 parts, which would be equal to the number of Running Sums, to utilise efficiently one more functionality of the SRS process. The decoding and decision about which of the four SRS processes connected should use the comparator resources can be also shared between the two Running Sums coming in parallel and at the same point in time from the same Shift Register. Figure 7-2 shows the first part of the TC process that is able to output the result of the comparisons for the two first Running Sums coming from all the channels used in the card.

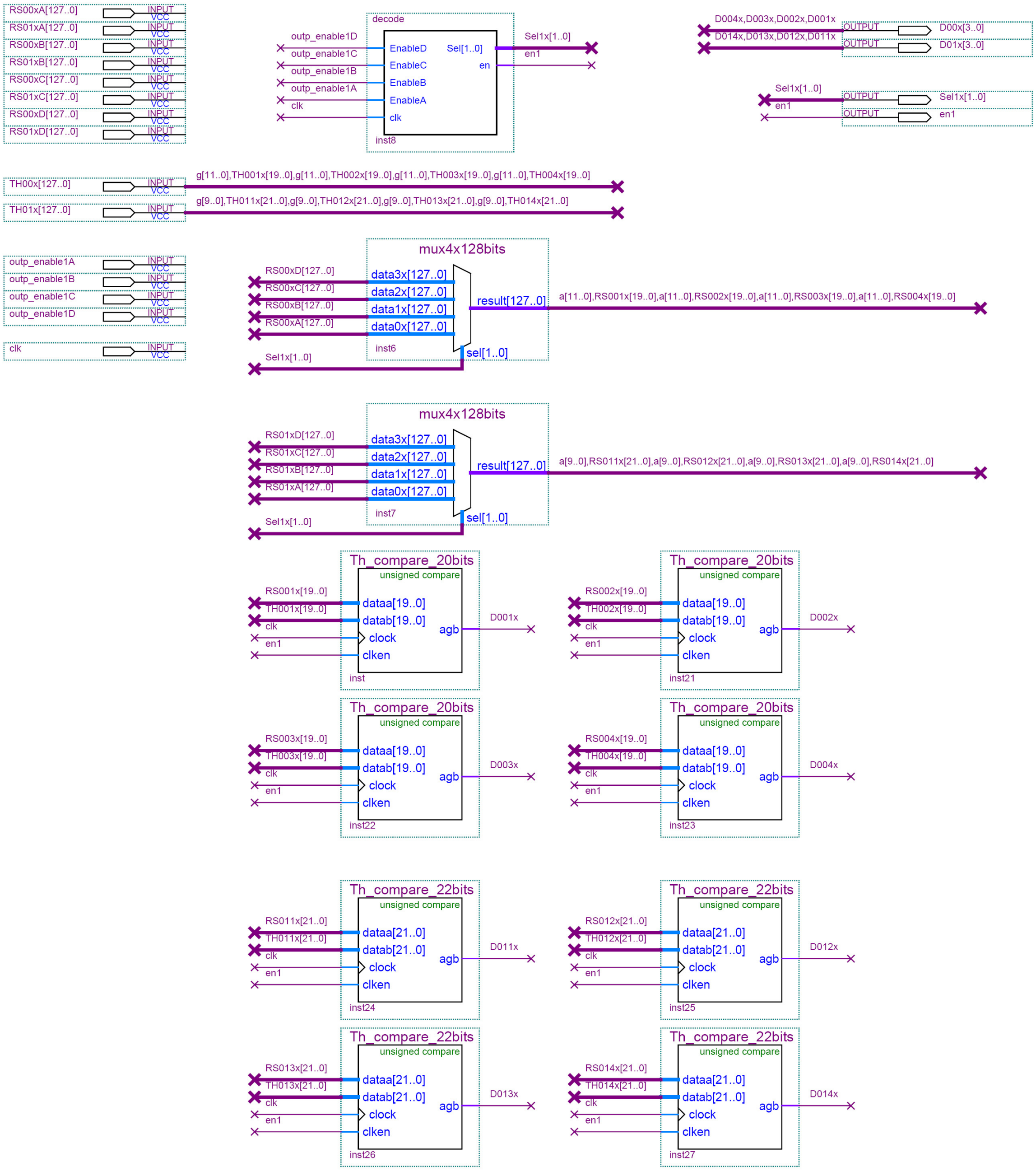
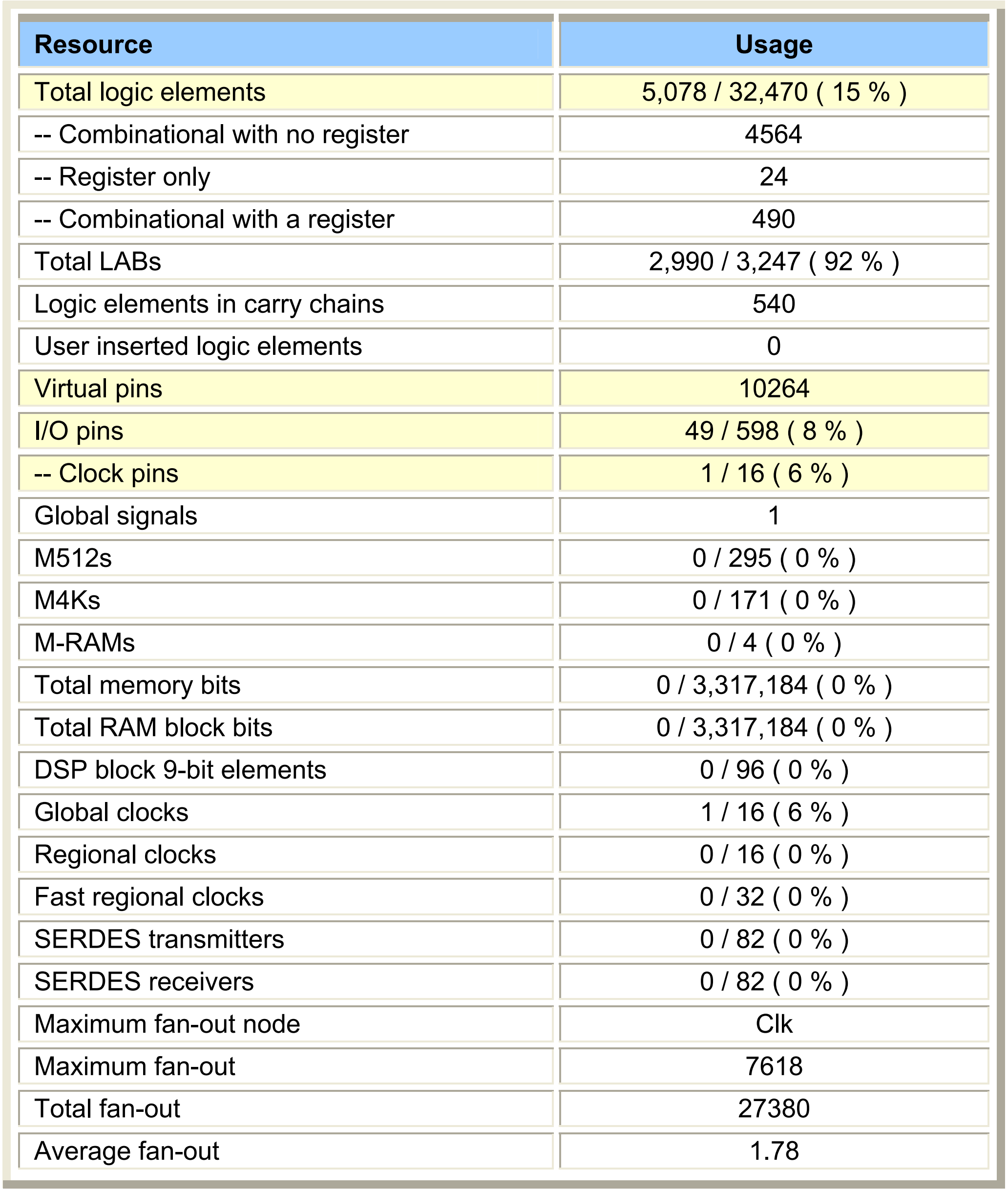


Figure 7-2: TC Process Part (Comparing two Running Sums from 16 Detectors).

7.3.2 RESOURCE UTILISATION BY THE TC PROCESS

The resource usage for the complete TC process is approximately 15% of the Logic Elements available in the EP1S30 device. The efficiency achieved has a great impact on the BLMTC's fitting, since it reduced the usage by approximately 75%, by using the time-multiplexing scheme. The resource usage report for the complete TC process needed for the BLMTC system, calculated by the Quartus II Fitter tool, can be seen in Table 7-4.

Table 7-4: Resource Utilisation by the TC Process.



Resource	Usage
Total logic elements	5,078 / 32,470 (15 %)
-- Combinational with no register	4564
-- Register only	24
-- Combinational with a register	490
Total LABs	2,990 / 3,247 (92 %)
Logic elements in carry chains	540
User inserted logic elements	0
Virtual pins	10264
I/O pins	49 / 598 (8 %)
-- Clock pins	1 / 16 (6 %)
Global signals	1
M512s	0 / 295 (0 %)
M4Ks	0 / 171 (0 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	0 / 3,317,184 (0 %)
Total RAM block bits	0 / 3,317,184 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	1 / 16 (6 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	Clk
Maximum fan-out	7618
Total fan-out	27380
Average fan-out	1.78

7.4 IMPLEMENTATION OF THE THRESHOLD TABLE

The table will no longer be universal, but specific to the card. The data will be loaded to the mezzanine card's NVRAM through an external dedicated connector. From there it will be possible to change the internal RAM of the FPGA by setting an internal flag signifying that there is an update available. The update procedure will be executed then when the Beam Permit status allows this.

7.4.1 MEMORY REQUIREMENTS FOR THE THRESHOLD TABLE

It has been calculated that 6,144 Threshold values are needed to be held on each card, if the beam energy is divided in 32 levels. The 12 Running Sum Values for each detector, as it was shown, will have different widths that span from 20 to 40bits depending on the creation stage they are coming from in the Successive Running Sum technique. The optimal implementation of the table would use equal widths also for the Threshold values. In Table 7-5, where the memory usage calculation for this case is done, it is shown that the total memory needed is around 22KB.

Table 7-5: Optimal Memory Utilisation by the Threshold Table.

Width (Bits)	Detectors	Energy Levels	Running Sums	Values	Total (Bits)
20	16	32	1	512	10,240
22	16	32	3	1,536	33,792
26	16	32	2	1,024	26,624
32	16	32	2	1,024	32,768
36	16	32	2	1,024	36,864
40	16	32	2	1,024	40,960
		Total	**12**	**6,144**	**181,248**

Nevertheless, for reasons that go deep into the memory construction, each memory block's data in order to be accessed faster and more efficiently have to done by values divisible by bytes. Thus, for the implementation of the table, two widths for the Threshold values will need to be used, a 32-bit and a 64-bit. Table 7-6 summarises the actual memory usage for that table when those widths are used. It can be seen that the total amount needed has been increased to 32KB.

Table 7-6: Actual Memory Utilisation by the Threshold Table.

Width (Bits)	Detectors	Energy Levels	Running Sums	Values	Total (Bits)
32	16	32	8	4,096	131,072
64	16	32	4	2,048	131,072
		Total	**12**	**6,144**	**262,144**

7.4.2 THE TH-TABLE FUNCTION

The 32 KB of data consisting of the Threshold values can be uploaded to each system independently whenever a change or update of a card is needed through the front-panel connector provided by the mezzanine card.

For the initial deployment on all cards, a boundary scan tool can be used. The advantage of such tool is that it supports the SCAN bridge [84] protocol provided by the VME crate and the DAB64x card for multi drop JTAG [85]. Therefore, with a single JTAG connection to the VME P1 connector it could program all the flash memories without having to move the JTAG connection between modules. The same protocol can be also used to program the Configuration devices [87] with the FPGA firmware.

At every initialisation of the crate, which can be done either by the power switch or a system wide reset sent to the PowerPC, the FPGA fetches its firmware from the Configuration device. The FPGA's first task, when it goes to operating mode will be to obtain the Threshold table data and copy them to its embedded memory. For added security, those memories are also initialised by Memory Initialisation Files (MIF) [88] in order to hold the minimum thresholds in case the mezzanine becomes problematic and no thresholds have been written during the power-up of the FPGA.

After, the initialisation finishes and normal operation begins, the first Running Sum values will start arriving to the TC process. The Th-Table function will provide the Threshold values by reading the 5-bit Beam Energy level. Figure 7-3 shows the implemented schematic block, with Quartus II, of the Th-Table function. This block will provide the Threshold values to the TC process.

This function has been implemented in a way that the TC process, which was shown in the previous Section, can request at the same time data for Running Sums from different SRS processes by having different memory spaces for each of them. At the same time, the Threshold memory update procedure will be able to access the complete memory space as one large memory space, equal to the sum of the individual parts by including the memory page decoding in this function.

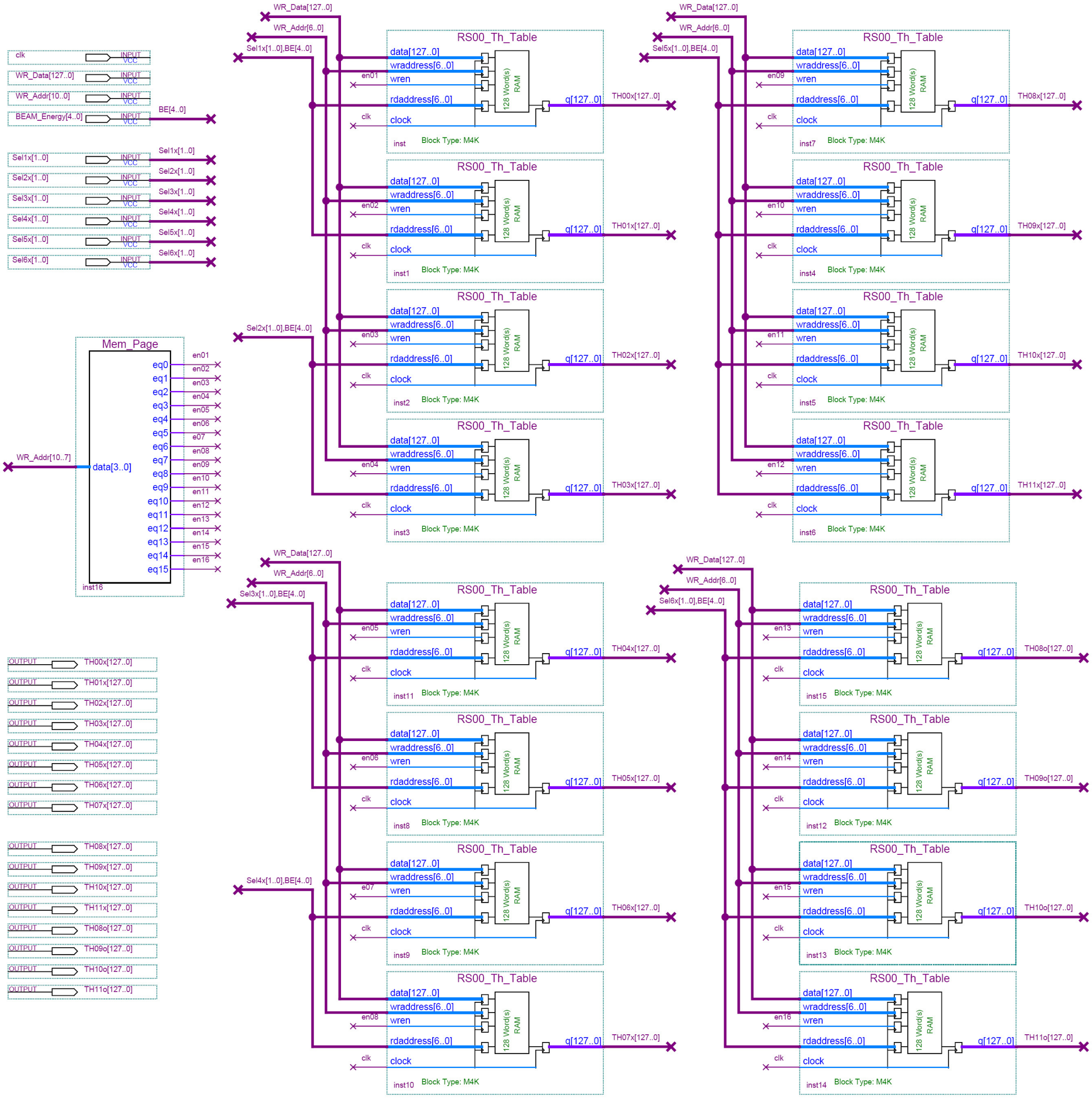
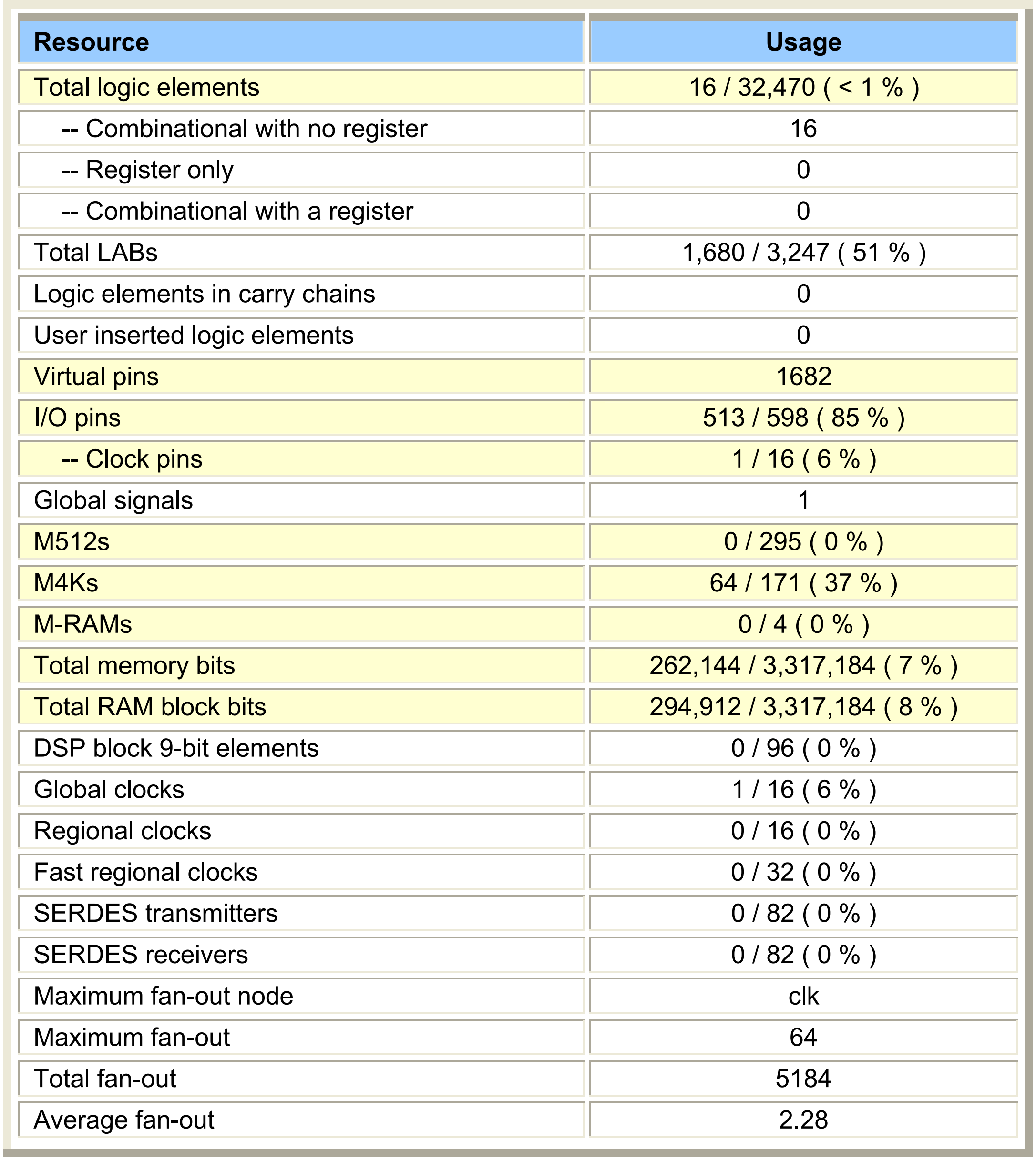


Figure 7-3: Schematic Block of the Th-Table Function.

7.4.3 RESOURCE UTILISATION BY THE THRESHOLD TABLE

The Logic Elements usage for the Th-Table function is negligible since only the decoding of the memory pages is realised using them. On the contrary, the memory block usage is 37% of the M4K blocks available in the EP1S30 device. Nevertheless, the efficiency is close to the maximum achievable as it was shown in Section 7.4.1. The resource utilisation report for the Th-Table function needed in the BLMTC system, calculated by the Quartus II Fitter tool, can be found in Table 7-7.

Table 7-7: Resource Utilisation by the Th-Table Function.



Resource	Usage
Total logic elements	16 / 32,470 (< 1 %)
-- Combinational with no register	16
-- Register only	0
-- Combinational with a register	0
Total LABs	1,680 / 3,247 (51 %)
Logic elements in carry chains	0
User inserted logic elements	0
Virtual pins	1682
I/O pins	513 / 598 (85 %)
-- Clock pins	1 / 16 (6 %)
Global signals	1
M512s	0 / 295 (0 %)
M4Ks	64 / 171 (37 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	262,144 / 3,317,184 (7 %)
Total RAM block bits	294,912 / 3,317,184 (8 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	1 / 16 (6 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	clk
Maximum fan-out	64
Total fan-out	5184
Average fan-out	2.28

7.5 Implementation of the Masking Process

The TC process forwards all the dump requests created from each of the Running Sums comparison with its corresponding threshold to the Masking process. Since the dump requests come from various channels at every reading, in addition, the decoding and channel selection for each of them is forwarded as well.

This information is used by the Masking process to retrieve the masking values needed from the table. Finally, the Inhibit blocks will provide the Maskable and Not-Maskable dump requests, as well as, post-processing information to understand which channel's Running Sum has exceeded the threshold level and requested the dump of the beam.

The VHDL code for the Inhibit block can be found in Appendix and Figure 7-4 illustrates the resulting Masking process implementation for the BLMTC system.

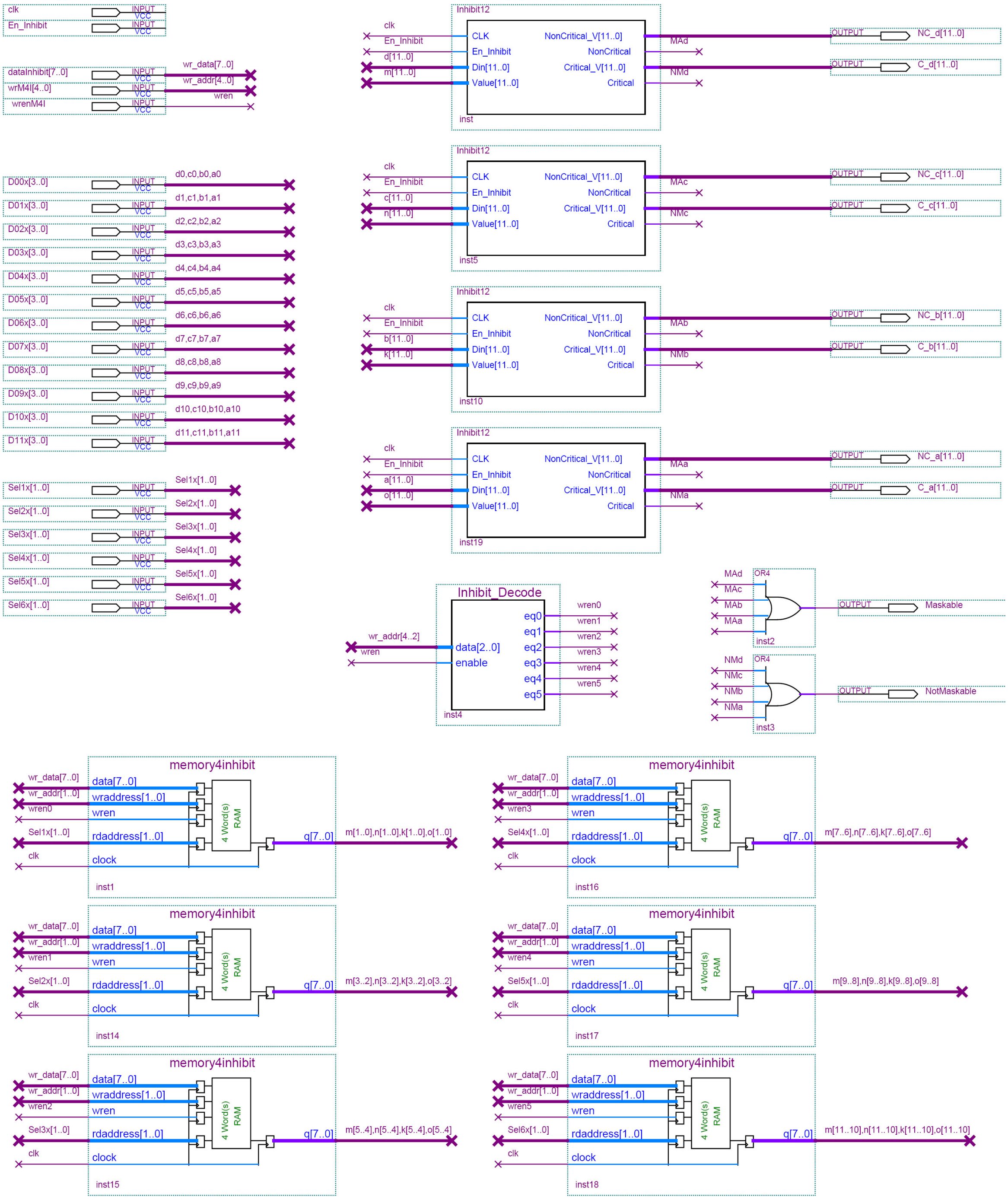
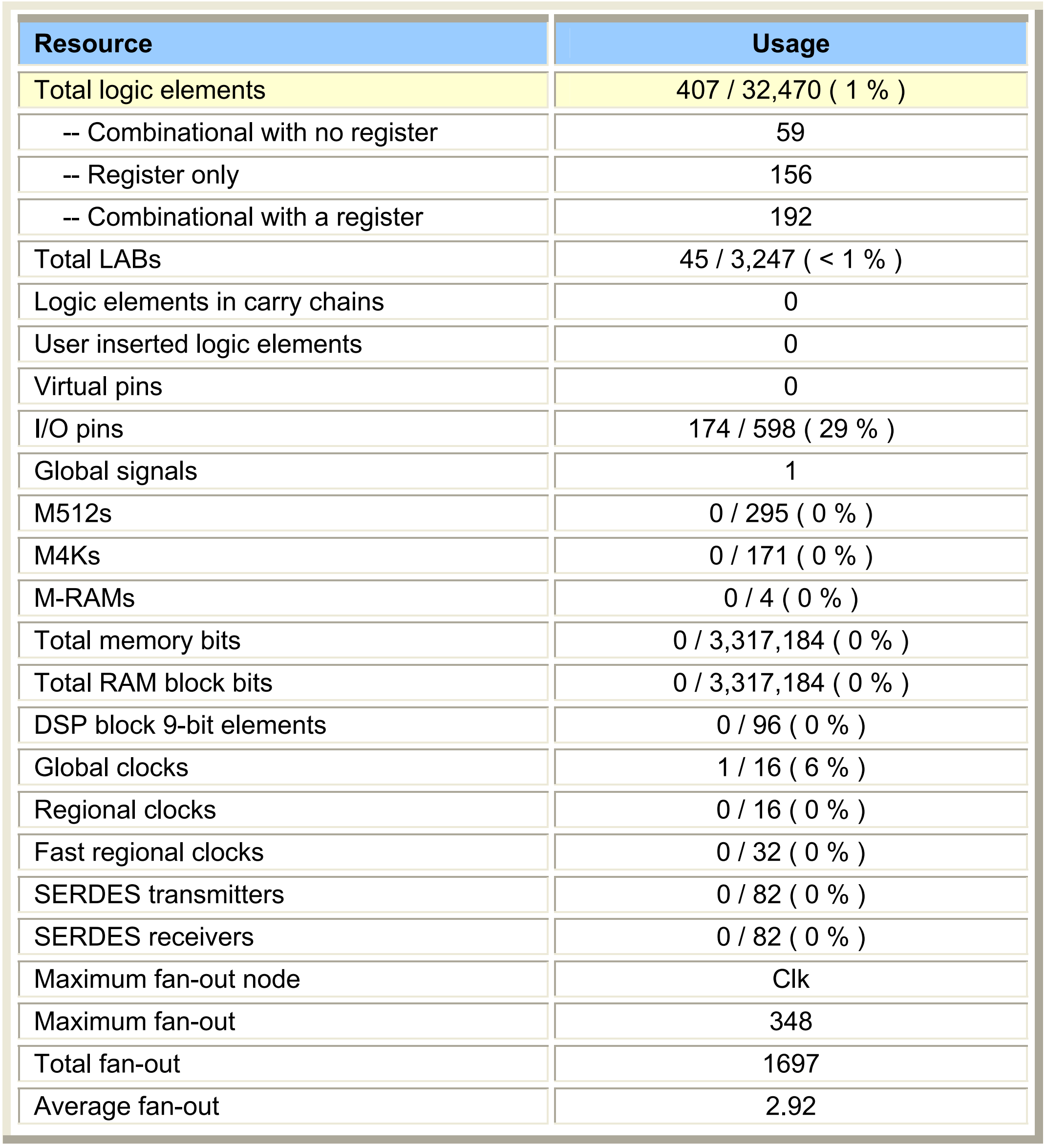


Figure 7-4: Masking Process for the BLMTC.

7.5.1 Resource Utilisation by the Masking Process

For the Masking process implementation the memory space needed was forced to be realised with Logic Cells instead of embedded memory blocks by using the necessary attributes in the Quartus II Analysis & Synthesis tool. Even though the memory space needed is very small, it has to follow and match the output of the TC process. Thus, if the embedded memory has been allowed to be used, at least six of M512 memory blocks would have been wasted. Instead, by using the Logic Cell option it is using less than 407 Logic Elements for its complete realisation. The resource usage report for the Masking process needed in the BLMTC system, calculated by the Quartus II Fitter tool, can be seen in Table 7-8.

Table 7-8: Resource Utilisation by the Masking Process.



Resource	Usage
Total logic elements	407 / 32,470 (1 %)
-- Combinational with no register	59
-- Register only	156
-- Combinational with a register	192
Total LABs	45 / 3,247 (< 1 %)
Logic elements in carry chains	0
User inserted logic elements	0
Virtual pins	0
I/O pins	174 / 598 (29 %)
Global signals	1
M512s	0 / 295 (0 %)
M4Ks	0 / 171 (0 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	0 / 3,317,184 (0 %)
Total RAM block bits	0 / 3,317,184 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	1 / 16 (6 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	Clk
Maximum fan-out	348
Total fan-out	1697
Average fan-out	2.92

7.6 SUMMARY

The threshold level for each of the approximately 4000 ionisation chambers needs to be set and the threshold value is inversely proportional to the beam energy, which will follow a ramp function. Therefore, the table consisting of those data could be considered as a 3-dimensional table showing the number of protons lost over time, over beam energy, and over ionisation chamber. The consideration of having a global database of different Threshold and Warning values stored on each card would be highly inefficient. It would actually mean a global table, which requires a significantly large memory space reserved on each of the cards, exceeding 3 billion values.

In order to minimise that table and the memory needed, it is proposed to spread the load between them. Thus, a unique block of values will be created for each of the cards of the complete system. The information included will be still for calculating its Running Sums, but for 32 Beam Energy Levels and the specific 16 Ionisation Chambers, each BLMTC card is reading. That would give a total of 6,144 Threshold values needed to be held on each card.

At the same time, the Warning levels could be calculated and transmitted to the Control Room by the PowerPC situated in the VME crate. The BLMTC system will provide the calculated Running Sums together with their corresponding threshold values through the Logging process to simplify the actions need to be done by the CPU.

The TC process will have the main task of comparing the calculated Running Sum values with their corresponding Threshold values for the given Beam Energy. Nevertheless, since 192 comparisons take place at each acquisition, some of them 32 and others 64-bit long, its implementation needed to use a highly efficient strategy.

The resource usage was significantly reduced by a resource sharing scheme taking advantage of the specific implementation of the SRS process and the time difference in the arrival of the Running Sum values. As a result, the resource utilisation for the complete TC process is approximately 15% of the Logic Elements available in the EP1S30 device. The efficiency achieved has a great impact on the BLMTC system’s fitting, since it reduced the usage by approximately 75%.

The Th-Table function was created to provide the Threshold values by reading the 5-bit Beam Energy level. This function has been implemented in a way that the TC process can request at the same time data for Running Sums from different SRS processes by having different memory spaces for each of them. At the same time, the Threshold memory update procedure will be able to access the complete memory space as one large memory space. Its memory block usage has efficiency close to the maximum achievable.

Similar strategy was proposed also for the Masking table preserving uniformity throughout the design. Thus, unique tables will need to be produced, holding only the necessary information specific to the channels the card is processing. Two pieces of information will be available, for each of the 16 channels the card is processing. Those will be the detector connection status and in which type of families it belongs, i.e. if it is Maskable or not.

The TC process forwards all the dump requests created from each of the Running Sums comparison with its corresponding threshold to the Masking process. The Masking process will provide the Maskable and Not-Maskable dump requests, as well as, post-processing information to understand which channel's Running Sum has exceeded the threshold level and requested the dump of the beam.

In the Masking process implementation the memory space needed was forced to be realised with Logic Cells instead of embedded memory blocks by using the necessary attributes in the Quartus II Analysis & Synthesis tool. Instead of wasting at least six of M512 memory blocks, by using the Logic Cell option it is using less than 1% of the Logic Elements available for its complete realisation.

Finally, many advantages are emerging from the proposed table strategy. The unique tables for each of the monitors can thoroughly be prepared and checked before they are uploaded, and they can be quickly and easily upgraded only for specific systems. The table can be used as a calibration and offset tool, and it is small enough to be kept internally by the Stratix device that will increase the access time and the implementation simplicity, if compared to an external memory device.

REFERENCES

[81] J.B. Jeanneret, D. Leroy, L. Oberli and T. Trenkler, LHC Project, *"Quench levels and transient beam losses in LHC magnets"*, LHC Report 44, CERN, July 1996

[82] A. Arauzo-Garcia et al., *"LHC Beam Loss Monitors"*, 5th European Workshop on Diagnostics and Beam Instrumentation DIPAC 2001, Grenoble, France

[83] CERN LHC Project Document, *"The Beam Interlock System for the LHC"*, Engineering Specification, LHC-CIB-ES-0001-00-10, rev. 1.0, 17-02-2005 [online: https://edms.cern.ch/document/567256]

[84] IEEE Standard Board, IEEE std 1149.5-1995, *"Module Test and Maintenance Bus (MTM-Bus) Protocol"*, 1995.

[85] JTAG Technologies, [online: https://www.jtag-technologies.com]

[86] SCAN bridge, [online: http://www.national.com/appinfo/scan]

[87] Altera Corp. *"Configuration Devices"*, [online: http://altera.com/products/devices/config/cfg-index.html]

[88] Altera Corp. *"Memory Initialisation File (.mif) Definition"* [online:http://www.altera.com/support/software/nativelink/quartus2/glossary/def_mif.html]

08

Data Logging & Post-Mortem Recording

Chapter 8. Data Logging & Post-Mortem Recording

In the LHC, storage of the loss measurements are needed to allow to trace back the loss signal developments as well as the origin of the beam losses in conjunction with other particle beam observation systems.

Transient data must be recorded for all beam and equipment parameters over a sufficiently long time interval around a beam or power abort to reconstruct the event sequence. This data set is referred to as the Post-Mortem Event. The collection and the concentration of the data in the form of the post-mortem event and the subsequent analysis of this data are the task of the Post-Mortem System. Its main objective is the reconstruction of the event sequence that leads to a beam or power abort. [89]

The main purpose of the Logging System is to continuously record and provide an online display to the Control Room of the machine status and show slow or infrequent changes. The BLM System will contribute by providing the loss rates, normalised with respect to the quench levels so that abnormal or higher local rates could thereby be spotted easily.

The storage of the beam loss (logging) data will allow more sophisticated off-line studies like frequency analysis and the possibility of long term summation for comparison with data on integrated radiation doses all around the machine. [90]

An overview of the Logging and Post Mortem systems is illustrated in Figure 8-1. The BLM System is one of the systems that have to provide data to both of them. Because not only it will provide general diagnostic data but also, since it will be a critical part of the Machine Protection System that has the ability to trigger a Beam Abort, its performance in this role must be systematically recorded.

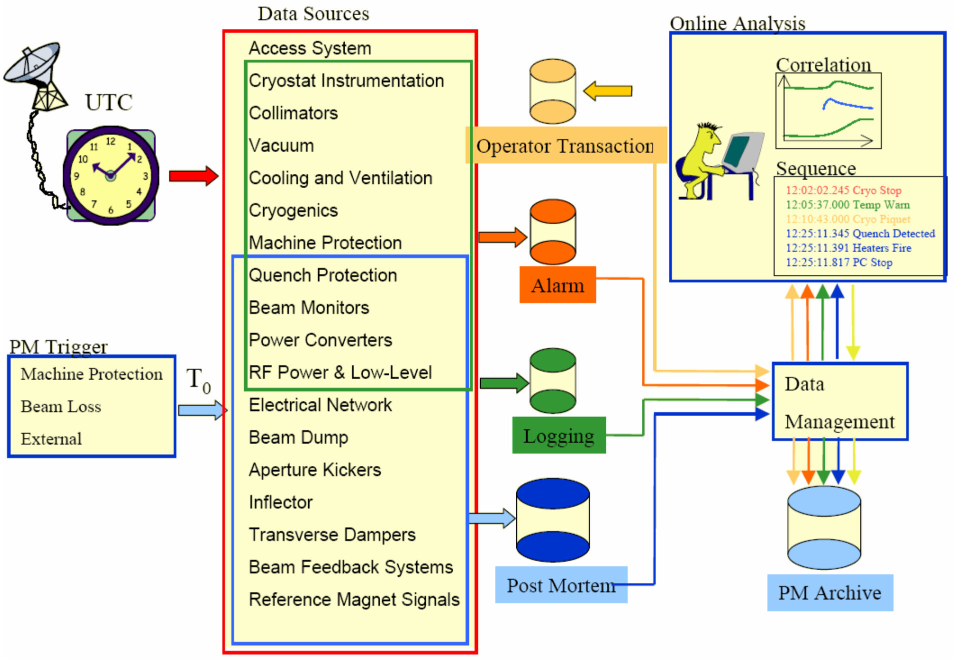


Figure 8-1: Logging and Post Mortem System Overview.[91]

This detailed fault diagnostics will help improve the operational efficiency of the LHC. It will provide diagnostics of “incidents” for the Control Room with the aim of identifying the origin of failures in order to initiate appropriate actions and restore operation, as well as building long term understanding of accelerator operation by collecting and managing the hardware and beam performance data. Furthermore, it will ensure comprehensive monitoring of the quench detection, the machine protection and the beam dumping systems and in the case of damage occurrence, to explain the mechanism. [91]

In this Chapter, a strategy and the necessary parameters needed for the BLMTC to prepare and provide the data to both systems will be defined as well as the implementation proposed will be shown.

8.1 DATA LOGGING

Each BLMTC card handles 16 detectors (Ionisation Chambers). From each of these cards three sets of information have to be transmitted for the logging purposes: the Threshold Table's Values used, the Maximum Values of the Beam Loss Data, and the Error and Status reports.

The Maximum values of each of the Running sums occurred in the last second are calculated and are read with the same rate in order to be stored in a database as well as to give a graphical representation in the control room. The Maximum values between different Running sums can have many orders of difference, thus, in order to be easily comparable they need to be normalised. One solution for this normalisation is the division with their corresponding threshold value.

The complete threshold table holds the data needed for the comparison with the running sum values calculated. The threshold data are detector specific, i.e. each card holds unique data for each detector, and are also depending to the beam energy. This table is not allowed to change while a beam is circulating for security reasons, thus it is not necessary to be read on every logging interval. Nevertheless, it is needed to know which threshold values were used for the comparison. Firstly, the PowerPC will have them available to normalise the Running Sums for the display and secondly the discovery of potential problems with the reading of the beam energy or the comparison will be possible.

The last set of data needed for the Logging is the Error and Status reports collected by the system. Those will mainly include the CRC errors, to indicate the errors detected in the optical transmission, and the Status bits, to indicate whether a status flag was raised during that time from the tunnel, i.e. from the High Tension, the CFC power, or any counter error. Additional information of the number of acquisitions made by the CFC cards, the number of packet transmissions made from the CFC card and to the PowerPC, as well as loss of synchronisations and lost packets will be available.

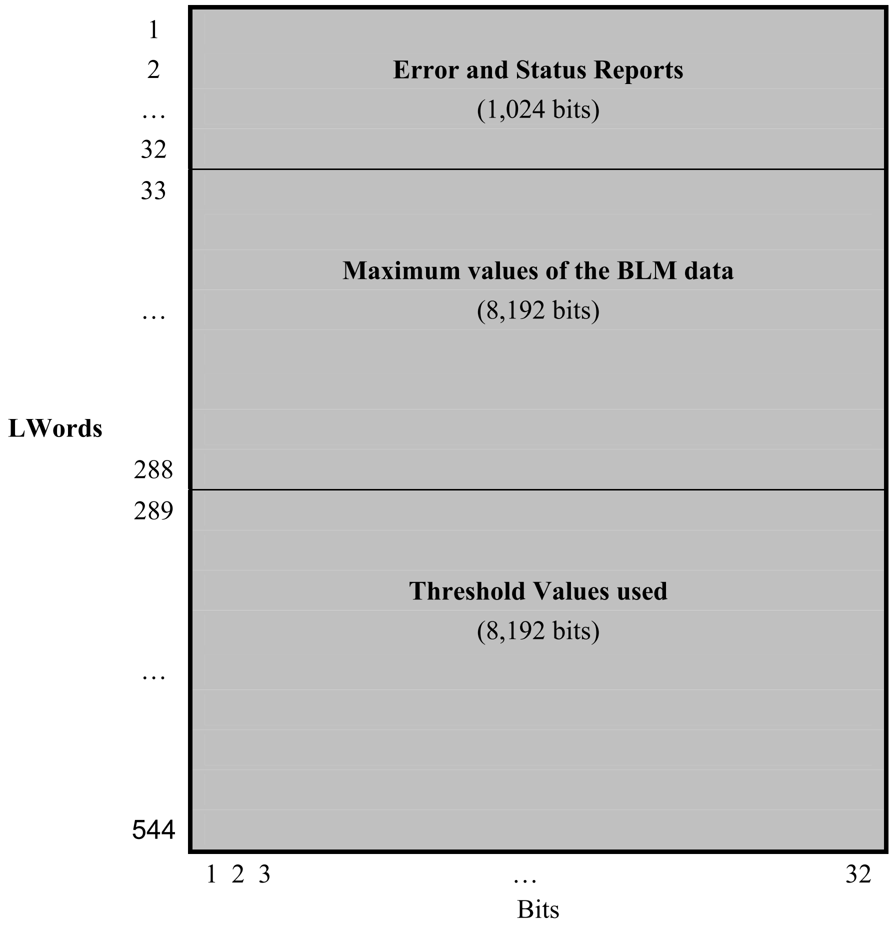


Figure 8-2: Transmitted Packet for the Logging System.

In Figure 8-2 is shown the format of the packet transmitted for the logging purposes. The VME access employs reading of long words, i.e. 32 bits, and will need to read from each card 544 long words every second.

Finally, every crate accommodates 16 BLMTC cards from which the CPU has to read those packets. Each card will have internally a memory location assigned for this operation where it will write those data and update them once every second. The CPU will arrange its read access circularly to each card but without any synchronisation, thus each card will need to know when its data have been read and when to update them without any loss of data.

8.2 POST MORTEM RECORDING

When there is a beam dump requested, the data recording for the Post Mortem (PM) should freeze with no delay. In that case, each BLMTC will be informed to freeze its PM's recording through the VME bus by the crate CPU.

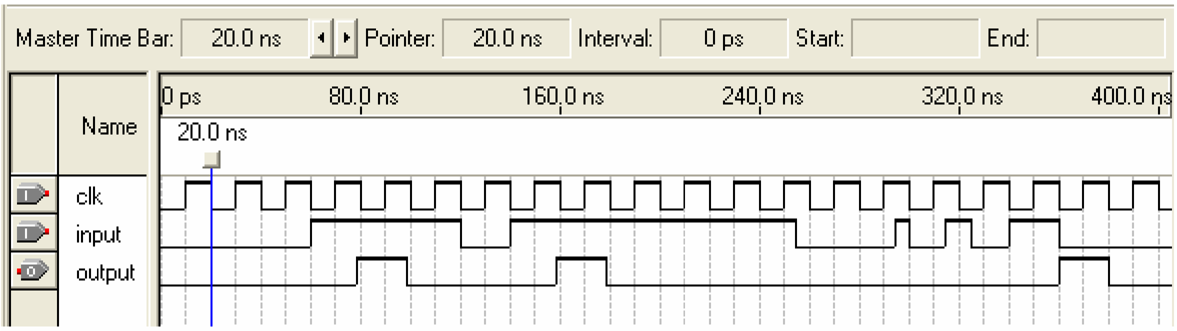
The information that a dump has occurred, i.e. the PM trigger, together with a precise universal time reference, the Universal Time Co-ordinates (UTC) [92] timing, is transmitted by the Timing, Trigger and Control (TTC) system [93]. The BOBR card [94], which will be included in each VME crate, will receive this information and will inform each BLMTC card about the dump occurred as well as provide a time-stamp of this event. Finally, the PowerPC will be able to combine the time-stamp with the PM data it will read from each card before storing it into the database.

Instead of the system having to wait for the read of all the PM data from each BLMTC card before these are deleted or overwritten and additionally to overcome the problem of two consecutive stop triggers the implementation will make use of two circular buffers in each card. The trigger will be utilised to toggle the writing from one to the other. In this way, the recording will never have to be stopped and will be independent of the CPU access.

8.3 IMPLEMENTATION OF THE ERROR AND STATUS REPORTING

In Chapter 4, it was shown in the implementation of the RCC process that many signals and flags were prepared and outputted specially to be used by the Error and Status Reporting process. Some of them were the errors in the CRC calculations or the comparisons, the status flags coming from the tunnel installations and the new packet arrival flag from the optical links.

Thus, most of the work in the implementation of this process has already been done. It was only needed to place a counter for each of the signals that would increment at every new assertion. In order to ensure that each flag assertion was only counted once a "OnePulse" circuit was included in the input of the counter. The VHDL source code for this circuit is shown in Appendix A.4.1. It samples the input using the system clock and provides a pulse which length is equal to one clock cycle disregarding the longer input pulses. The functionality of the OnePulse circuit can be seen in the timing waveform of Figure 8-3.



Master Time Bar: 20.0 ns Pointer: 20.0 ns Interval: 0 ps Start: End:

Figure 8-3: OnePulse Circuit Functionality.

8.3.1 THE ESLOG FUNCTION

Some additional signals were also included, like the Maximum Values' reset and the VME access flags, and supervisor circuits were created to check if packets were missing from the optical links. An asynchronous reset was also included to all of them to clear the counter values when necessary. An illustration of the schematic block that includes the above functions can be found in Figure 8-4.

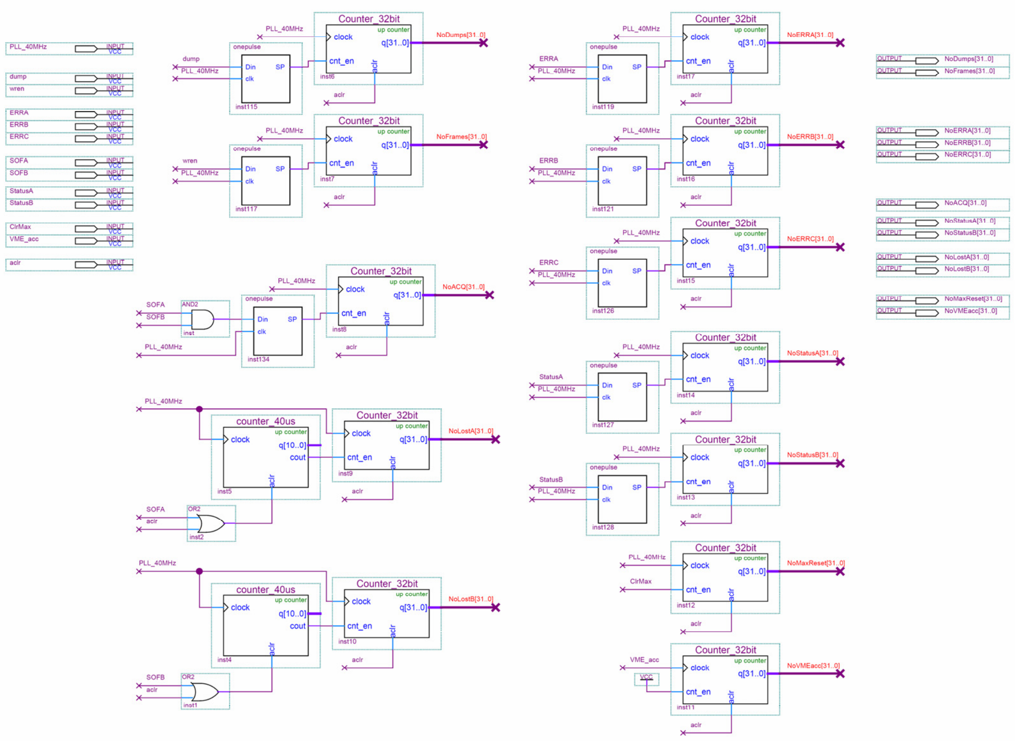


Figure 8-4: ESLog Block for the Error and Status Reporting.

8.3.2 ERROR AND STATUS REPORTING (ESR) IN THE BLMTC

In order to implement the Error and Status Reporting for the BLMTC system, two of the “ESLog” blocks have been used which accommodate mainly the signals related to one CFC card. Additionally it includes the internal storage of these values and the readout function necessary for the PowerPC access. The update of the storage is initiated after every VME access and is using two multiplexer circuits to store the data, from the ESLog functions and the other locations, sequentially. Figure 8-5 shows the implementation of the Error and Status Reporting process.

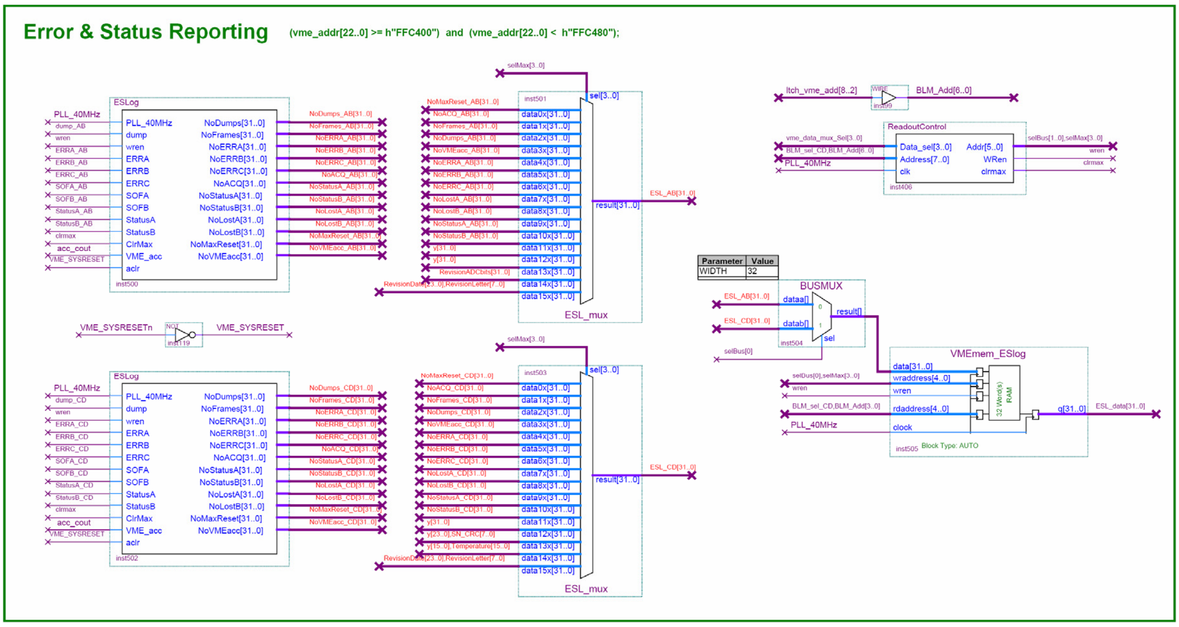


Figure 8-5: Error and Status Reporting (ESR) Process.

Table 8-1 provides the memory mapping for the Error and Status Reporting available in each BLMTC card needed to be read by the PowerPC. The arrangement simply follows the sequence of the complete readout from each of the two “ESLmux” multiplexers used.

Table 8-1: Memory Mapping of the Error and Status Reporting.

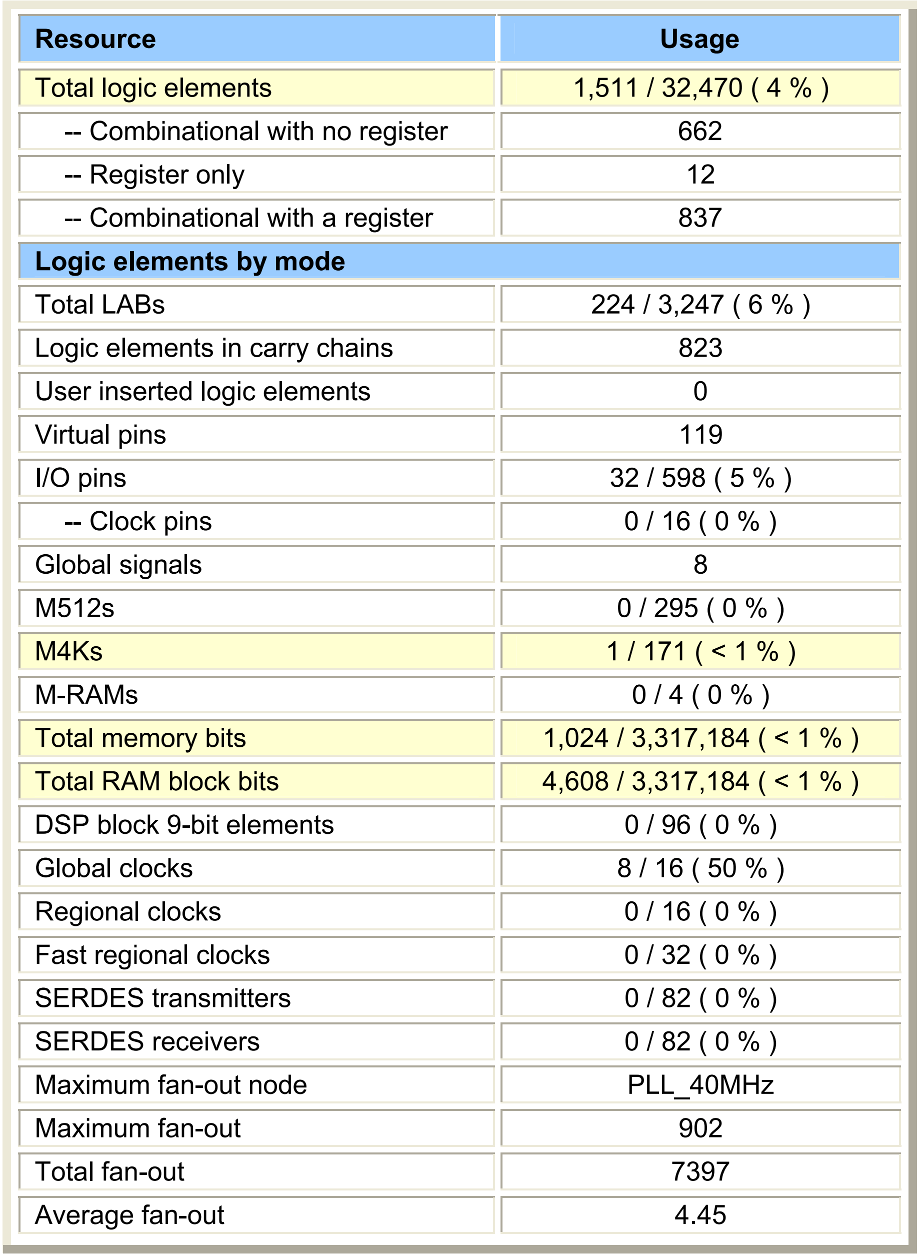
Error and Status Reporting

address	Description	address	Description
0xFFC400	Resets of Maximum Values	0xFFC440	Resets of Maximum Values
0xFFC404	Acquisitions by CFC1	0xFFC444	Acquisitions by CFC2
0xFFC408	VME Reads	0xFFC448	VME Reads
0xFFC40C	Dumps from CFC1	0xFFC44C	Dumps from CFC2
0xFFC410	Reserved	0xFFC450	Reserved
0xFFC414	ERRA (Ch.1)	0xFFC454	ERRA (Ch.3)
0xFFC418	ERRB (Ch.2)	0xFFC458	ERRB (Ch.4)
0xFFC41C	ERRC (Ch.1 & 2)	0xFFC45C	ERRC (Ch.3 & 4)
0xFFC420	LostFramesA (Ch.1)	0xFFC460	LostFramesA (Ch.3)
0xFFC424	LostFramesB (Ch.2)	0xFFC464	LostFramesB (Ch.4)
0xFFC428	StatusA (Ch.1)	0xFFC468	StatusA (Ch.3)
0xFFC42C	StatusB (Ch.2)	0xFFC46C	StatusB (Ch.4)
0xFFC430	Reserved	0xFFC470	Reserved
0xFFC434	Reserved	0xFFC474	CN_CRC
0xFFC438	Revision Rounding	0xFFC478	DAB Temperature
0xFFC43C	Revision of PartA	0xFFC47C	Revision of PartB

8.3.3 RESOURCE UTILISATION BY THE ESR PROCESS

The resource usage by the ESR process requires a small amount of Logic Elements, which is of the order of 4% of the EP1S30 or 3% of the EP1S40 device, and only one M4K memory block for its storage needs. The complete report after fitting the process using Quartus II for the EP1S30 device can be seen in Table 8-2.

Table 8-2: Resource Utilisation by the ESR Process



Resource	Usage
Total logic elements	1,511 / 32,470 (4 %)
-- Combinational with no register	662
-- Register only	12
-- Combinational with a register	837
Logic elements by mode	
Total LABs	224 / 3,247 (6 %)
Logic elements in carry chains	823
User inserted logic elements	0
Virtual pins	119
I/O pins	32 / 598 (5 %)
-- Clock pins	0 / 16 (0 %)
Global signals	8
M512s	0 / 295 (0 %)
M4Ks	1 / 171 (< 1 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	1,024 / 3,317,184 (< 1 %)
Total RAM block bits	4,608 / 3,317,184 (< 1 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	8 / 16 (50 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	PLL_40MHz
Maximum fan-out	902
Total fan-out	7397
Average fan-out	4.45

8.4 IMPLEMENTATION OF THE MAXIMUM VALUES LOGGING

The Running Sum values as it was shown in Table 6-2 have different widths and thus registers with different numbers of bits are used to hold them. Those registers can be 20, 22, 26, 32, 36, or 40-bit long. Additionally, since their production is successive, those values are updated and become available at different times. The implementation chosen for the calculation and logging of the Maximum values is taking into account those parameters in order to limit the unnecessary calculations that have a direct impact on the fitting and the power consumption of the FPGA. It uses unique calculation blocks for the six different register widths as well as the new data available flags for each of them.

8.4.1 MAXIMUM VALUE CALCULATION

The function that calculates the maximum value uses, every time a new value arrives, a comparator to compare this value with one that it holds in its register. The arrival of the new value is signalled each time by the read enable flag given by the data processing block, i.e. the SRS process shown in Chapter 6.

If the new value arriving is found to be bigger than the one stored in its register, the register is updated with this new value. Otherwise, it retains its old value. A reset initialises and restarts the calculation by setting all the registers values to zero. The VHDL source code for the 20-bit maximum value calculation, one of the six implemented for the different input widths, is shown in Appendix A.4.2. The code in each case instantiates a comparator equal to the input width and provides the output with a latency of one clock cycle.

Consequently, the six VHDL source codes were imported in the Quartus II to create symbol blocks. Those symbols were used as necessary to build the schematic block to treat the signals coming from the SRS processes. The outputs have been driven to a multiplexer circuit that could be controlled by a function to store all of them sequentially.

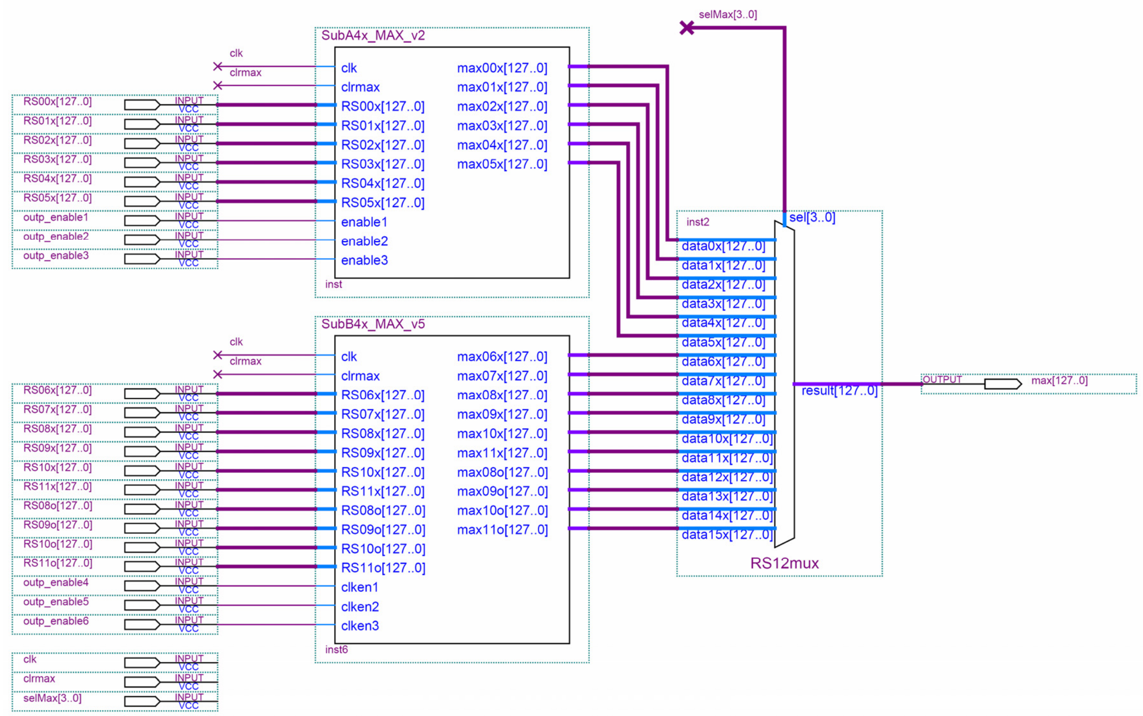


Figure 8-6: Maximum Values Calculation of the Successive Running Sums from four Detectors.

Figure 8-6 illustrates the schematic implementation of the “MaximumValues_4x” block to match the “RunningSums_4x” block’s outputs. It calculates and provides the maximum values for the 12 Successive Running Sums outputted from each of the four detectors treated by this function.

8.4.2 MAXIMUM VALUES LOGGING FOR THE BLMTC

In order to implement the Maximum Values Logging for the BLMTC system four “MaximumValues_4x” blocks have been used to accommodate all 16 detectors’ data. Additionally it includes the internal storage of these values and the readout function necessary for the PowerPC access. Figure 8-7 shows the implementation of the Maximum Values Logging process.

The calculation is synchronised with the read of the PowerPC by inspecting the VME access on those memory locations. After the complete table is read, a function is initiated which initially updates the table with the maxima that occurred since the last readout and then the registers are cleared so that a new calculation begins.

Thus, this implementation provides the data delayed by one VME reading, in this case it is delayed by one second. This strategy was done with the intention to gain the advantages that there is no loss of data if one reading for some reason is skipped. Moreover, the PowerPC will be allowed to vary the reading interval without any change needed on the FPGA firmware. Both situations are possible since it is yet not known if the CPU will be able to cope with the processing needs when all BLMTCs connect to the crate.

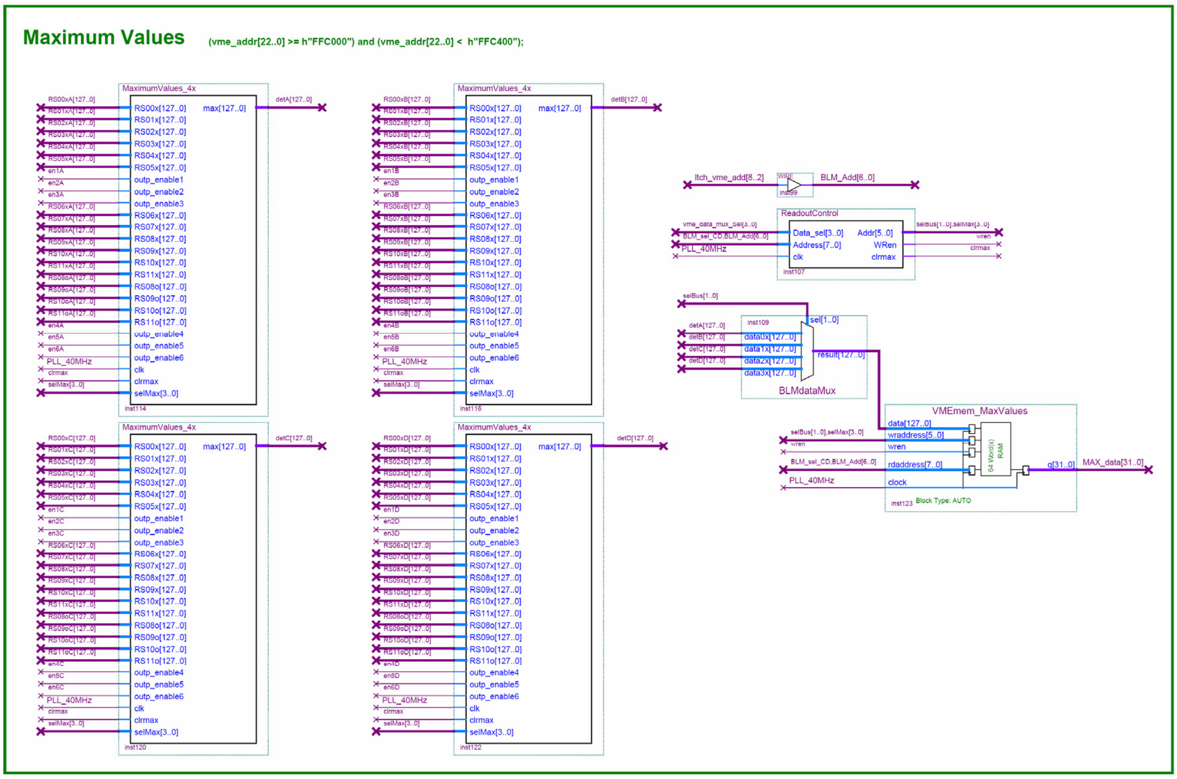


Figure 8-7: Maximum Values Logging for the BLMTC System.

Finally, Table 8-3 provides the memory mapping for the Maximum Values of the Successive Running Sums in each of the BLMTC cards needed to be read by the PowerPC. The arrangement simply follows the sequence of the complete readout from each of the four “MaximumValue_4x” used.

Table 8-3: Memory Mapping for the Maximum Values of the Successive Running Sums.

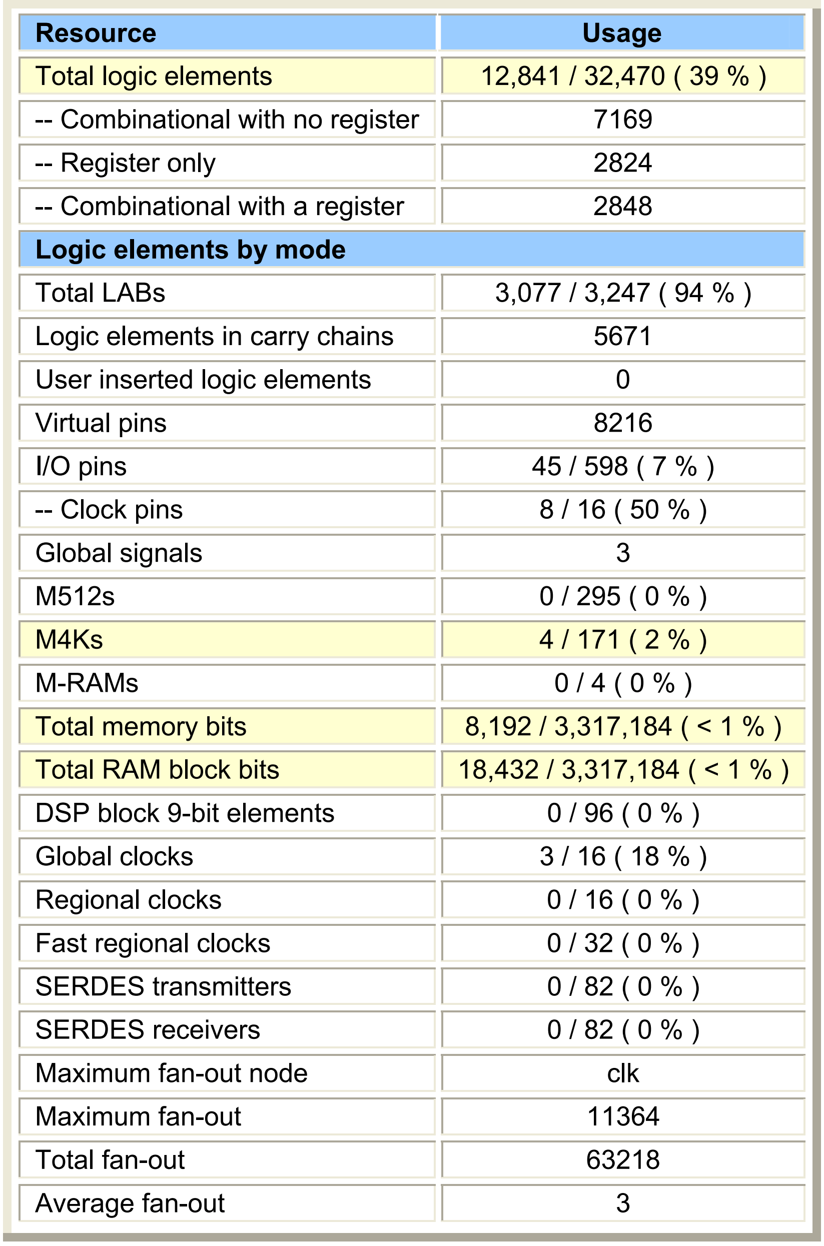
Maximum Values of the BLMTC Data

address	Description	address	Description
0xFFC000	Detector 1 / Max 1	0xFFC200	Detector 9/ Max 1
0xFFC004	Detector 2 / Max 1	0xFFC204	Detector 10/ Max 1
0xFFC008	Detector 3 / Max 1	0xFFC208	Detector 11/ Max 1
0xFFC00C	Detector 4 / Max 1	0xFFC20C	Detector 12/ Max 1
0xFFC010	Detector 1 / Max 2	0xFFC210	Detector 9/ Max 2
0xFFC014	Detector 2 / Max 2	0xFFC214	Detector 10/ Max 2
	. . .		. . .
0xFFC0B0	Detector 1 / Max 12	0xFFC2B0	Detector 9/ Max 12
0xFFC0B4	Detector 2 / Max 12	0xFFC2B4	Detector 10/ Max 12
0xFFC0B8	Detector 3 / Max 12	0xFFC2B8	Detector 11/ Max 12
0xFFC0BC	Detector 4 / Max 12	0xFFC2BC	Detector 12/ Max 12
0xFFC0C0	Detector 1 / Ovr 9	0xFFC2C0	Detector 9/ Ovr 9
0xFFC0C4	Detector 2 / Ovr 9	0xFFC2C4	Detector 10/ Ovr 9
0xFFC0C8	Detector 3 / Ovr 9	0xFFC2C8	Detector 11/ Ovr 9
0xFFC0CC	Detector 4 / Ovr 9	0xFFC2CC	Detector 12/ Ovr 9
0xFFC0D0	Detector 1 / Ovr 10	0xFFC2D0	Detector 9/ Ovr 10
0xFFC0D4	Detector 2 / Ovr 10	0xFFC2D4	Detector 10/ Ovr 10
0xFFC0D8	Detector 3 / Ovr 10	0xFFC2D8	Detector 11/ Ovr 10
	. . .		. . .
0xFFC0FC	Detector 4 / Ovr 12	0xFFC2FC	Detector 12/ Ovr 12
0xFFC100	Detector 5 / Max 1	0xFFC300	Detector 13/ Max 1
0xFFC104	Detector 6 / Max 1	0xFFC304	Detector 14/ Max 1
0xFFC108	Detector 7 / Max 1	0xFFC308	Detector 15/ Max 1
0xFFC10C	Detector 8 / Max 1	0xFFC30C	Detector 16/ Max 1
0xFFC110	Detector 5 / Max 2	0xFFC310	Detector 13/ Max 2
0xFFC114	Detector 6 / Max 2	0xFFC314	Detector 14/ Max 2
	. . .		. . .
0xFFC1B0	Detector 5 / Max 12	0xFFC3B0	Detector 13/ Max 12
0xFFC1B4	Detector 6 / Max 12	0xFFC3B4	Detector 14/ Max 12
0xFFC1B8	Detector 7 / Max 12	0xFFC3B8	Detector 15/ Max 12
0xFFC1BC	Detector 8 / Max 12	0xFFC3BC	Detector 16/ Max 12
0xFFC1C0	Detector 5 / Ovr 9	0xFFC3C0	Detector 13/ Ovr 9
0xFFC1C4	Detector 6 / Ovr 9	0xFFC3C4	Detector 14/ Ovr 9
0xFFC1C8	Detector 7 / Ovr 9	0xFFC3C8	Detector 15/ Ovr 9
0xFFC1CC	Detector 8 / Ovr 9	0xFFC3CC	Detector 16/ Ovr 9
0xFFC1D0	Detector 5 / Ovr 10	0xFFC3D0	Detector 13/ Ovr 10
0xFFC1D4	Detector 6 / Ovr 10	0xFFC3D4	Detector 14/ Ovr 10
0xFFC1D8	Detector 7 / Ovr 10	0xFFC3D8	Detector 15/ Ovr 10
0xFFC1F8	. . .	0xFFC3F8	. . .
0xFFC1FC	Detector 8 / Ovr 12	0xFFC3FC	Detector 16/ Ovr 12

8.4.3 Resource Utilisation by the Maximum Value prosess

The resource usage by the Maximum Values process, on the contrary with the other Logging processes, requires a significant amount of Logic Elements, which is of the order of 39% of the EP1S30 or 28% of the EP1S40 device. This usage is dictated by the amount of signals needed to be supervised by this process. Thus, if a reduction is found necessary, signals coming from Running Sums with integration times longer than 1 sec could be transmitted directly reducing the usage by approximately 30%. The complete report after fitting the process using Quartus II for the EP1S30 device can be seen in Table 8-4.

Table 8-4: Resource Utilisation by the Maximum Value Process.



Resource	Usage
Total logic elements	12,841 / 32,470 (39 %)
-- Combinational with no register	7169
-- Register only	2824
-- Combinational with a register	2848
Logic elements by mode	
Total LABs	3,077 / 3,247 (94 %)
Logic elements in carry chains	5671
User inserted logic elements	0
Virtual pins	8216
I/O pins	45 / 598 (7 %)
-- Clock pins	8 / 16 (50 %)
Global signals	3
M512s	0 / 295 (0 %)
M4Ks	4 / 171 (2 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	8,192 / 3,317,184 (< 1 %)
Total RAM block bits	18,432 / 3,317,184 (< 1 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	3 / 16 (18 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	clk
Maximum fan-out	11364
Total fan-out	63218
Average fan-out	3

8.5 IMPLEMENTATION OF THE POST MORTEM

With the decision for many of the fine details for the Post Mortem System implementation still pending only a proposal and an evaluation of the possibilities was chosen to be made for the preparation and storage of these data in the BLMTC.

In Figure 8-8 can be seen an overview of the proposed Post Mortem implementation. Two circular buffers can be used to store the data. The realisation of those buffers will exploit two of the three available SRAMs in the DAB64x card. The PM trigger will toggle the writing between those buffers by asserting one of the two write enable pins as well as, in the inverse order, toggle the read location.

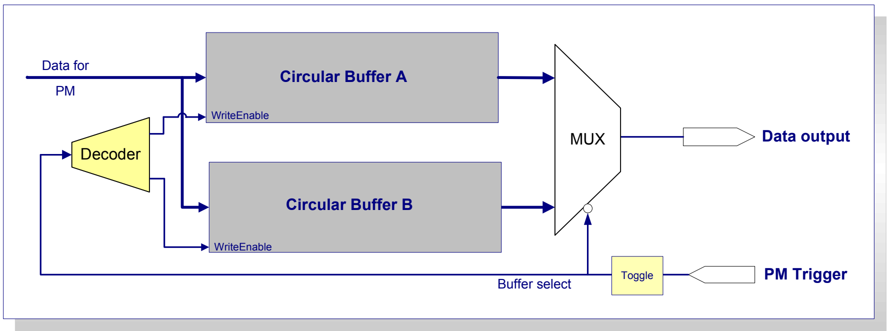


Figure 8-8: Overview of Post Mortem data storage.

The depth of the Post Mortem data should be sufficient to reconstruct approximately the last 1000 LHC turns. The LHC turn is 89µs thus, 2000 acquisitions will be sufficient. If both transmissions with their redundancies are included then 8000 packets will be needed to be saved on each BLMTC's buffer. Each packet was shown to be 256 bits thus, 250KB will be necessary to be read by the PowerPC for each card.

NB: The above gives 4MB per crate without calculating in it the time-stamps. Nevertheless, the resources, that is, the SRAM memory available on each BLMTC, can allow around eight times more data to be kept. If needed a circular buffer of a longer depth could be implemented. The limitation will lie at the PowerPC usage and its ability to cope with processing needs to recover, prepare and store the data.

8.6 DATA LOGGING USER INTERFACE

The Control Room, one of the main recipients of this process, would like to inspect the readings of the detectors also in a constant visual representation. (Figure 8-9 illustrates the architecture of the LHC controls.) For this purpose, the crate CPU will be used to normalise and distinguish the important data to pass further in the Control Room as well as calculate the Warning level alerts. The Warning level will initially be set as 30% of the Threshold level.

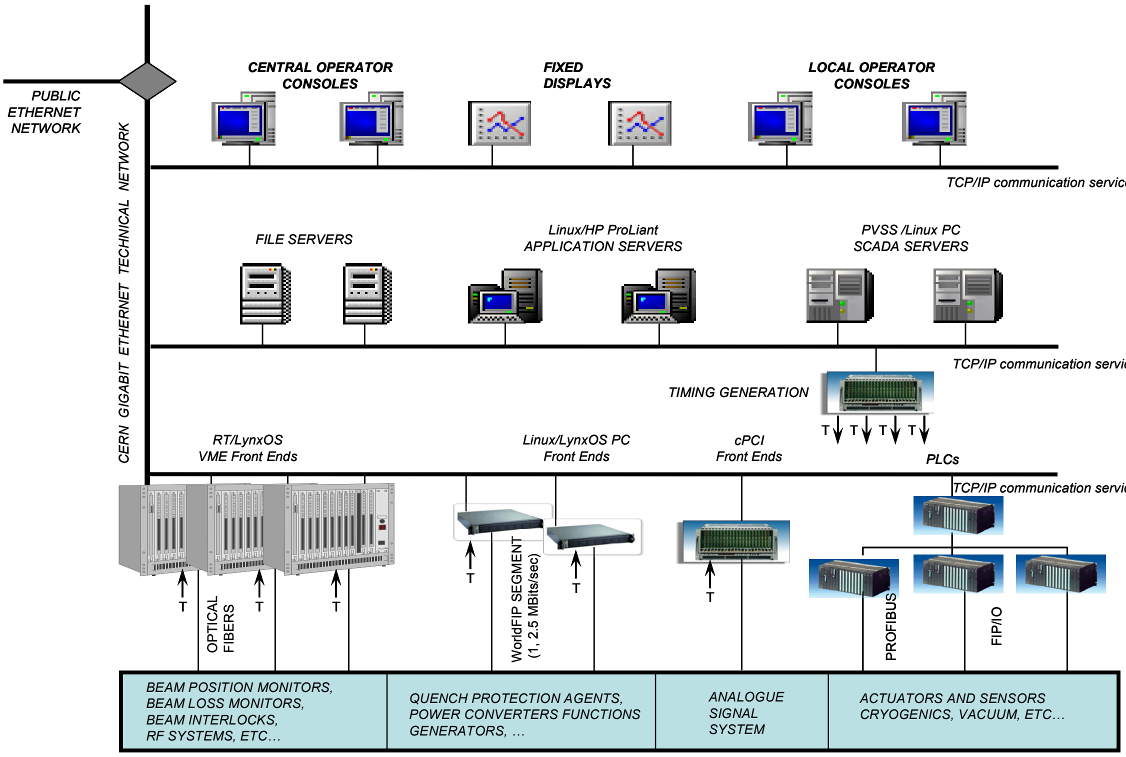

PUBLIC ETHERNET NETWORK
CERN GIGABIT ETHERNET TECHNICAL NETWORK
CENTRAL OPERATOR CONSOLES
FIXED DISPLAYS
LOCAL OPERATOR CONSOLES
TCP/IP communication services
FILE SERVERS
Linux/HP ProLiant APPLICATION SERVERS
PVSS /Linux PC SCADA SERVERS
TCP/IP communication services
TIMING GENERATION
T T T T
RT/LynxOS VME Front Ends
Linux/LynxOS PC Front Ends
cPCI Front Ends
PLCs
TCP/IP communication services
T
OPTICAL FIBERS
T
T
T
WorldFIP SEGMENT (1, 2.5 MBits/sec)
T
T
PROFIBUS
FIP/IO
BEAM POSITION MONITORS, BEAM LOSS MONITORS, BEAM INTERLOCKS, RF SYSTEMS, ETC…
QUENCH PROTECTION AGENTS, POWER CONVERTERS FUNCTIONS GENERATORS, …
ANALOGUE SIGNAL SYSTEM
ACTUATORS AND SENSORS CRYOGENICS, VACUUM, ETC…

Figure 8-9: LHC controls architecture diagram. [95]

More specifically, the 12 values measured and calculated for each detector will be normalised by their corresponding Threshold values and from those 12, the highest value will finally be displayed as a representative of that detector.

A graphical representation, that might look similar to the one of Figure 8-10, will give to the operator in a glimpse of an eye that the detectors numbered “3”, “5”, “3999” and “4000” have exceeded their warning levels and that the detector “3999” has exceeded its dump level and a beam dump procedure must have already happened because of it.

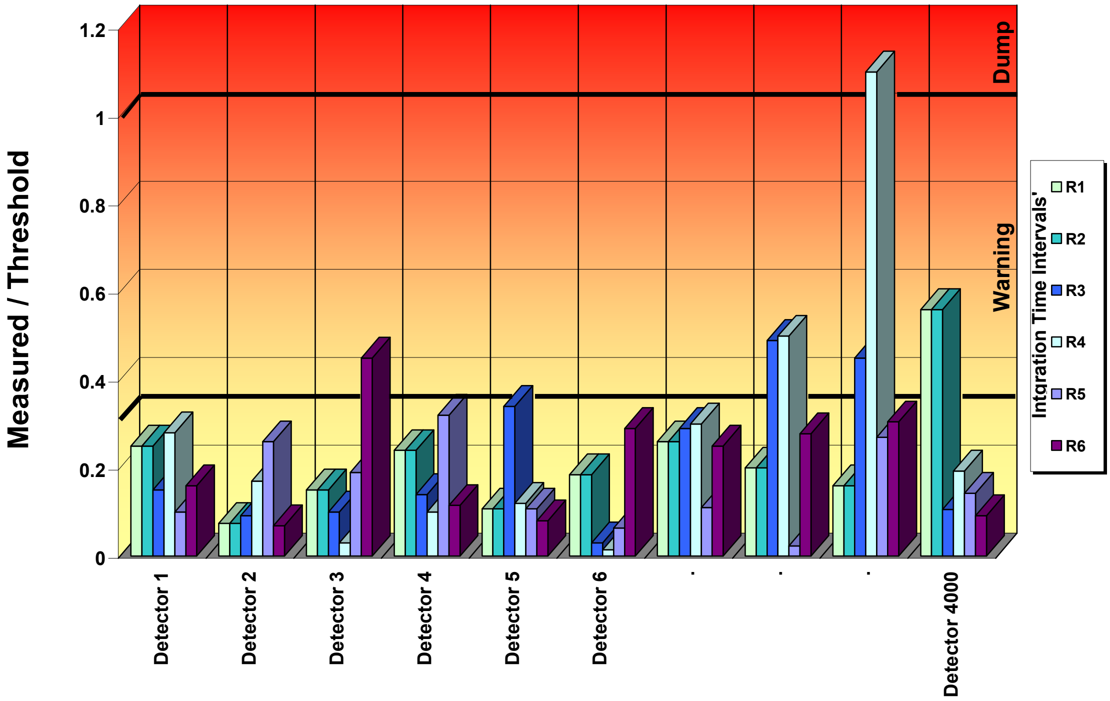


Figure 8-10: Proposed Graphical Representation of the Logging in the Control Room.

The realisation of such a system, usually referred as a Graphical User Interface (GUI) system, will be a task for the Control Group. Nevertheless, before this system becomes available, it was decided a simpler system to be implemented that will allow the display of the information collected by the BLMTC card. It was named BLM Acquisition Navigator and was implemented [96] within the FESA framework, using the proposal seen on this chapter.

The Front-End Software Architecture framework, known as FESA [97], is a complete environment for the equipment specialists to design, develop, test and deploy real-time control software for front-end computers. This framework is used to develop the LHC rings and injection chain front-end equipment software. Based on the BISCoTO [98] tools and functionality, the primary objective of this framework is to standardize, speed up and simplify the task of writing front-end software. The software tools (configuration, navigator and process activity survey tools) have been developed in Java and are thus platform independent. They are currently used under Linux and Windows 2000/XP. [99]

In the crate PowerPC a process is running that reads with an interval of one second the data provided for the logging, and stores in a circular buffer a history of the last 20 readings. In a PC, with an internet connection, a Java [100] application is launched that initiates the FESA tools, starts the Navigator and notifies the server running also in the PowerPC that it will like to receive the data collected. The Navigator thereafter receives the data and provides an interface to visualise them as well as to store them.

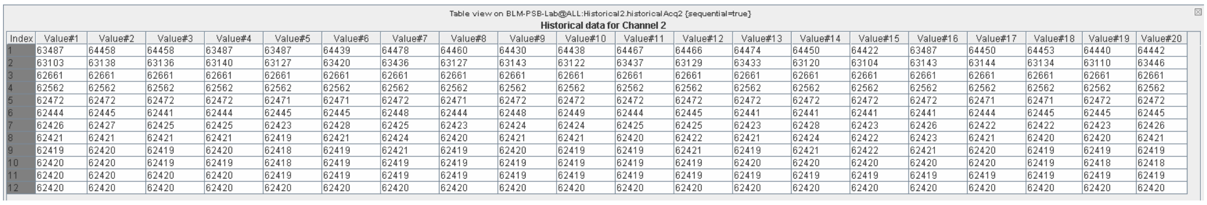


Table view on BLM-PSB-Lab@ALL:Historical2.historicalAcq2 {sequential=true}

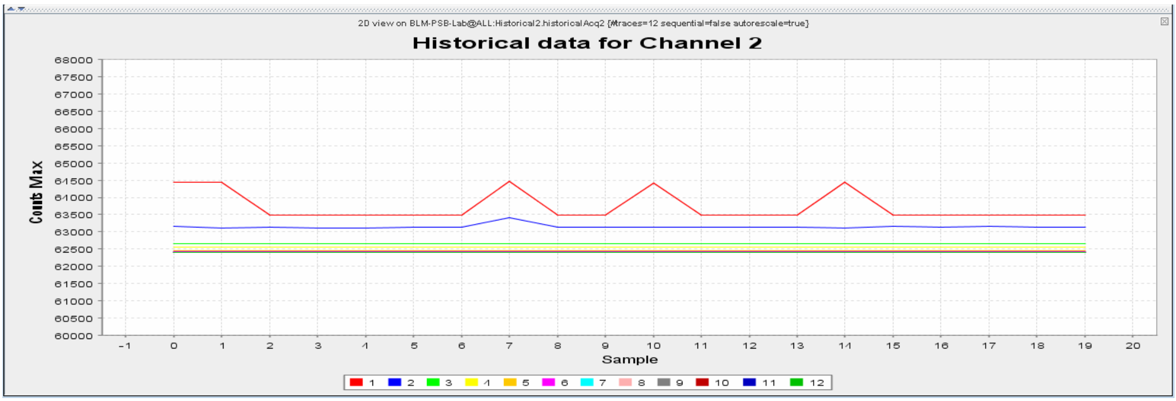
Historical data for Channel 2

Index	Value#1	Value#2	Value#3	Value#4	Value#5	Value#6	Value#7	Value#8	Value#9	Value#10	Value#11	Value#12	Value#13	Value#14	Value#15	Value#16	Value#17	Value#18	Value#19	Value#20
1	63487	64458	64458	63487	63487	64439	64478	64460	64430	64438	64467	64466	64474	64450	64422	63487	64450	64453	64440	64442
2	63103	63138	63136	63140	63127	63420	63436	63127	63143	63122	63437	63129	63433	63120	63104	63143	63144	63134	63110	63446
3	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661	62661
4	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562	62562
5	62472	62472	62472	62472	62471	62471	62472	62471	62472	62472	62472	62472	62472	62472	62472	62472	62471	62471	62472	62472
6	62444	62445	62441	62444	62445	62445	62448	62444	62448	62449	62444	62445	62441	62441	62441	62441	62444	62445	62445	62445
7	62426	62427	62425	62425	62423	62428	62425	62423	62424	62424	62425	62425	62423	62428	62423	62426	62422	62422	62423	62426
8	62421	62421	62421	62421	62419	62421	62424	62420	62421	62421	62420	62422	62421	62424	62422	62423	62421	62420	62420	62421
9	62419	62420	62419	62420	62418	62419	62421	62419	62420	62419	62419	62421	62419	62421	62422	62421	62420	62419	62419	62419
10	62420	62420	62419	62419	62418	62419	62419	62419	62419	62419	62419	62419	62420	62420	62420	62420	62420	62419	62418	62418
11	62420	62420	62420	62420	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419	62419
12	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420	62420

Figure 8-11: Display of Maximum Values History by the Navigator.

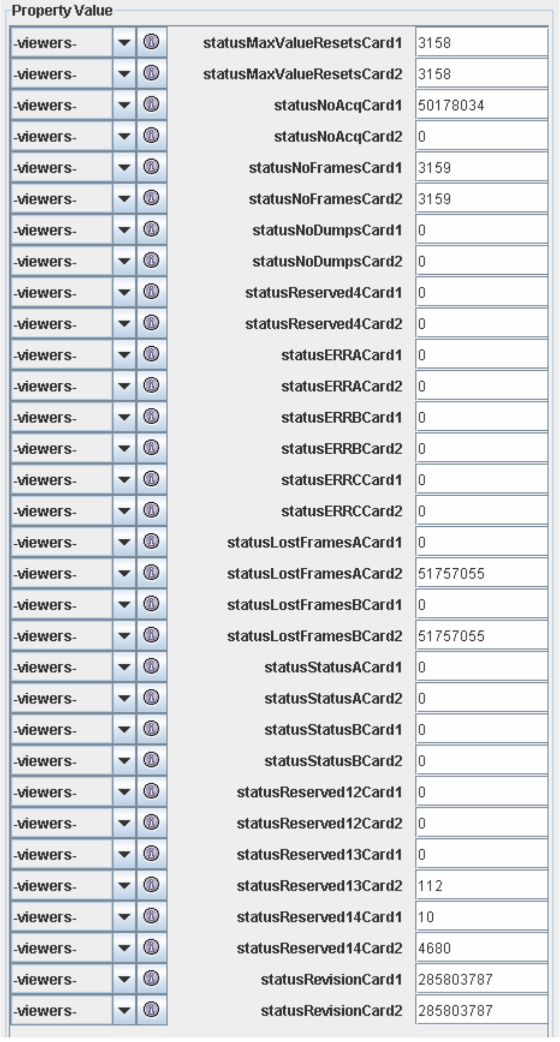
Figure 8-11 is a snapshot of the history of the 12 Maximum Values recorded for one of the 16 channels available from the Navigator. The data can be viewed in either the hexadecimal, the decimal, or the binary numerical base system by selection through a switch.

Figure 8-12 shows the available graph view of the data acquired for the same channel. Each trend displays the history of the value recorded for one of the Maximum Values of that channel with a depth of the last 20 readings. The table and the graph are dynamically updated every new reading.



2D view on BLM-PSB-Lab@ALL:Historical2.historicalAcq2 {#traces=12 sequential=false autorescale=true}

Figure 8-12: Plotting of Maximum Values History by the Navigator.



Property Value

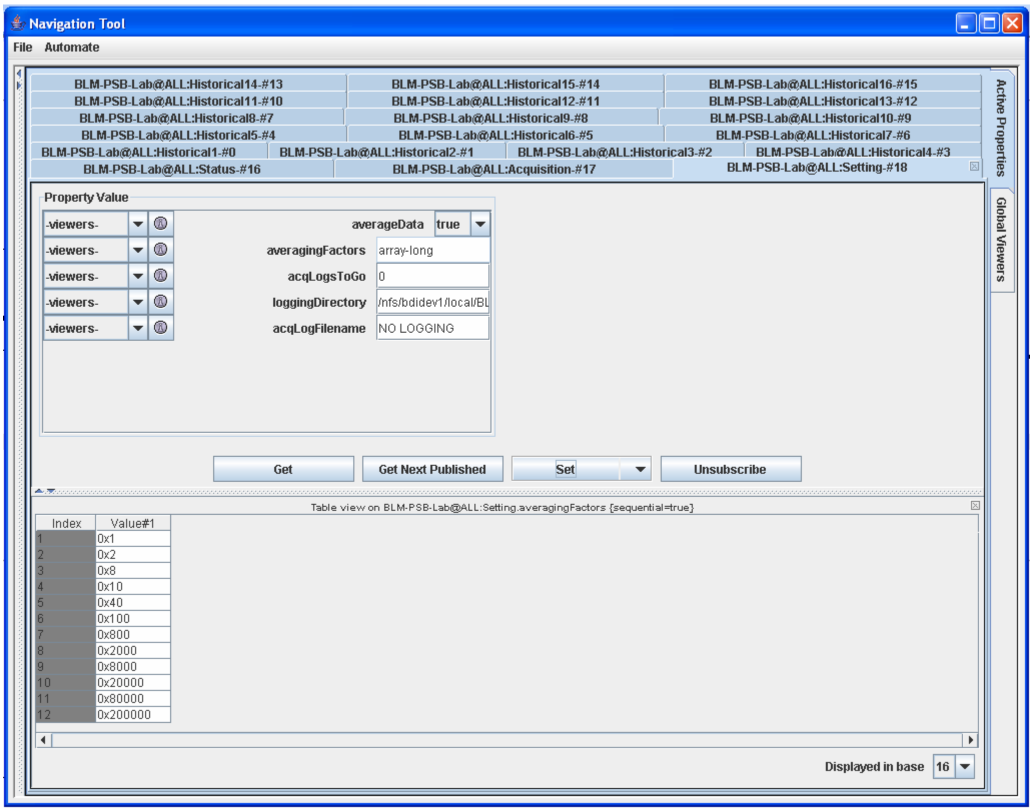
-viewers-	statusMaxValueResetsCard1	3158
-viewers-	statusMaxValueResetsCard2	3158
-viewers-	statusNoAcqCard1	50178034
-viewers-	statusNoAcqCard2	0
-viewers-	statusNoFramesCard1	3159
-viewers-	statusNoFramesCard2	3159
-viewers-	statusNoDumpsCard1	0
-viewers-	statusNoDumpsCard2	0
-viewers-	statusReserved4Card1	0
-viewers-	statusReserved4Card2	0
-viewers-	statusERRACard1	0
-viewers-	statusERRACard2	0
-viewers-	statusERRBCard1	0
-viewers-	statusERRBCard2	0
-viewers-	statusERRCCard1	0
-viewers-	statusERRCCard2	0
-viewers-	statusLostFramesACard1	0
-viewers-	statusLostFramesACard2	51757055
-viewers-	statusLostFramesBCard1	0
-viewers-	statusLostFramesBCard2	51757055
-viewers-	statusStatusACard1	0
-viewers-	statusStatusACard2	0
-viewers-	statusStatusBCard1	0
-viewers-	statusStatusBCard2	0
-viewers-	statusReserved12Card1	0
-viewers-	statusReserved12Card2	0
-viewers-	statusReserved13Card1	0
-viewers-	statusReserved13Card2	112
-viewers-	statusReserved14Card1	10
-viewers-	statusReserved14Card2	4680
-viewers-	statusRevisionCard1	285803787
-viewers-	statusRevisionCard2	285803787

Figure 8-13: Error and Status panel at the Acquisition Navigator.

Figure 8-13 shows another snapshot of the Navigator interface. In this panel, it provides the information log from the Error and Status reporting feature seen previously in Section 8.3. It comprises the reports from both acquisition cards connected.

Figure 8-14 shows the panel that the user can use to request averaging or not of the data and provide the values to be used for the averaging. This feature can be used very creatively to guide the display of the data. For example, by dividing all values with the decimal 1024 then only the number of counts will be shown in the data and the 10-bit ADC will be trimmed out. Another example is to divide each value with the number of elements it holds shown in Table 6-2 of the Successive Running Sums configuration. Of course, its main use will be to normalise the data by dividing them with their threshold values.

Additionally, through this panel, the user can request an ASCII log file to be produced that includes all the data. The ability to choose the depth of this history recording is also given and each reading will include a UTC time-stamp added by the PowerPC.


Navigation Tool
File
Automate
BLM-PSB-Lab@ALL:Historical14-#13
BLM-PSB-Lab@ALL:Historical15-#14
BLM-PSB-Lab@ALL:Historical16-#15
BLM-PSB-Lab@ALL:Historical11-#10
BLM-PSB-Lab@ALL:Historical12-#11
BLM-PSB-Lab@ALL:Historical13-#12
BLM-PSB-Lab@ALL:Historical8-#7
BLM-PSB-Lab@ALL:Historical9-#8
BLM-PSB-Lab@ALL:Historical10-#9
BLM-PSB-Lab@ALL:Historical5-#4
BLM-PSB-Lab@ALL:Historical6-#5
BLM-PSB-Lab@ALL:Historical7-#6
BLM-PSB-Lab@ALL:Historical1-#0
BLM-PSB-Lab@ALL:Historical2-#1
BLM-PSB-Lab@ALL:Historical3-#2
BLM-PSB-Lab@ALL:Historical4-#3
BLM-PSB-Lab@ALL:Status-#16
BLM-PSB-Lab@ALL:Acquisition-#17
BLM-PSB-Lab@ALL:Setting-#18
Active Properties
Global Viewers
Property Value
-viewers-
-viewers-
-viewers-
-viewers-
-viewers-
Get
Get Next Published
Set
Unsubscribe

averageData: true
averagingFactors: array-long
acqLogsToGo: 0
loggingDirectory: /nfs/bdidev1/local/BL
acqLogFilename: NO LOGGING

Table view on BLM-PSB-Lab@ALL:Setting.averagingFactors {sequential=true}

Index	Value#1
1	0x1
2	0x2
3	0x8
4	0x10
5	0x40
6	0x100
7	0x800
8	0x2000
9	0x8000
10	0x20000
11	0x80000
12	0x200000

Displayed in base: 16

Figure 8-14: Averaging and Storing of Maximum Values History by the Navigator.

Concluding, the Navigator tool has provided a lot of information during the last stages of the development of the BLMTC system. It helped to test the BLMTC Logging process and discover problematic areas and not well-implemented functions. In the near future, it will additionally provide a very good starting point for the development of the complete GUI system for the Control Room.

8.7 SUMMARY

The BLMTC is one of the systems needed to provide data to both Post Mortem and Logging Systems because of its criticality to the Machine Protection System and its ability to trigger a Beam Abort.

The main objective for the Logging System is to continuously record and provide an online display to the Control Room of the machine status and show slow or infrequent changes, and for the Post Mortem system is to store enough data to allow the reconstruction of the event sequence that led to a beam or power abort.

From each BLMTC card, three sets of information will be prepared for the logging purposes: the Threshold Table's Values used, the Maximum Values of the Beam Loss Data, and the Error and Status reports.

The implementation proposed is taking into account the complete system in order to save resources and limit the unnecessary calculations that have a direct impact on the fitting and the power consumption of the FPGA. Additionally, each card will be able to synchronise independently with the PowerPC readout for the data calculation and update, a feature that will ensure that there is no loss of data if one or more readings for some reason are skipped. The PowerPC will have the ability to vary the reading interval without any change needed on the FPGA firmware. In other words, the implementation provides advantages that might be crucial for the system stability when the complete load will be applied in the crate CPU.

Finally, it provided specifications to the software team to implement a scaled down version of the Logging Navigator for the Control Room and helped in the debugging process by providing test vectors and verifying the outputs on various system setups. The functionalities of this system include historical view or plot of the Maximum Values data, storage of data, and normalisation of those data by user-defined tables.

REFERENCES

[89] LHC Project Note 303, Jorg Wenninger, *"The LHC Post-mortem System"*, 15-10-2002

[90] H. Burkhardt, "How to use beam loss monitors at the LHC", Proceedings of the Chamonix XI workshop and CERN SL/2001-003 (DI), 2001.

[91] R. Lauckner, "What Data Is Required to Understand Failures During LHC Operation?", Proceedings of the Chamonix XI LHC Workshop , 2001

[92] R.G. Beetham, J-C. Michelon, B. Puccio, J-B. Ribes, *"GPS Precision Timing at CERN"*, CERN SL-Note 98-050 (CO),
[online: http://sl.web.cern.ch/SL/publications/co98_050.pdf]

[93] B.G. Taylor, *"Timing Distribution at the LHC"*, Presented at the 8th Workshop on Electronics for LHC Experiments, LECC02, Colmar, 9 - 13 September 2002

[94] Engineering Specifications, *"BOBR, The Beam Synchronous Timing Receiver Interface for the Beam Observation System"* draft,
[online: http://ttc.web.cern.ch/TTC/BOBRspec.pdf]

[95] B.Frammery, *"The LHC Control System"*, International Conference on Accelerator and Large Experimental Physics Control Systems, ICALEPCS2005, Geneva, Switzerland.

[96] Implementation of BLM Acquisition Navigator by Stephen Jackson, unpublished.

[97] online: http://sl-div-bi-sw.web.cern.ch/sl-div-bi-sw/Activities/FEComSA/entry.htm

[98] A. Guerrero and S. Jackson, *"Common Templates and Organisation for CERN Beam Instrumentation Front-End Software Upgrade"*, International Conference on Accelerator and Large Experimental Physics Control Systems, ICALEPCS2003, Gyeongju, Korea

[99] A. Guerrero, J-J Gras, J-L Nougaret, M. Ludwig, M. Arruat, S. Jackson, *"CERN Front-End Software Architecture for Accelerator Controls"*, International Conference on Accelerator and Large Experimental Physics Control Systems, ICALEPCS2003, Gyeongju, Korea

[100] Sun Microsystems, Inc. "The Java programming language",
[online: http://java.com/en/about/]

Conclusions

Conclusions

The real-time data analysis and decision system for particle flux detection in the LHC accelerator has been specified, designed and tested. However, those were not the only contributions made on the Beam Loss Monitoring (BLM) system. It was necessary to define and in some cases even construct parts of this system in order to ensure the best liaison of the measurement with the processing parts as well as to guarantee the availability of advantages needed to allow the successful implementation of the analysis and decision system.

For those reasons, the work began with an evaluation of the processing module and various proposals have been made to increase its versatility. Those included the addition of two more access points, with direct connection to the FPGA pins, and the ability to exchange FPGA devices with different densities by using a layout design process usually referred to as Vertical Migration.

In addition, a physical data link was constructed for the communication between the acquisition and the processing installations with reliability and availability as primary targets. It included redundancy and very careful choice of radiation tolerant devices together with the employment of the combination of two powerful algorithms for encoding the data that augmented not only the reception but also the error discovery capabilities.

A BLM mezzanine card was constructed that provided a practical solution for the reception of the data, low latency, through its gigabit data range, and exhibited not only the required functionality but allows simple future maintenance and upgradeability.

Proposals have also been made for the design of the tunnel electronics. Those included an acquisition strategy able to provide high accuracy and avoid errors, the collection of extra information to supervise its correct operation and the formatting definition of the transmitted packets.

For the processing of the BLM data most of the features available by an FPGA device have been used together with design methods that provided an optimal resources utilisation. The FPGA design was partitioned in logic blocks and each block has been designed, optimised and tested independently. Using this design methodology, not only the design and test cycles were less complex and time consuming but also specific optimisations have been applied to each of them, as well as, more effort was given to those that were more demanding in order to achieve higher global efficiency.

The Current-to-Frequency conversion (CFC) and ADC data acquired for each detector signal are different and a pre-processing was realised that merged them into one value. The measurement of the pulses, produced by the CFC, using a counter relates to the average current induced from the previous acquisition and the voltage measured, by the ADC, is the fraction of the charge in the capacitor. In addition, in this process was included a simple but efficient way to cut-off the noise introduced through the ADC, as well as, a function to classify the received data in fast or slow losses and apply different merging methods on them. Therefore, it will provide, to the later main processing and decision stages, accurate and compact data.

Thereafter, the system was realised to be able of processing the 16 channels in parallel by producing and keeping continuously updated 192 data histories. The histories are kept with the use of shift registers implemented in the embedded memory blocks. The shift registers used provide intermediate outputs, a feature effectively used to combine overlapping memory contents. Consecutively, the outputs have been used to create an equal number of Running Sums where the number of data values held or the integration time periods they signify could be modified simply by changing the parameters of the shift registers.

Moreover, the very long integration time periods requested by the specifications were constructed without the use of the less reliable external memories or the need for intensive computation. They are made to fit in the device available employing one more scheme of resource sharing, named Successive Running Sums (SRS). The SRS makes use of the already calculated sums in order to calculate bigger in length running sums, minimising in this way the memory storage needed for the histories. The only expense of some additional latency, coming from the refreshing delay, has insignificant contribution to the approximation accuracy since it is always a small fraction of the integration time under calculation. The achievement could be shown in a comparison of this system with a configuration similar to those used in other accelerators. Using the different configuration, the Shift Registers, would need to hold approximately 3 million values for each of the 16 detectors (a total of 150MB) to achieve the same approximation error. Instead, by using the SRS technique the system is now using only some of the FPGA internal memory since it does not need more than 100Kbytes for the whole system.

The resource utilisation for the comparison of the calculated Running Sum values with their corresponding threshold values was significantly reduced by using the same comparators more than once. In this case, the design is taking advantage of the specific implementation of the SRS process and the time difference in the arrival of the Running Sum values.

In the same process, the task of providing the threshold values for the given beam energy was also greatly simplified without losing any of its functionality by adding unique threshold tables for each of the 4000 detectors. On the contrary the additional advantage emerged from this strategy was that the tables can now include also an offset and a calibration factor.

In the Logging Recording, each crate CPU, which has the task of accessing the logging data from 16 cards, will be allowed by each processing module to synchronise independently the readout for the data calculations and their update, a feature that will ensure that there is no loss of data if one or more readings for some reason are skipped. More importantly the CPU will have the ability to vary the reading interval without any change needed on the FPGA firmware. Thus, it provides advantages that might be crucial later for the Logging system stability.

Nevertheless, the greatest advantage of the design methods used were discovered much later though, when similar systems with different parameters, like number of channels or integration time periods, were necessary to be produced. The redesign was simply to accommodate the changes in the blocks concerned and the test cycle was usually confined only to those changed, an advantage that might be very valuable at the start-up of the LHC when unexpected specification changes may be needed and a different configuration might provide better results.

Finally, the system implemented will exhibit all the necessary functionality and provide the requested crucial protection to the LHC accelerator. Since several months, two BLM systems are collecting successfully data. They are installed in CERN’s PS Booster pre-accelerator and in DESY’s HERA storage ring. Both systems show in their long term run the same satisfying features already seen the laboratory.

APPENDIX

Appendix A. Implementation of the BLMTC..............185
Appendix B. BLM Mezzanine Schematic Drawings....194
Appendix C. BLMTC Versions......................................196
Appendix D. Linearity Test of the BLM system...........202

Appendix A. Implementation of the BLMTC

A.1 Design Report – RCC (Receive, Check and Compare) Process

A.1.1 VHDL CODE FOR CRC CHECK

```
-----------------------------------------------------------------
-- File:       CRC_Check_v2.vhd
-- Description: CRC combinatorial calculation in one clock cycle
-- Designer:    Christos Zamantzas
-- Date:        Fri 04 Aug 2005 rev.10
-----------------------------------------------------------------

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

-- Entity declaration

ENTITY CRC_Check_v2 IS
PORT
(
CLK         : IN STD_LOGIC;
--Status: 0x not valid D, 1x valid D, x0 not SOF, x1 SOF
status      : IN STD_LOGIC_VECTOR(1 downto 0);
RESET       : IN STD_LOGIC;
D           : IN STD_LOGIC_VECTOR(15 downto 0);
calcCRC     : OUT STD_LOGIC_VECTOR(31 downto 0);
CRCout      : OUT STD_LOGIC_VECTOR(31 downto 0);
D16out      : OUT STD_LOGIC_VECTOR(15 downto 0);
count       : OUT STD_LOGIC_VECTOR(3 downto 0);
DValid      : OUT STD_LOGIC; -- =1 when not in IDLE state
CRCready    : OUT STD_LOGIC; -- =1 when CRCout ready for compare
ERR         : OUT STD_LOGIC; -- =1 when different CRC was expected
CFCstatus   : OUT STD_LOGIC; -- =1 when error detected in the tunnel(HT,PS,or CFC)
SOF         : OUT STD_LOGIC  -- =1 Indicates start of frame
);

END CRC_Check_v2;

-- Architecture declaration

ARCHITECTURE CRC_CHECK_architecture OF CRC_Check_v2 IS

-- Function declaration

function nextCRC
    ( Data:  std_logic_vector(15 downto 0);
      CRC:   std_logic_vector(31 downto 0) )
    return std_logic_vector is

    variable D: std_logic_vector(15 downto 0);
    variable C: std_logic_vector(31 downto 0);
    variable NewCRC: std_logic_vector(31 downto 0);

  begin

    D := Data;
    C := CRC;

    NewCRC(0)  := D(12) xor D(10) xor D(9) xor D(6) xor D(0) xor C(16) xor
                 C(22) xor C(25) xor C(26) xor C(28);
    NewCRC(1)  := D(13) xor D(12) xor D(11) xor D(9) xor D(7) xor D(6) xor
                 D(1) xor D(0) xor C(16) xor C(17) xor C(22) xor C(23) xor
                 C(25) xor C(27) xor C(28) xor C(29);
    NewCRC(2)  := D(14) xor D(13) xor D(9) xor D(8) xor D(7) xor D(6) xor
                 D(2) xor D(1) xor D(0) xor C(16) xor C(17) xor C(18) xor
                 C(22) xor C(23) xor C(24) xor C(25) xor C(29) xor C(30);
    NewCRC(3)  := D(15) xor D(14) xor D(10) xor D(9) xor D(8) xor D(7) xor
                 D(3) xor D(2) xor D(1) xor C(17) xor C(18) xor C(19) xor
                 C(23) xor C(24) xor C(25) xor C(26) xor C(30) xor C(31);
    NewCRC(4)  := D(15) xor D(12) xor D(11) xor D(8) xor D(6) xor D(4) xor
                 D(3) xor D(2) xor D(0) xor C(16) xor C(18) xor C(19) xor
                 C(20) xor C(22) xor C(24) xor C(27) xor C(28) xor C(31);
    NewCRC(5)  := D(13) xor D(10) xor D(7) xor D(6) xor D(5) xor D(4) xor
                 D(3) xor D(1) xor D(0) xor C(16) xor C(17) xor C(19) xor
                 C(20) xor C(21) xor C(22) xor C(23) xor C(26) xor C(29);
    NewCRC(6)  := D(14) xor D(11) xor D(8) xor D(7) xor D(6) xor D(5) xor
```

```
                 D(4) xor D(2) xor D(1) xor C(17) xor C(18) xor C(20) xor
                 C(21) xor C(22) xor C(23) xor C(24) xor C(27) xor C(30);
    NewCRC(7) := D(15) xor D(10) xor D(8) xor D(7) xor D(5) xor D(3) xor
                 D(2) xor D(0) xor C(16) xor C(18) xor C(19) xor C(21) xor
                 C(23) xor C(24) xor C(26) xor C(31);
    NewCRC(8) := D(12) xor D(11) xor D(10) xor D(8) xor D(4) xor D(3) xor
                 D(1) xor D(0) xor C(16) xor C(17) xor C(19) xor C(20) xor
                 C(24) xor C(26) xor C(27) xor C(28);
    NewCRC(9) := D(13) xor D(12) xor D(11) xor D(9) xor D(5) xor D(4) xor
                 D(2) xor D(1) xor C(17) xor C(18) xor C(20) xor C(21) xor
                 C(25) xor C(27) xor C(28) xor C(29);
    NewCRC(10) := D(14) xor D(13) xor D(9) xor D(5) xor D(3) xor D(2) xor
                  D(0) xor C(16) xor C(18) xor C(19) xor C(21) xor C(25) xor
                  C(29) xor C(30);
    NewCRC(11) := D(15) xor D(14) xor D(12) xor D(9) xor D(4) xor D(3) xor
                  D(1) xor D(0) xor C(16) xor C(17) xor C(19) xor C(20) xor
                  C(25) xor C(28) xor C(30) xor C(31);
    NewCRC(12) := D(15) xor D(13) xor D(12) xor D(9) xor D(6) xor D(5) xor
                  D(4) xor D(2) xor D(1) xor D(0) xor C(16) xor C(17) xor
                  C(18) xor C(20) xor C(21) xor C(22) xor C(25) xor C(28) xor
                  C(29) xor C(31);
    NewCRC(13) := D(14) xor D(13) xor D(10) xor D(7) xor D(6) xor D(5) xor
                  D(3) xor D(2) xor D(1) xor C(17) xor C(18) xor C(19) xor
                  C(21) xor C(22) xor C(23) xor C(26) xor C(29) xor C(30);
    NewCRC(14) := D(15) xor D(14) xor D(11) xor D(8) xor D(7) xor D(6) xor
                  D(4) xor D(3) xor D(2) xor C(18) xor C(19) xor C(20) xor
                  C(22) xor C(23) xor C(24) xor C(27) xor C(30) xor C(31);
    NewCRC(15) := D(15) xor D(12) xor D(9) xor D(8) xor D(7) xor D(5) xor
                  D(4) xor D(3) xor C(19) xor C(20) xor C(21) xor C(23) xor
                  C(24) xor C(25) xor C(28) xor C(31);
    NewCRC(16) := D(13) xor D(12) xor D(8) xor D(5) xor D(4) xor D(0) xor
                  C(0) xor C(16) xor C(20) xor C(21) xor C(24) xor C(28) xor
                  C(29);
    NewCRC(17) := D(14) xor D(13) xor D(9) xor D(6) xor D(5) xor D(1) xor
                  C(1) xor C(17) xor C(21) xor C(22) xor C(25) xor C(29) xor
                  C(30);
    NewCRC(18) := D(15) xor D(14) xor D(10) xor D(7) xor D(6) xor D(2) xor
                  C(2) xor C(18) xor C(22) xor C(23) xor C(26) xor C(30) xor
                  C(31);
    NewCRC(19) := D(15) xor D(11) xor D(8) xor D(7) xor D(3) xor C(3) xor
                  C(19) xor C(23) xor C(24) xor C(27) xor C(31);
    NewCRC(20) := D(12) xor D(9) xor D(8) xor D(4) xor C(4) xor C(20) xor
                  C(24) xor C(25) xor C(28);
    NewCRC(21) := D(13) xor D(10) xor D(9) xor D(5) xor C(5) xor C(21) xor
                  C(25) xor C(26) xor C(29);
    NewCRC(22) := D(14) xor D(12) xor D(11) xor D(9) xor D(0) xor C(6) xor
                  C(16) xor C(25) xor C(27) xor C(28) xor C(30);
    NewCRC(23) := D(15) xor D(13) xor D(9) xor D(6) xor D(1) xor D(0) xor
                  C(7) xor C(16) xor C(17) xor C(22) xor C(25) xor C(29) xor
                  C(31);
    NewCRC(24) := D(14) xor D(10) xor D(7) xor D(2) xor D(1) xor C(8) xor
                  C(17) xor C(18) xor C(23) xor C(26) xor C(30);
    NewCRC(25) := D(15) xor D(11) xor D(8) xor D(3) xor D(2) xor C(9) xor
                  C(18) xor C(19) xor C(24) xor C(27) xor C(31);
    NewCRC(26) := D(10) xor D(6) xor D(4) xor D(3) xor D(0) xor C(10) xor
                  C(16) xor C(19) xor C(20) xor C(22) xor C(26);
    NewCRC(27) := D(11) xor D(7) xor D(5) xor D(4) xor D(1) xor C(11) xor
                  C(17) xor C(20) xor C(21) xor C(23) xor C(27);
    NewCRC(28) := D(12) xor D(8) xor D(6) xor D(5) xor D(2) xor C(12) xor
                  C(18) xor C(21) xor C(22) xor C(24) xor C(28);
    NewCRC(29) := D(13) xor D(9) xor D(7) xor D(6) xor D(3) xor C(13) xor
                  C(19) xor C(22) xor C(23) xor C(25) xor C(29);
    NewCRC(30) := D(14) xor D(10) xor D(8) xor D(7) xor D(4) xor C(14) xor
                  C(20) xor C(23) xor C(24) xor C(26) xor C(30);
    NewCRC(31) := D(15) xor D(11) xor D(9) xor D(8) xor D(5) xor C(15) xor
                  C(21) xor C(24) xor C(25) xor C(27) xor C(31);
    return NewCRC;

  end nextCRC;

--------------------------------------------------------------------------
-- Main Program
--------------------------------------------------------------------------

signal   Cnt      :integer range 0 to 15;

begin

CRC_parallel : process( CLK, RESET )

variable D16      :std_logic_vector(15 downto 0);
variable oldD16   :std_logic_vector(15 downto 0);
variable CRC32    :std_logic_vector(31 downto 0);
variable CRCrx    :std_logic_vector(31 downto 0);
variable S        :std_logic_vector(1 downto 0);
variable WordCnt  :integer range 0 to 15;
```

```
begin
   if RESET='1'
      then
         CRC32:=(others => '1');
         WordCnt:=0;
         DValid<='0';
         CRCready<='0';
         ERR<='0';
         SOF<='0';
         CFCstatus<='0';
      elsif CLK'event and CLK='1'
         then
            S:=status(1 downto 0);
            D16:=D(15 downto 0);
            case S is
               when "00" =>
                        WordCnt:=0;
                        DValid<='0';              -- IDLE
                        SOF<='0';

               when "01" =>
                        CRC32:=(others => '1');
                        WordCnt:=1;
                        DValid<='0';              -- SOF (NOT IDLE)
                        SOF<='1';
                        CRCready<='0';
                        ERR<='0';
               when "10" =>
                        DValid<='0';              -- Data (NOT IDLE)
                        SOF<='0';
                        if WordCnt>0 then
                          if WordCnt<15 then
                            DValid<='1';
                          end if;
                          case WordCnt is

                            ---------- check CRC --------------
                                 when 14 => -- CRC's 1st part
                                    oldD16:=D16;
                                 when 15 => -- CRC's 2nd part
                                    CRCrx:=oldD16 & D16;
                                    calcCRC<=CRCrx; -- out for comparison
                                    CRCready<='1';
                                    ERR<='0';

                                    if (CRC32 /= CRCrx) then
                                       ERR<='1';
                                    end if;
                            ---------- check CRC end --------------

                            ---------- check Status --------------
                                  when 4 =>
                                        if   (( D16(15)
                                             or D16(14)
                                             or D16(13)
                                             or D16(12)
                                             or D16(11)
                                             or D16(10)
                                             or D16(9)
                                             or D16(8)
                                             or D16(7)
                                             or D16(6)
                                             ) = '1')
                                        then CFCstatus<='1';
                                        else CRC32:=nextCRC(D16,CRC32);
                                        end if;
                             ---------- check status end --------------

                                  when others =>
                                    CRC32:=nextCRC(D16,CRC32);
                          end case;
                        WordCnt:=WordCnt+1;
                        end if;
               when others =>
                        CRC32:=(others => '1');
                        WordCnt:=0;
                        DValid<='0';
                        ERR<='1';
                        SOF<='0';
                        CFCstatus<='1';
            end case;
   end if;
```

```
--------- Outputs --------

   CRCout<=CRC32;
   D16out<=D16;
   Cnt <= WordCnt-1;

end process CRC_parallel;

process (clk)   --------- Provide Write Address Signal for Buffer -------
begin
   if CLK'event and CLK='1' then
      Count<=std_logic_vector(to_signed((Cnt),4));
   end if;
end process;

end CRC_CHECK_architecture;
```

A.2 Design Report – Data-Combine Process

A.2.1 VHDL CODE FOR DATA-COMBINE

```
------------------------------------------------------------------------
--  Data-Combine  -----------------------------------------------------
------------------------------------------------------------------------
-- this process will combine the data of the ADC and the counts using  -
-- minimum ADC value hold circuit to cutoff the noise seen in the ADC  -
------------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

ENTITY DataCombibing IS

PORT(Din     : IN    STD_LOGIC_VECTOR(19 DOWNTO 0);
     reset   : IN    STD_LOGIC;
     clken   : IN    STD_LOGIC;
     clk     : IN    STD_LOGIC;
     Dout    : OUT   STD_LOGIC_VECTOR(19 DOWNTO 0);
     clken1  : OUT   STD_LOGIC
    );

END DataCombibing;

ARCHITECTURE RTL OF DataCombibing IS
    Signal Din_ADC_old : STD_LOGIC_VECTOR(9 DOWNTO 0);
    Signal Din_ADC_new : STD_LOGIC_VECTOR(9 DOWNTO 0);
    Signal Din_Counts  : STD_LOGIC_VECTOR(17 DOWNTO 0);
    Signal Dout_ADC    : STD_LOGIC_VECTOR(9 DOWNTO 0);
    Signal Dout_int    : STD_LOGIC_VECTOR(17 DOWNTO 0);
    Signal clken1_int  : STD_LOGIC;
BEGIN

PROCESS (clk, reset)
    Variable Din_ADC_min : STD_LOGIC_VECTOR(9 DOWNTO 0);
    Variable Dout_ADC    : STD_LOGIC_VECTOR(9 DOWNTO 0);

BEGIN

    IF  rising_edge(clk) THEN
        IF reset = '1' THEN
          Dout_int <= (others =>'0');
          Din_ADC_min := (others =>'1');
        ELSIF clken = '1' THEN
          Din_ADC_old <= Din_ADC_new;
          Din_ADC_new <= Din(11 DOWNTO 2);
          Din_Counts <= Din(19 DOWNTO 12) & "0000000000";

          IF Din_Counts = "000000000000000000" THEN
            IF Din_ADC_min > Din_ADC_old OR
               Din_ADC_min = "0000000000" THEN
               Din_ADC_min := Din_ADC_old;
            END IF;
            IF Din_ADC_min < Din_ADC_new THEN
               Dout_int <= (others =>'0');
            ELSE
               Dout_ADC := Din_ADC_min - Din_ADC_new;
               Dout_int <= Din_Counts + Dout_ADC;
            END IF;

          ELSE
            Dout_ADC := Din_ADC_min - Din_ADC_new;
            Din_ADC_min := Din_ADC_new;
            IF (Din_ADC_old < Din_ADC_new) THEN
              Dout_int <= Din_Counts - "10000000000" + Dout_ADC;
            ELSE
              Dout_int <= Din_Counts + Dout_ADC;
            END IF;
          END IF;
        END IF;
    END IF;
END PROCESS;

PROCESS (clk, reset)
BEGIN
```

```
IF  rising_edge(clk) THEN
    clken1_int <= clken;
END IF;

END PROCESS;

Dout <= "00" & Dout_int(17 DOWNTO 0);
clken1 <= clken1_int;

END RTL;
```

A.3 Design Report – Masking Process

A.3.1 VHDL CODE FOR THE INHIBIT FUNCTION

```
-------------------------------------------------------------------
-- File:    Inhibit12.vhd
-- Description:   At the last stage of the BLM, the output of some
--                detectors (non-critical) could be inhibited.
--
-- Designer:   Christos Zamantzas
-- Date:       Tue 25 Jan 2005 rev.2
-------------------------------------------------------------------

LIBRARY ieee;
USE ieee.std_logic_1164.all;


ENTITY Inhibit12 IS
PORT (CLK         : IN STD_LOGIC;
      En_Inhibit  : IN STD_LOGIC;
      Din         : IN STD_LOGIC_VECTOR(11 downto 0);
      Value       : IN STD_LOGIC_VECTOR(11 downto 0);
      NonCritical_V : OUT STD_LOGIC_VECTOR(11 downto 0);
      NonCritical : OUT STD_LOGIC;
      Critical_V    : OUT STD_LOGIC_VECTOR(11 downto 0);
      Critical  : OUT STD_LOGIC);

END Inhibit12;

ARCHITECTURE RTL OF Inhibit12 IS

signal Cint :  STD_LOGIC_VECTOR(11 downto 0);
signal NCint :  STD_LOGIC_VECTOR(11 downto 0);

begin
   process (clk,En_Inhibit)
   begin
      if rising_edge(clk) then

         Cint <= Din;
         NCint <= (others => '0');

         if En_Inhibit= '1' then
               Cint <= Din and Value;
               NCint <= Din and not Value;
         end if;

      end if;
   end process;
   NonCritical_V <= NCint;
   NonCritical <= NCint(0) or
                  NCint(1) or
                  NCint(2) or
                  NCint(3) or
                  NCint(4) or
                  NCint(5) or
                  NCint(6) or
                  NCint(7) or
                  NCint(8) or
                  NCint(9) or
                  NCint(10) or
                  NCint(11) ;

   Critical_V <= Cint;
   Critical <= Cint(0) or
               Cint(1) or
               Cint(2) or
               Cint(3) or
               Cint(4) or
               Cint(5) or
               Cint(6) or
               Cint(7) or
               Cint(8) or
               Cint(9) or
               Cint(10) or
               Cint(11) ;

end;
```

A.4 Design Report –Logging Process

A.4.1 VHDL CODE FOR THE ONESHOT CIRCUIT

```
------------------------------------------------------------------
-- File:          oneshot.vhd
-- Description:   The output will go high for only one clock cycle.
-- Designer:      Christos Zamantzas
-- Date:          03/12/2004 rev.3
------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

ENTITY OnePulse IS

    PORT(Din, clk   : IN    STD_LOGIC;
         SP         : OUT   STD_LOGIC
         );
END OnePulse;

ARCHITECTURE RTL OF onepulse IS

SIGNAL Din_delay : STD_LOGIC;

BEGIN
 PROCESS (clk)
 BEGIN
    IF  rising_edge(clk) THEN
        IF Din = '1' AND Din_delay = '0' THEN
            SP <= '1';
        ELSE
            SP <= '0';
        END IF;
    Din_delay <= Din;
    END IF;


 END PROCESS;

END RTL;
```

A.4.2 VHDL CODE FOR CALCULATING THE MAXIMUM VALUE

```
---------------------------------------------------------------------
-- File:          max_20bit.vhd
-- Description:   Max value calculation of the input for 20-bit data
-- Designer:      Christos Zamantzas
-- Date:          03/12/2004 rev.3
---------------------------------------------------------------------

LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY Max_20bit IS
   port
   (
      data   :  IN  STD_LOGIC_VECTOR(19 downto 0);
      clk    :  IN  STD_LOGIC;
      enable :  IN  STD_LOGIC;
      reset  :  IN  STD_LOGIC;
      max    :  OUT  STD_LOGIC_VECTOR(19 downto 0)
   );
END Max_20bit;

ARCHITECTURE Structure OF Max_20bit IS

component max_compare_20bits
   PORT(dataa : IN STD_LOGIC_VECTOR(19 downto 0);
       datab : IN STD_LOGIC_VECTOR(19 downto 0);
       agb : OUT STD_LOGIC
   );
end component;

signal   en_max  :  STD_LOGIC;
signal   greater :  STD_LOGIC;
signal   max_int :  STD_LOGIC_VECTOR(19 downto 0);

BEGIN

instmax20 : max_compare_20bits
PORT MAP(dataa => data,
       datab => max_int,
       agb => greater);


process(clk,reset)
begin
if (reset = '1') then
   max_int <= (others => '0');

elsif (rising_edge(clk)) then
   if (en_max = '1') then
   max_int <= data;
   end if;
end if;
end process;

en_max <= greater AND enable;
max <= max_int;

END Structure;
```

Appendix B. BLM Mezzanine Schematic Drawings

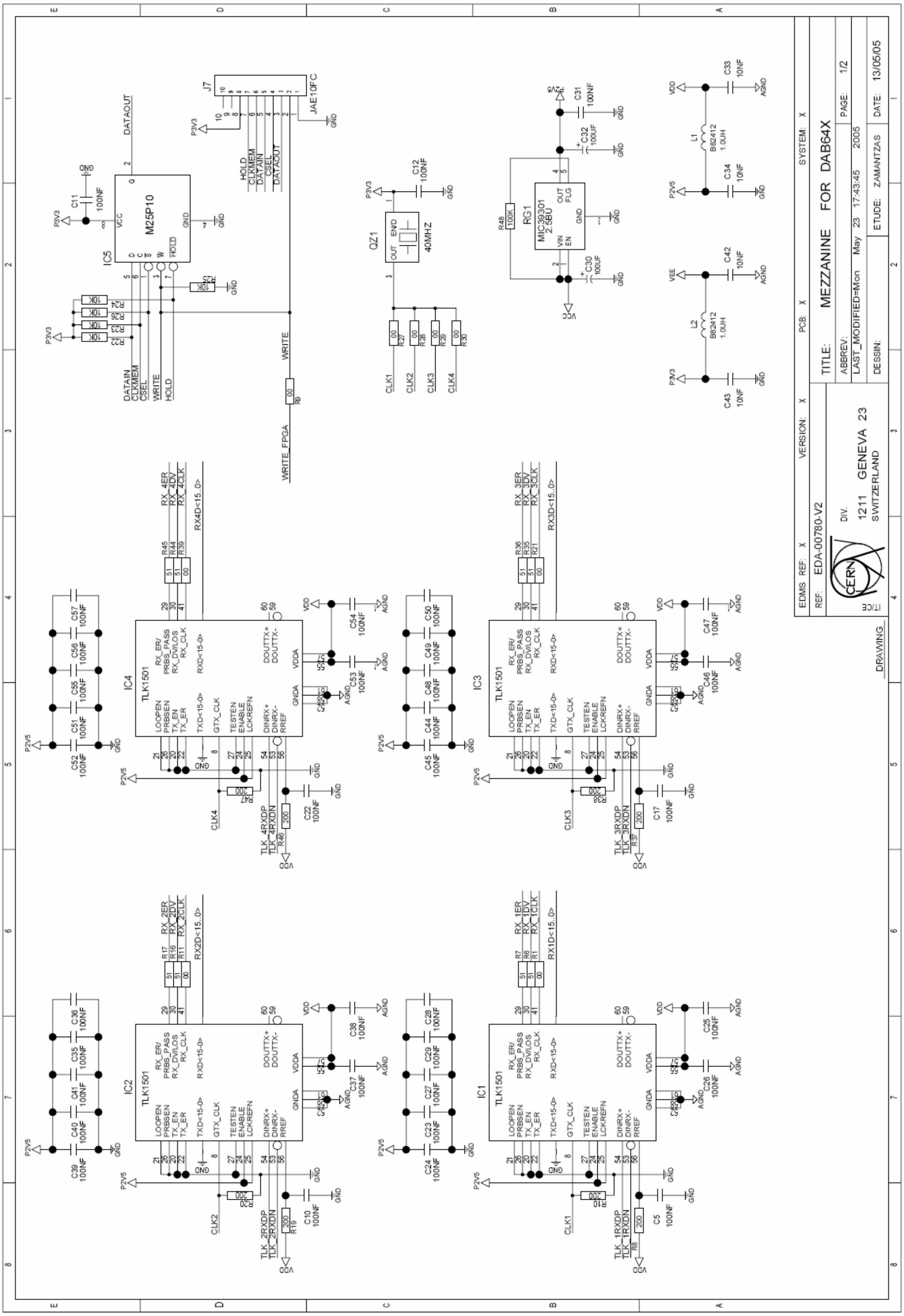


Figure B-1: BLM Mezzanine Schematic Drawing (page 1 of 2).

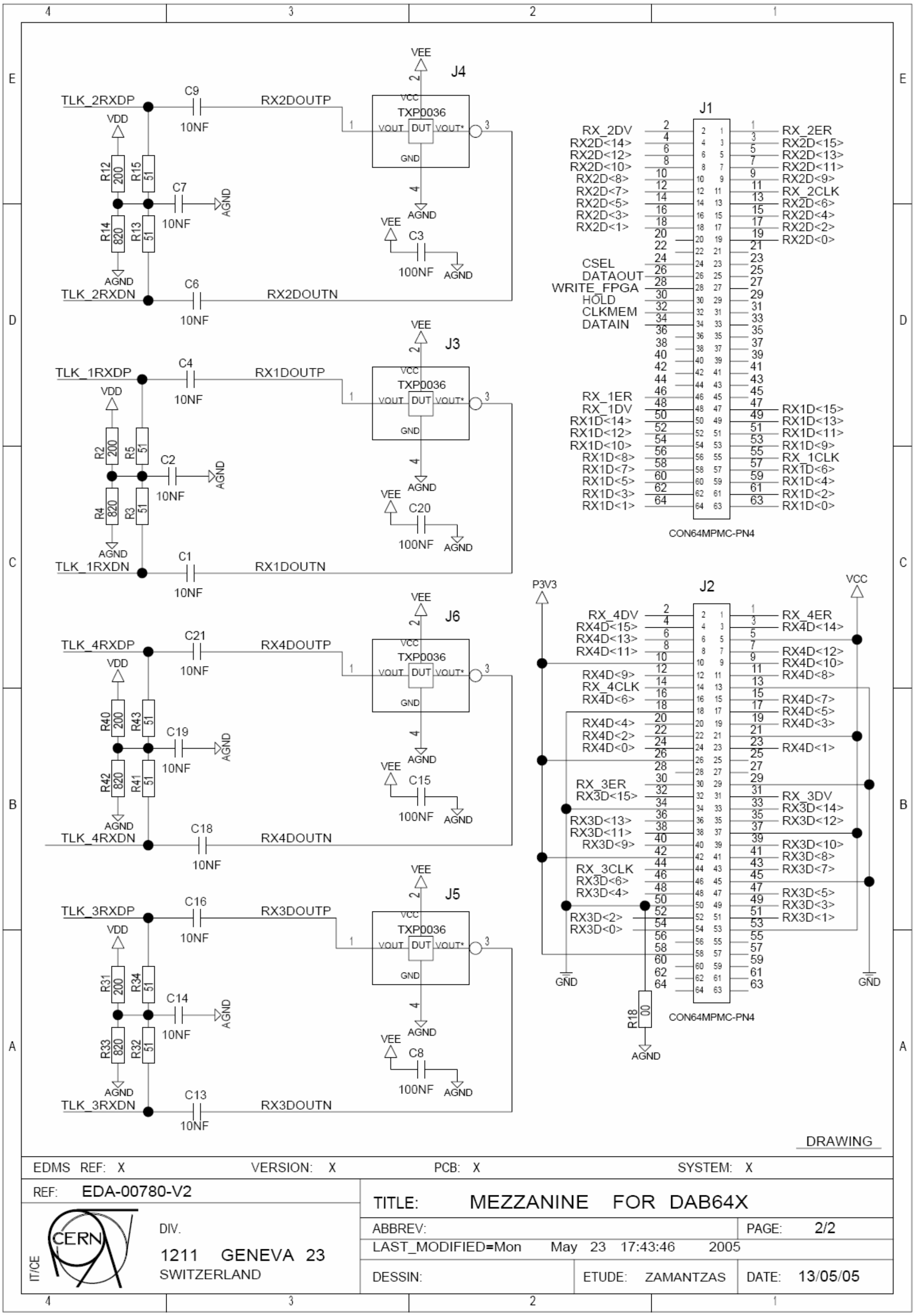


Figure B-2: BLM Mezzanine Schematic Drawing (page 2 of 2).

Appendix C. BLMTC Versions

Since the LHC is still under construction, the BLM system was decided to be installed in other available accelerators. Two places were easily accessible. The first one was in the beam dump area of the HERA storage ring at DESY in Hamburg, Germany and the second in one of the pre-accelerators of LHC the PS Booster.

These efforts were done mainly to provide real data to be compared with those expected from simulations of the conversion ratio of the particle flux to electrical signal and define with higher accuracy the threshold levels.

For both of these installations, the observation windows were not necessary to span up to 100 seconds. Especially in the case of DESY, where the detectors have been placed around its dump block, the system would record short but with high amplitude losses whenever the beam was extracted.

Thus, the BLMTC was constructed with a different configuration for the Successive Running Sums that provided integration time up to 3.2 seconds but with better granularity. That can also be seen as an example of the great flexibility the implementation chosen can provide. This configuration of the SRS can be seen in Table C-1.

Finally, the Top-Level schematic design file for each installation's implementation can be seen in Figures C-1 and C-2 and their resource utilisation in Tables C-2 and C-3. The PS Booster's implementation uses a smaller FPGA device, the EP1S20, and processes data only for eight detectors.

C.1 BLMTC for the PS Booster at CERN

Table C-1: Successive Running Sum Configuration for the PS Booster (including unused outputs).

	Moving Windows		Refreshing [steps]	Single channel length	Shift Register Name	Signal Name
	40µs steps	ms				
System A (0.4ms – 10ms)	1	0.04	1	1		RS0
	2	0.08	1	2		RS1
	4	**0.16**	**1**	**4**	**SR1**	**N/C 1**
	6	**0.24**	**1**	**6**		**N/C 2**
	8	0.32	1	8		RS2
	16	0.64	1	16		RS3
	64	2.56	2	32	**SR2**	RS4
	128	**5.12**	**2**	**64**		**N/C 3**
	192	**7.68**	**2**	**96**		**N/C 4**
	256	10.24	2	128		RS5
System B (41ms – 3.2s)	1024	40.96	64	16	**SR3**	RS6
	2048	**81.92**	**64**	**32**		**N/C 5**
	3072	**122.88**	**64**	**48**		**N/C 6**
	4096	163.84	64	64		RS7
	16384	655.36	1024	16	**SR4**	RS8
	32768	1310.72	1024	32		RS9
	49152	**1966.08**	**1024**	**48**		**N/C 7**
	65536	2621.44	1024	64		RS10
	81920	3276.80	1024	80		RS11
	163840	**6553.60**	**1024**	**96**		**N/C 8**
	245760	**9830.4**	**1024**	**112**		**N/C 9**
	327680	**13107.2**	**1024**	**128**		**N/C 10**

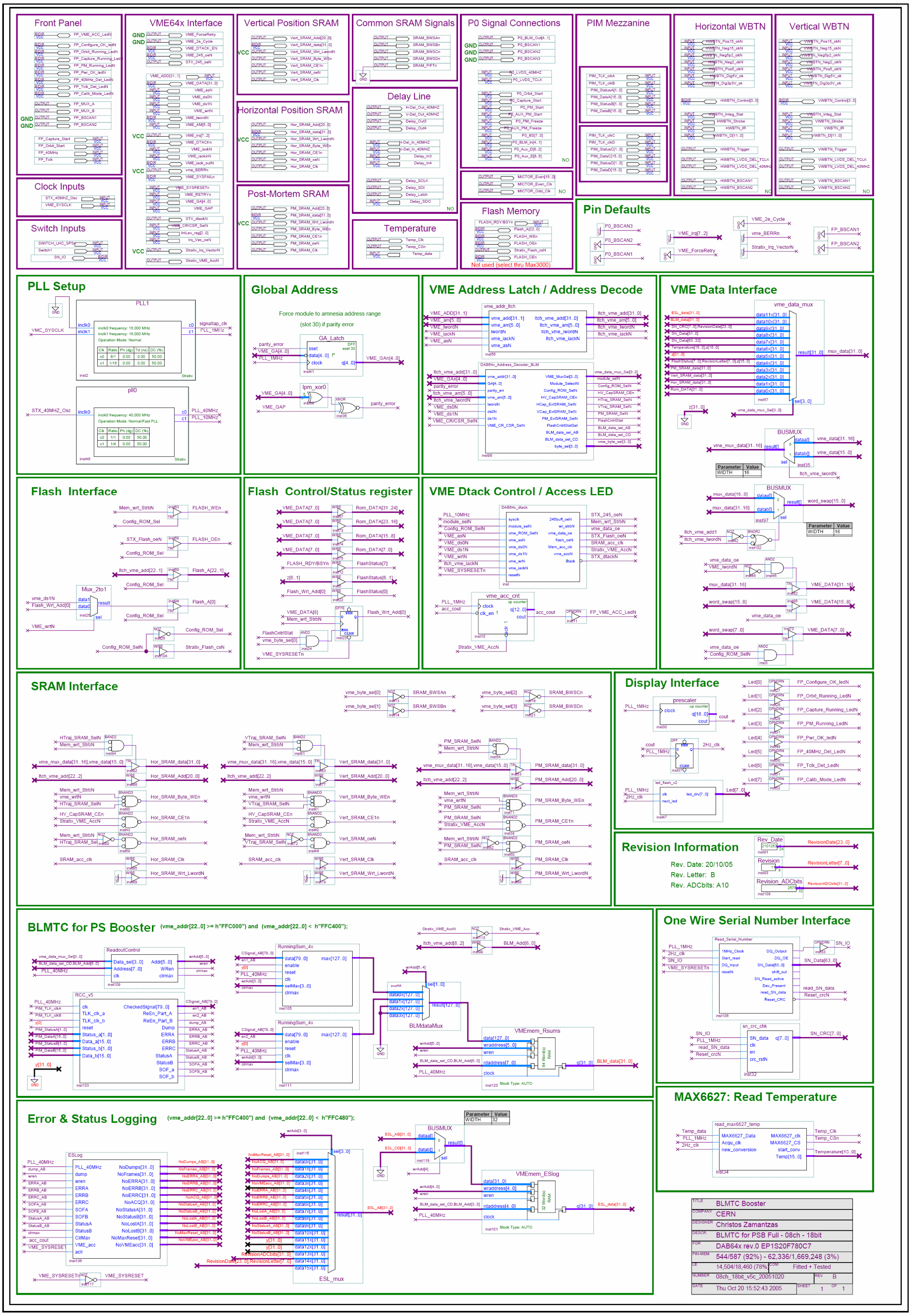
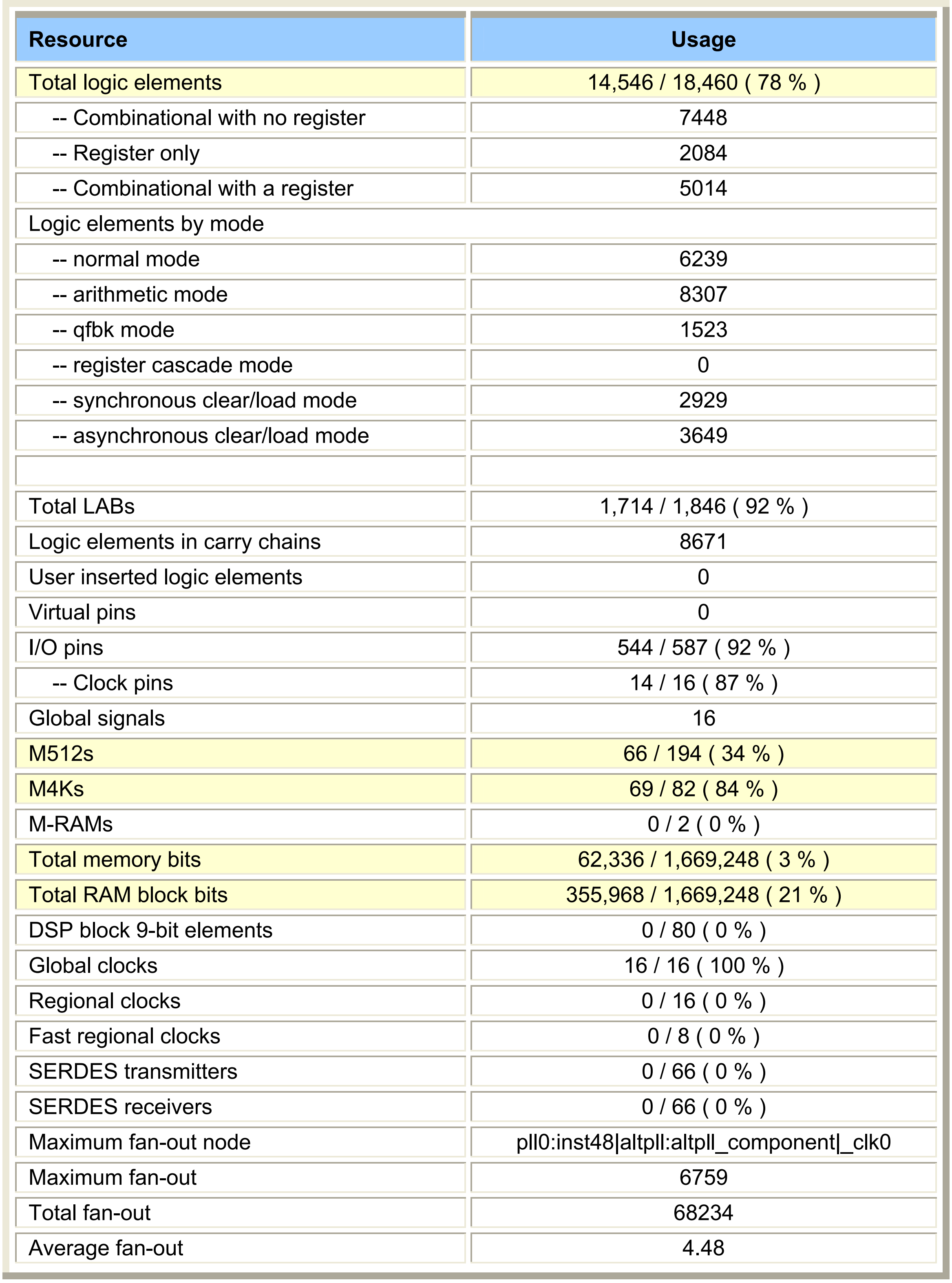


Figure C-1: Top-Level Schematic of BLMTC for the PS Booster Installation.

Table C-2: Resource Utilisation of BLMTC System for the PS Booster Installation.



Resource	Usage
Total logic elements	14,546 / 18,460 (78 %)
-- Combinational with no register	7448
-- Register only	2084
-- Combinational with a register	5014
Logic elements by mode	
-- normal mode	6239
-- arithmetic mode	8307
-- qfbk mode	1523
-- register cascade mode	0
-- synchronous clear/load mode	2929
-- asynchronous clear/load mode	3649
Total LABs	1,714 / 1,846 (92 %)
Logic elements in carry chains	8671
User inserted logic elements	0
Virtual pins	0
I/O pins	544 / 587 (92 %)
-- Clock pins	14 / 16 (87 %)
Global signals	16
M512s	66 / 194 (34 %)
M4Ks	69 / 82 (84 %)
M-RAMs	0 / 2 (0 %)
Total memory bits	62,336 / 1,669,248 (3 %)
Total RAM block bits	355,968 / 1,669,248 (21 %)
DSP block 9-bit elements	0 / 80 (0 %)
Global clocks	16 / 16 (100 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 8 (0 %)
SERDES transmitters	0 / 66 (0 %)
SERDES receivers	0 / 66 (0 %)
Maximum fan-out node	pll0:inst48|altpll:altpll_component|_clk0
Maximum fan-out	6759
Total fan-out	68234
Average fan-out	4.48

C.2 BLMTC for the HERA at DESY

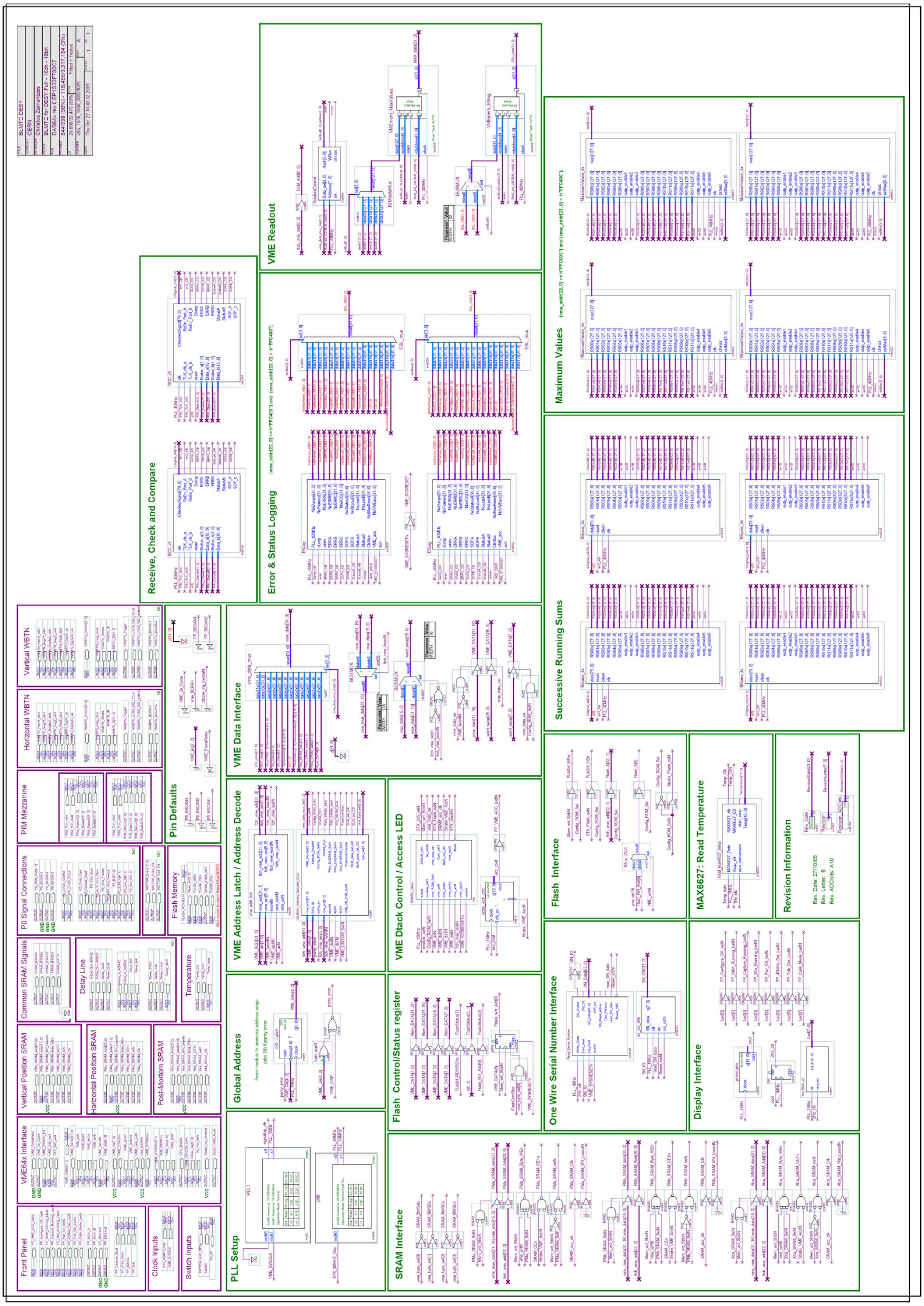
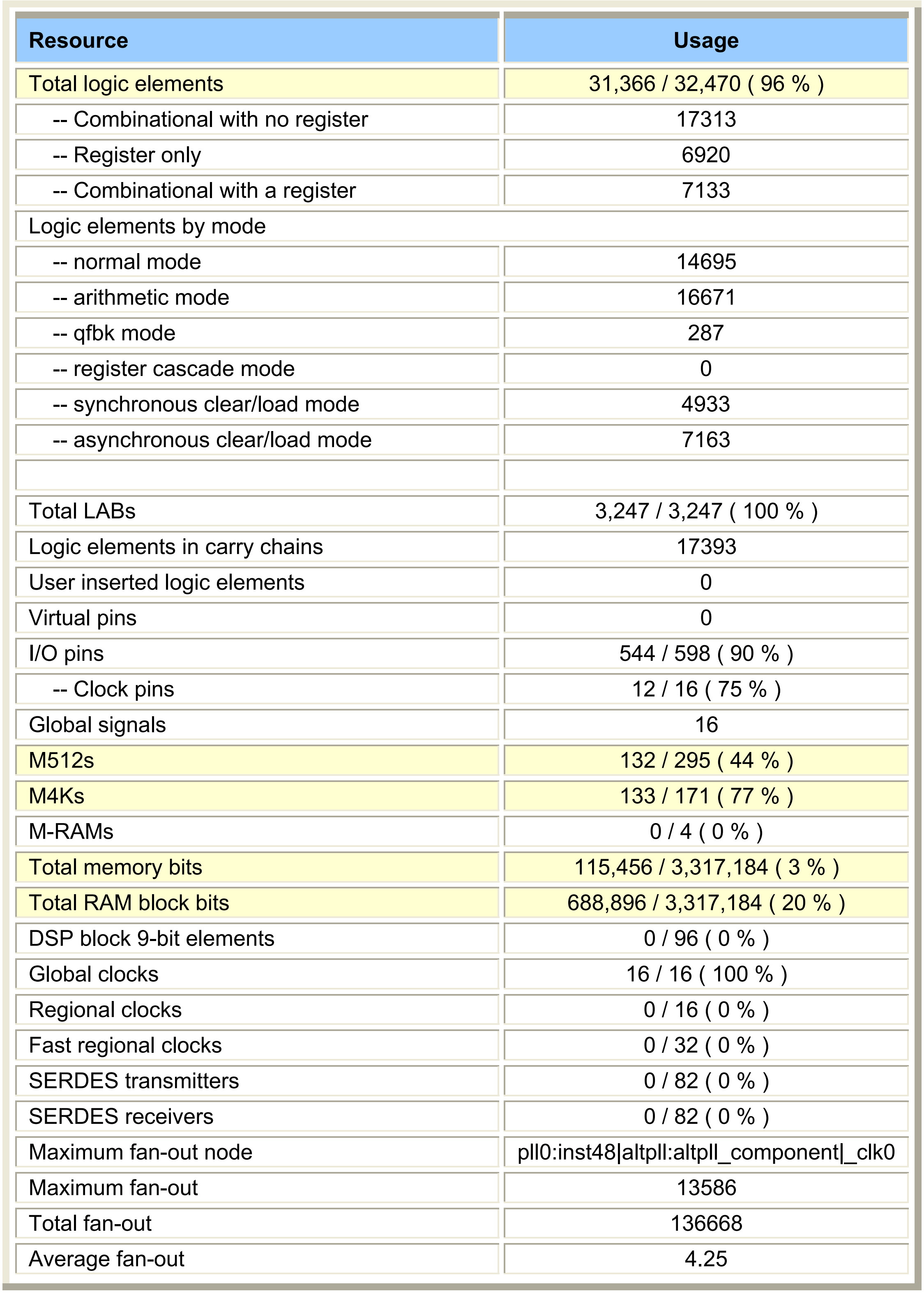


Figure C-2: Top-Level Schematic of BLMTC for the DESY Installation.

Table C-3: Resource Utilisation of BLMTC System for the DESY Installation.



Resource	Usage
Total logic elements	31,366 / 32,470 (96 %)
-- Combinational with no register	17313
-- Register only	6920
-- Combinational with a register	7133
Logic elements by mode	
-- normal mode	14695
-- arithmetic mode	16671
-- qfbk mode	287
-- register cascade mode	0
-- synchronous clear/load mode	4933
-- asynchronous clear/load mode	7163
Total LABs	3,247 / 3,247 (100 %)
Logic elements in carry chains	17393
User inserted logic elements	0
Virtual pins	0
I/O pins	544 / 598 (90 %)
-- Clock pins	12 / 16 (75 %)
Global signals	16
M512s	132 / 295 (44 %)
M4Ks	133 / 171 (77 %)
M-RAMs	0 / 4 (0 %)
Total memory bits	115,456 / 3,317,184 (3 %)
Total RAM block bits	688,896 / 3,317,184 (20 %)
DSP block 9-bit elements	0 / 96 (0 %)
Global clocks	16 / 16 (100 %)
Regional clocks	0 / 16 (0 %)
Fast regional clocks	0 / 32 (0 %)
SERDES transmitters	0 / 82 (0 %)
SERDES receivers	0 / 82 (0 %)
Maximum fan-out node	pll0:inst48|altpll:altpll_component|_clk0
Maximum fan-out	13586
Total fan-out	136668
Average fan-out	4.25

Appendix D. Linearity Test of the BLM system

A current scan of a big portion of the dynamic range, that is from 10pA to 1mA, in log steps has been done to evaluate the linearity of the BLM system. A current supply (Keithley 6430) was programmed to increase the current in 97 log steps with each step to have duration of 20 seconds. The output data were recorded using the Maximum Values storage functionality of the Logging Navigator shown in Section 8.6.

Devices under test:

- BLMCFC Card V3 (FPGA Firmware: CFC Version 3.07)
- BLMTC Module V2 (FPGA Firmware: DESY Version 5c)

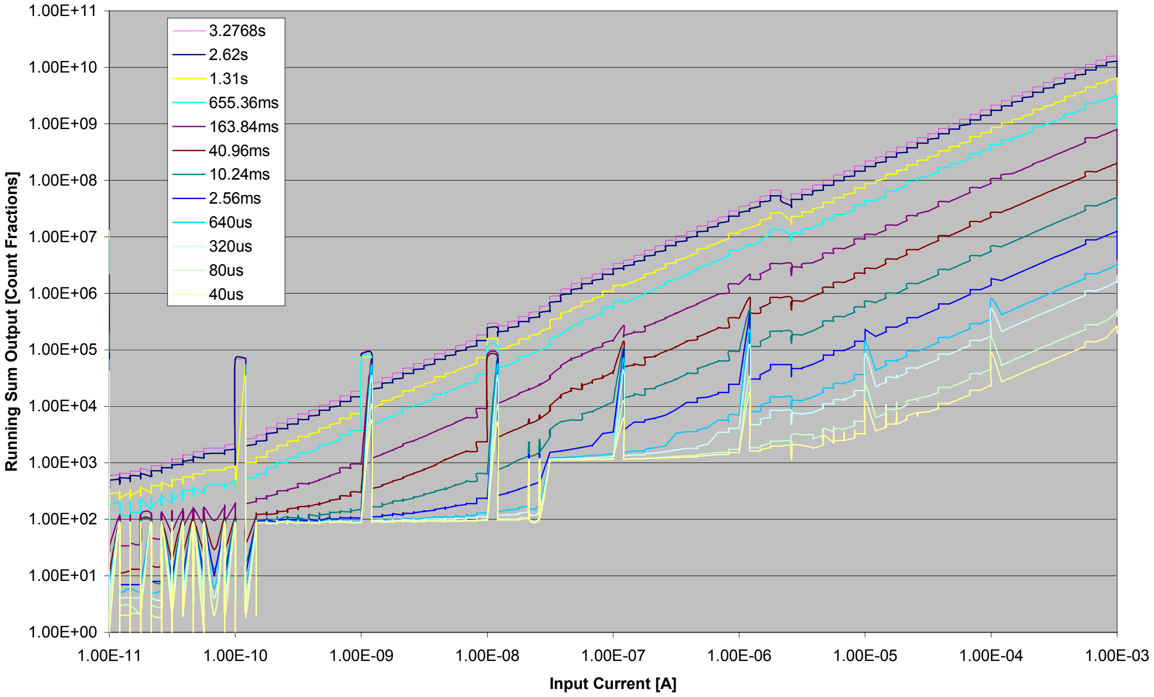


Figure D-1: Linearity Test using a Current Source.

Figure D-1 shows a plot of the results. The parameterisation is for the different loss durations. The perturbations have been found to be simply product of the switching circuitry of the current source and thus can be ignored.

List of Figures

Figure 1-1: The CERN Accelerator Chain. [1] ... 6

Figure 1-2: LHC Layout. [1] ... 8

Figure 1-3: Superconducting Magnet for the LHC. [1] ... 11

Figure 1-4: Comparison of Energy stored in the beam for LHC and other accelerators. [10] ... 12

Figure 1-5: Dump request distribution and the employment of the beam loss system. [15]13

Figure 1-6: Quench Level curves for various Loss Durations as function of the Beam Energy. ... 14

Figure 1-7: Overview of the Beam Energy Acquisition. [12] ... 15

Figure 1-8: Architecture of the Beam Interlock System. [11] ... 16

Figure 2-1: LHC Beam Loss Monitoring System Overview. ... 21

Figure 2-2: Drawing of the LHC ionisation chamber. [16] ... 22

Figure 2-3: Drawing of the secondary emission monitor. [16] ... 23

Figure 2-4: Arrangement of the Detectors around the Magnet. [18] ... 23

Figure 2-5: Picture of the Tunnel (BLMCFC) Card. ... 25

Figure 2-6: Principle of the balanced charge Current-to-Frequency Converter. [17] ... 26

Figure 2-7: Quench Levels defined by number of counts. ... 28

Figure 2-8: Quench Levels defined by number of counts (zoom in 100μs region). ... 29

Figure 2-9: Block Diagram of the BLMCFC ... 30

Figure 2-10: Picture of the BLMTC Processing Module. ... 34

Figure 2-11: Block diagram of processes running in the surface FPGA. ... 36

Figure 2-12: Block Diagram of Combiner Card. [29] ... 37

Figure 2-13: Beam Loss VME to BIC\BE Connection. [29] ... 38

Figure 3-1: Structure of the BLM Packet. ... 49

Figure 3-2: Data Arrangement in the BLM Packet ... 51

Figure 3-3: Gigabit Optical Transmission for the BLM System. ... 52

Figure 3-4: Block Diagram of the GOL Device. [38] ... 53

Figure 3-5: The Gigabit Opto-Hybrid Configuration. [41] ... 54

Figure 3-6: NTPPT-3 series PIN Photodiode with horizontal flange. [47] ... 56

Figure 3-7: Top view of the BLM Mezzanine PCB. ... 58

Figure 3-8: Bottom view of the BLM Mezzanine PCB. ... 59

Figure 3-9: The GOL Test Board. 60
Figure 3-10: Oscilloscope / Control Flag Outputs for a Read Cycle. 61
Figure 3-11: Oscilloscope / Zoom on the signal transitions. 62
Figure 3-12: Oscilloscope / Data available every 40µs. 62
Figure 4-1: Data Block Comparison Using the CRC Algorithm. 69
Figure 4-2: Block Diagram of the RCC (Receive, Check and Compare) Process. 71
Figure 4-3: Block Diagram of CRC-32 Parallel Implementation. [57] 74
Figure 4-4: Block Symbol of CRC-32 Parallel Implementation created with Quartus II. .. 75
Figure 4-5: Block Diagram of the 8B/10B Encoding Process. 78
Figure 4-6: Implementation of the Signal Select Function using Logic Gates. 83
Figure 4-7: RCC Block Symbol created with Quartus II. 85
Figure 4-8: RCC Schematic Implementation with Quartus II. 86
Figure 4-9: RCC Simulation / Only One Signal Correct. 88
Figure 4-10: RCC Simulation / Both CRCs have Error. 89
Figure 4-11: SignalTap II / Maximum Length of Acquisition. 90
Figure 4-12: SignalTap II / Non-Synchronised TLK clocks. 90
Figure 4-13: SignalTap II / SOF Problem. 91
Figure 5-1: Systems Used for the Data Acquisition. 97
Figure 5-2: Timing waveform of read cycle. 97
Figure 5-3: ADC and CFC Output (1st case). 98
Figure 5-4: ADC and CFC Output (2nd case). 99
Figure 5-5: LabVIEW Online Display Application GUI. 101
Figure 5-6: Recording and display of the raw ADC Data using a LabVIEW application. 102
Figure 5-7: Plot (10,000 samples recording) of the ADC data output (ADC_out) and after the Minimum-Hold-Value function (ADC_min). 103
Figure 5-8: Plot (1,000 samples recording) of the ADC data output (ADC_out) and after the Minimum-Hold-Value function (ADC_min). 103
Figure 5-9: ADC Fraction Output when using the Minimum-Value-Hold Function. 104
Figure 5-10: Data-Combine Output when using the Minimum-Value-Hold Function. 105
Figure 5-11: Over-estimation when using the Minimum-Value-Hold (MVH) Function. . 106
Figure 5-12: Overview of the Data-Combine Process. 107
Figure 5-13: Timing Simulation / Case of very slow losses 108
Figure 5-14: Timing Simulation / Minimum Value is retained. 108

Figure 5-15: Timing Simulation / Data-Combine Function Validation. 109
Figure 5-16: SignalTap II / Very Slow Losses - Correct Output.....................................110
Figure 5-17: SignalTap II / Very Slow Losses - Error Output. 110
Figure 5-18: SignalTap II / Slow Losses – Revised Version to include delayed arrival of Count..111
Figure 6-1: Dynamic range for the BLMAs and BLMSs. [78] 116
Figure 6-2: Data Processing Techniques Comparison (Case I). 117
Figure 6-3: Data Processing Techniques Comparison (Case II)......................................118
Figure 6-4: Production of Successive Running Sums. 119
Figure 6-5: Cascading of the Shift Registers in the SRS technique. 120
Figure 6-6: Running Sum Calculation. 123
Figure 6-7: Using a Shift Register with Taps for keeping a Running Sum. 124
Figure 6-8: Optimisation in Memory Usage by the Shift Register...................................126
Figure 6-9: Production of two Running Sums for four Detectors with Memory Sharing. 128
Figure 6-10: Successive Running Sums from 40μs to 10ms (for four detectors).............129
Figure 6-11: Successive Running Sums from 81ms to 84sec (for four detectors). 130
Figure 6-12: Successive Running Sums in the BLMTC..131
Figure 6-13: Simulation / Sub-SystemA's Running Sum (RS) outputs (slow losses).......133
Figure 6-14: Simulation / Sub-SystemA's Running Sum (RS) outputs (fast losses). 134
Figure 7-1: Threshold Comparator (TC) and Masking Processes Block Diagram...........138
Figure 7-2: TC Process Part (Comparing two Running Sums from 16 Detectors). 145
Figure 7-3: Schematic Block of the Th-Table Function. 149
Figure 7-4: Masking Process for the BLMTC. 151
Figure 8-1: Logging and Post Mortem System Overview.[91] 158
Figure 8-2: Transmitted Packet for the Logging System..160
Figure 8-3: OnePulse Circuit Functionality...162
Figure 8-4: ESLog Block for the Error and Status Reporting. 163
Figure 8-5: Error and Status Reporting (ESR) Process. 164
Figure 8-6: Maximum Values Calculation of the Successive Running Sums from four Detectors. 167
Figure 8-7: Maximum Values Logging for the BLMTC System. 168
Figure 8-8: Overview of Post Mortem data storage. 171
Figure 8-9: LHC controls architecture diagram. [95] 172

Figure 8-10: Proposed Graphical Representation of the Logging in the Control Room. .. 173
Figure 8-11: Display of Maximum Values History by the Navigator. 174
Figure 8-12: Plotting of Maximum Values History by the Navigator. 174
Figure 8-13: Error and Status panel at the Acquisition Navigator. 175
Figure 8-14: Averaging and Storing of Maximum Values History by the Navigator. 176
Figure B-1: BLM Mezzanine Schematic Drawing (page 1 of 2). 194
Figure B-2: BLM Mezzanine Schematic Drawing (page 2 of 2). 195
Figure C-1: Top-Level Schematic of BLMTC for the PS Booster Installation. 198
Figure C-2: Top-Level Schematic of BLMTC for the DESY Installation. 200
Figure D-1: Linearity Test using a Current Source. 202

List of Tables

Table 2-1: Current-to-Frequency Converter and Ionisation Chamber Specifications.........26

Table 2-2: Estimated Fluence Levels at 0.45 and 7 TeV..27

Table 2-3: Number of Counts Produced at 0.45TeV...27

Table 2-4: Number of Counts Produced at 7TeV...28

Table 2-5: Stratix Package Options & I/O Pin Counts. [27] 33

Table 2-6: Stratix Device Family Features. [27] 33

Table 2-7: Radiation Dose withstood without Error...40

Table 3-1: Formatting of the Status Word...49

Table 3-2: Summary of Data included in the Transmission Packet. 50

Table 3-3: Receive Status Signals. 55

Table 4-1: Parameters for Various Standard CRC Algorithms. 74

Table 4-2: Outputs of the Valid Signal Selection Function...82

Table 4-3: Truth Table of Signal Select Function. 83

Table 4-4: RCC Fitter Resource Utilisation Summary..87

Table 5-1: Data-Combine Resource Utilisation Summary...107

Table 5-2: Comparison of Acquisition Outputs with Expected Values............................109

Table 6-1: Dynamic range for the BLMAs and BLMSs in p/m/s. [78]............................115

Table 6-2: Successive Running Sums configuration used in BLMTC.............................122

Table 6-3: Successive Running Sum Configuration (including unused outputs).............125

Table 6-4: Memory Resources Utilisation for different configurations of the Shift Registers..127

Table 6-5: Resource Usage by the Successive Running Sums for four Detectors. 132

Table 7-1: Threshold Table Example for one Detector..141

Table 7-2: Functional families of Beam Loss Monitors...142

Table 7-3: Example of information stored in the BLMTC card’s Masking Table...........143

Table 7-4: Resource Utilisation by the TC Process..146

Table 7-5: Optimal Memory Utilisation by the Threshold Table.....................................147

Table 7-6: Actual Memory Utilisation by the Threshold Table. 147

Table 7-7: Resource Utilisation by the Th-Table Function..150

Table 7-8: Resource Utilisation by the Masking Process...152

Table 8-1: Memory Mapping of the Error and Status Reporting. 164

Table 8-2: Resource Utilisation by the ESR Process..165

Table 8-3: Memory Mapping for the Maximum Values of the Successive Running Sums. 169

Table 8-4: Resource Utilisation by the Maximum Value Process....................................170

Table C-1: Successive Running Sum Configuration for the PS Booster (including unused outputs). 197

Table C-2: Resource Utilisation of BLMTC System for the PS Booster Installation. 199

Table C-3: Resource Utilisation of BLMTC System for the DESY Installation...............201

List of Acronyms

ADC	Analogue-to-Digital Converter
ASIC	Application Specific Integrated Circuit
BEA	Beam Energy Acquisition
BEM	Beam Energy Meter
BETS	Beam Energy Tracking System
BGA	Ball Grid Array
BIC	Beam Interlock Controller
BLM	Beam Loss Monitoring
BLMTC	Beam Loss Monitor Threshold Comparator
CERN	Conseil Européen pour la Recherche Nucléaire.
CFC	Current-to-Frequency Converter
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DDR	Double Data Rate
DSP	Digital Signal Processing
FBGA	FineLine BGA
FESA	Front-End Software Architecture framework
FPGA	Field Programmable Gate Array
GOH	Gigabit Opto-Hybrid
GOL	Gigabit Optical Link ASIC device
GUI	Graphical User Interface
HDL	Hardware Description Language
I/O	Input/Output
IP	Intellectual Property
IP	LHC Interaction Point
JTAG	Joint Test Action Group
LAB	Altera Logic Array Block

LBDS	LHC Beam Dumping System
LBIS	LHC Beam Interlock System
LE	Altera Logic Element
LEP	Large Electron Positron Collider
LVDS	Low Voltage Differential Signaling
LUT	Look-up Table
MTBF	Mean Time Between Failures
MAC	Media Access Control
MIPS	Million Instructions per Second
NVRAM	Non-Volatile RAM
OTP	One Time Programmable
PCB	Printed Circuit Board
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
PS	Proton Synchrotron
PSB	PS Booster
PQFP	Plastic Quad Flat Pack
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register Transfer Level
SEU	Single Event Upset
SoC	System-on-Chip
SLP	Safe LHC Parameters
SPS	Super Proton Synchrotron
SRAM	Static Random Access Memory
TTC	Timing, Trigger and Control
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration
VME	VERSAmodule Eurocard
UTC	Universal Time Co-ordinates