

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 29 juin 2022 par

Etienne FORTIN

Commissioning and performance of the trigger and readout
system of the liquid argon calorimeter of the ATLAS experiment.

Discipline

Physique et Sciences de la Matière

Spécialité

Instrumentation

École doctorale

ED 352 Physique et sciences de la matière

Laboratoire/Partenaires de recherche

Aix-Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France

Composition du jury

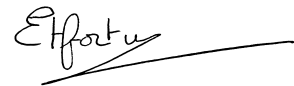
Michel Lefebvre Université de Victoria, Canada	Rapporteur
Caroline Collard IPHC Strasbourg	Rapporteuse
Philippe Schwemling Université Paris Cité et Irfu/CEA-Saclay	Examinateur
Luis Hervas CERN et Université Au- tonome de Madrid	Examinateur
Emmanuel Monnier CNRS/CPPM	Directeur de thèse
Georges Aad CNRS/CPPM	Co-Directeur de thèse
Cristinel Diaconu CNRS/CPPM	Examinateur et Président du jury

Affidavit

I, undersigned, Etienne FORTIN, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific direction of Emmanuel Monnier and Georges Aad, in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body.

Marseille the 22/04/2022



Cette œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Liste de publications et participation aux conférences

Liste des publications réalisées dans le cadre du projet de thèse:

1. Aad, G., Berthold, AS., Calvet, T. et al. Artificial Neural Networks on FPGAs for Real-Time Energy Reconstruction of the ATLAS LAr Calorimeters. Comput Softw Big Sci 5, 19 (2021). <https://doi.org/10.1007/s41781-021-00066-y>
2. G. Aad, A. V. Akimov, K. Al Khoury. et al. The Phase-I Trigger Readout Electronics Upgrade of the ATLAS Liquid Argon Calorimeters (2022), <https://doi.org/10.48550/arXiv.2202.07384>

Participation aux conférences et écoles d'été au cours de la période de thèse:

1. EPS-HEP2019: Poster, "Development of the ATLAS Liquid Argon Calorimeter Readout Electronics for the HL-LHC"
2. ISOTDAQ 2020, International School of Trigger and Data Acquisition
3. INSTR20, Talk "The Phase-I Trigger Readout Electronics Upgrade of the ATLAS Liquid Argon Calorimeters"
4. Journées techniques détecteur IN2P3 2021
5. TIP2021: Poster, "ATLAS LAr Calorimeter Commissioning for LHC Run-3"
6. ICNFP2021: Talk, "Machine Learning for Real-Time Processing of ATLAS Liquid Argon Calorimeter Signals with FPGAs"

Résumé

Les recherches en physique des hautes énergies cherchent à identifier les constituants fondamentaux de la matière et à découvrir les lois qui régissent leurs interactions. Pour mener des expériences à haute énergie, on utilise des accélérateurs de particules et des collisionneurs. Le collisionneur de particules le plus puissant est le grand collisionneur de hadrons (LHC) installé au CERN (Organisation Européenne pour la Recherche Nucléaire). Quatre expériences sont hébergées aux quatre points de collision de ce collisionneur. L'une de ces expériences est ATLAS, qui consiste en un détecteur de particules à usage général. Pour détecter les phénomènes à faible probabilité dans les données de collisions, une grande quantité de collisions est nécessaires. Le LHC va connaître deux phases de mise à niveau afin d'augmenter sa luminosité et donc le nombre de collisions. La première phase de mise à niveau est en cours de finalisation à l'heure actuelle, avant le début d'une nouvelle campagne de prise de données appelée " Run 3 ". La deuxième phase débutera en 2026 et aboutira au LHC à haute luminosité (HL-LHC).

Une mise à niveau du détecteur ATLAS est nécessaire à chacune des phases de mise à niveau du LHC. Cela permet de s'assurer que l'acquisition de données et les performances en physique ne sont pas dégradées par les conditions plus strictes induites par la luminosité plus élevée du LHC. Au cours de la première phase de mise à niveau, un nouveau chemin de données a été ajouté au système de déclenchement du calorimètre à argon liquide. L'électronique de ce nouveau chemin numérise les signaux provenant du détecteur grâce à des cartes électroniques, situées sur le détecteur, avant de les envoyer aux systèmes de traitement d'ATLAS. Cela permet d'augmenter d'un facteur dix la granularité de lecture. Ce système numérique, décrit dans cette thèse, remplace l'ancien système électronique basé sur une transmission analogique des signaux du détecteur. Les nouvelles cartes de ce système sont configurées et contrôlées par un nouveau système basé sur des serveurs OPC-UA. L'installation et la mise en service de ce système ont été achevés avec succès en mars 2022. La description du système complet, sa mise en œuvre et ses performances sont présentés et discutés. La deuxième phase de mise à niveau du calorimètre à argon liquide aura lieu lors du prochain arrêt prolongé du LHC, à partir de 2026. Au cours de cette deuxième phase, la chaîne électronique complète de ce calorimètre sera remplacée, en particulier ses cartes de traitement des signaux électroniques, apportant ainsi de nouvelles capacités de calcul et de traitements des signaux. Ce nouveau système de lecture est présenté puis l'utilisation de réseaux de neurones artificiels pour reconstruire en énergie les signaux du détecteur est décrite et étudiée. Embarqué sur ces cartes électroniques, la mise en œuvre de ces réseaux et les performances obtenues sont ensuite présentées

et discutées. Mots clés: LHC, ATLAS, Calorimètre à Argon Liquide, Déclenchement et acquisition de données, Système de contrôle de détecteur, Réseau de neurones récurrents, FPGAs

Abstract

Research in High Energy Physics seeks to identify the fundamental constituents of matter and uncover the laws that govern their interactions. To conduct experiments at high energy particle accelerators and colliders are being used. The actual most powerful particle collider is the Large Hadron Collider (LHC) installed at CERN. Four experiments are hosted on the four collision points of this collider. One of these experiments is ATLAS which consists of a general purpose particle detector. To detect phenomena with low probability in the collision data, large amount of collisions are needed. The LHC will undergo two upgrade phases to increase its luminosity and therefore the number of collisions. The first upgrade phase is being finalized at the moment before the start of the run 3. The second phase will start in 2026 and will lead to the High Luminosity LHC (HL-LHC).

An upgrade of the ATLAS detector is needed at each of the LHC upgrade phases. This insures that the data acquisition and the physics performance are not degraded with the more stringent conditions induced by the higher LHC luminosity. During the first upgrade phase a new trigger data path is added for the liquid argon calorimeter. The on-detector electronic boards of this new path digitize the electronic signals from the detector before sending them to the ATLAS processing systems. It allows to increase the readout granularity by a factor of ten. This digital system, described in this thesis, will replace the old electronic system which is based on analog transmission of the detector signals. The new on-detector boards of this system are configured and monitored using a new system based on OPC-UA servers. The installation and commissioning of this system was successfully completed in March 2022. The description of the complete system, its implementation and performances are presented and discussed. The second upgrade phase of the liquid argon calorimeter will take place in the next long shutdown of the LHC starting in 2026. During this second phase, the complete electronic chain of this calorimeter will be replaced, in particular its electronic signal processing boards, thus bringing new computing and signal processing capabilities. This new readout system is presented then the use of artificial neural networks to reconstruct the detector signals into energies is described and studied. Embedded on these electronic boards, the implementation of the neural networks and the performances obtained are presented and discussed.

Keywords: LHC, ATLAS, Liquid Argon Calorimeter, Trigger and data acquisition,

Detector control system, Recurrent neural networks, FPGAs.

Remerciements

First I would like to thank my supervisors , Georges Aad and Emmanuel Monnier for their support and advices during my 4 years at CPPM. I would like to thanks the CNRS and the Region Sud for the co-funding of my PhD and the partnership of the Nexvision company and its CEO, Mr Vincent Carrier. I would also like to thank all of my colleagues from the ATLAS LAr upgrade group at CPPM, in particular Thomas Calvet, Daniele Laugier, Franck Salomon and Bernard Dinkespiler for all their expertise and help. Furthermore I would like to thanks all the CPPM members who make my work possible, in particular Thierry Mouthuy, Patrick Pangaud , and Marlon Barbero as service and group leaders. A special thanks to Isabelle Richer who makes all my missions at CERN possible. The work presented here would not have been possible without the trust of the different CERN team involved. In particular I would like to thank Piotr Nikiel and Paris Moschovakos for all their support on the development of the DCS and OPC-UA related projects. I would like to thank the LAr operation team and all the run coordinators for their help to understand the complex LAr detector. A particular thanks goes for Clement Camincher who provide me precious knowledge on the ATLAS detector and the liquid Argon system. I would like to thanks all my defense committee: Michel Lefebvre, Caroline Collard, Philippe Schwemling, Luis Hervas, Cristinel Diaconu for the time spent to review my work. Finally I would like to thank all my friends, at Marseille and at CERN, for all their support during these years.

Résumé en français

0.1. Contexte

Les recherches en physique des hautes énergies visent à identifier et à étudier les constituants fondamentaux de la matière et à découvrir les lois qui régissent leurs interactions. Plusieurs approches contribuent à cet objectif : mesures de précision du modèle standard, contraintes sur les scénarios au-delà du modèle standard, et recherche de nouveaux processus de désintégration de particules connues. Plusieurs expériences ont été conçues à cette fin. Elles sont principalement basées sur des collisionneurs de particules produisant les collisions à haute énergie nécessaires à ces expériences.

Pour partager la complexité et les coûts croissants de ces expériences de physique des hautes énergies, d'énormes collaborations internationales ont été créées. L'un des principaux hôtes de ces collaborations et expériences est l'Organisation européenne pour la recherche nucléaire (CERN), située à Genève, à la frontière franco-suisse. Depuis sa fondation en 1952, il a été à l'origine de nombreuses découvertes en physique des hautes énergies. Le CERN abrite le plus grand et le plus puissant complexe d'accélérateurs au monde. Ce complexe d'accélérateurs fournit des faisceaux de particules à haute énergie à plusieurs expériences. L'accélérateur le plus grand et le plus puissant du CERN est le Grand collisionneur de hadrons (LHC). Il accélère deux faisceaux de particules circulant en sens inverse à une énergie pouvant atteindre 7 TeV par faisceau. Les faisceaux sont composés de nombreux paquets de particules qui entrent en collision en quatre points d'interaction le long de l'accélérateur et créent des événements de collision à une fréquence de 40 MHz. Chaque collision crée un flux de particules de haute énergie qui se désintègrent en de nombreuses particules secondaires qui finissent par aboutir dans le détecteur entourant le point de collision. La détection de ces particules créées permet de reconstruire l'événement complet. L'un des détecteurs situés à l'un des points d'interaction du LHC est le détecteur ATLAS.

Depuis sa première exploitation en 2011, le LHC a permis aux physiciens de réaliser de nombreuses études. Celles-ci ont conduit à la découverte du boson de Higgs en 2012. En dix ans d'exploitation, les données de collision enregistrées par le LHC sont à l'origine de plus de 2800 publications scientifiques, dont 1025 pour ATLAS. Après la découverte du boson de Higgs, les nombreuses mesures précises du modèle standard et la limitation des recherches de nouvelles particules, le travail des physi-

ciens se concentre aujourd'hui sur la recherche d'indices de nouvelle physique et d'événements très faibles. Pour atteindre cet objectif, il faut disposer de beaucoup plus de données sur les collisions. Pour répondre à ce besoin, le CERN a annoncé en 2013 un programme de mise à niveau du LHC et des expériences connexes. Une énorme amélioration de la luminosité est prévue, ainsi qu'une légère augmentation de l'énergie du faisceau de particules. La mise à niveau du LHC vers le HL-LHC sera effectuée au cours de deux longs arrêts de plusieurs années. Le premier arrêt a duré 3 ans et s'est terminé au début de cette année. Le deuxième arrêt est prévu pour 2026. La figure 0.1 montre le calendrier de cette mise à niveau.

L'amélioration du LHC s'accompagne d'une mise à jour des différentes expériences. Ces mises à jours permettent de conserver des bonnes performances lors du passage au HL-LHC. Les améliorations des expériences ont lieu aussi lors des long arrêts.

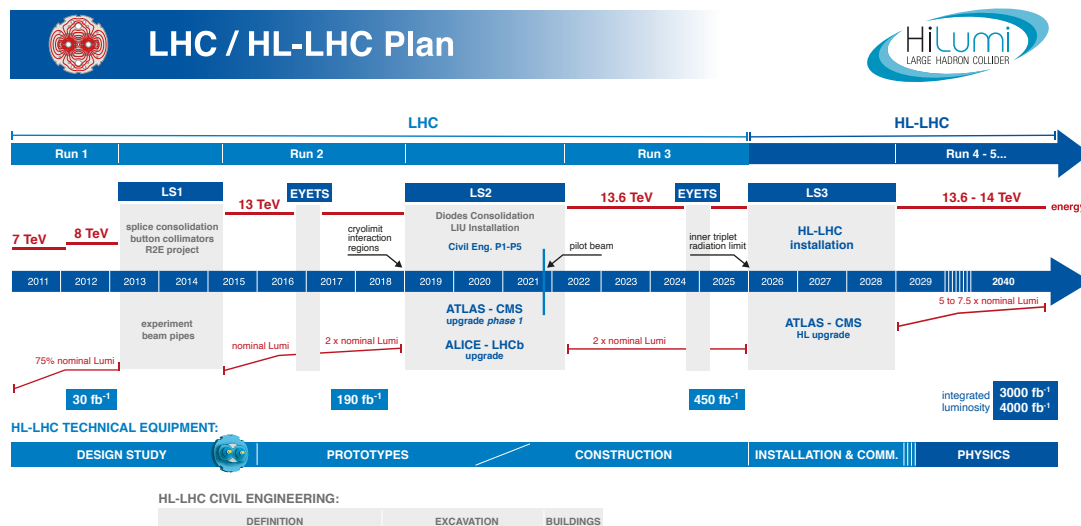


Figure 0.1.: Calendrier du HL-LHC en Janvier 2022

L'expérience ATLAS est un cylindre de 46 m de long, 25 m de diamètre et 7000 tonnes. La figure 0.2 montre cette expérience. Il est composé de six couches de sous-systèmes enroulés autour du point de collision à l'intérieur de deux systèmes de champ magnétique intense qui dévie la trajectoire des particules chargées. Ces sous-détecteurs enregistrent la trajectoire, la quantité de mouvement et l'énergie des particules pour les identifier. Ce détecteur permet d'étudier plusieurs phénomènes physiques, notamment le modèle standard, le boson de Higgs, le quark top, les dimensions supplémentaires et la matière noire.

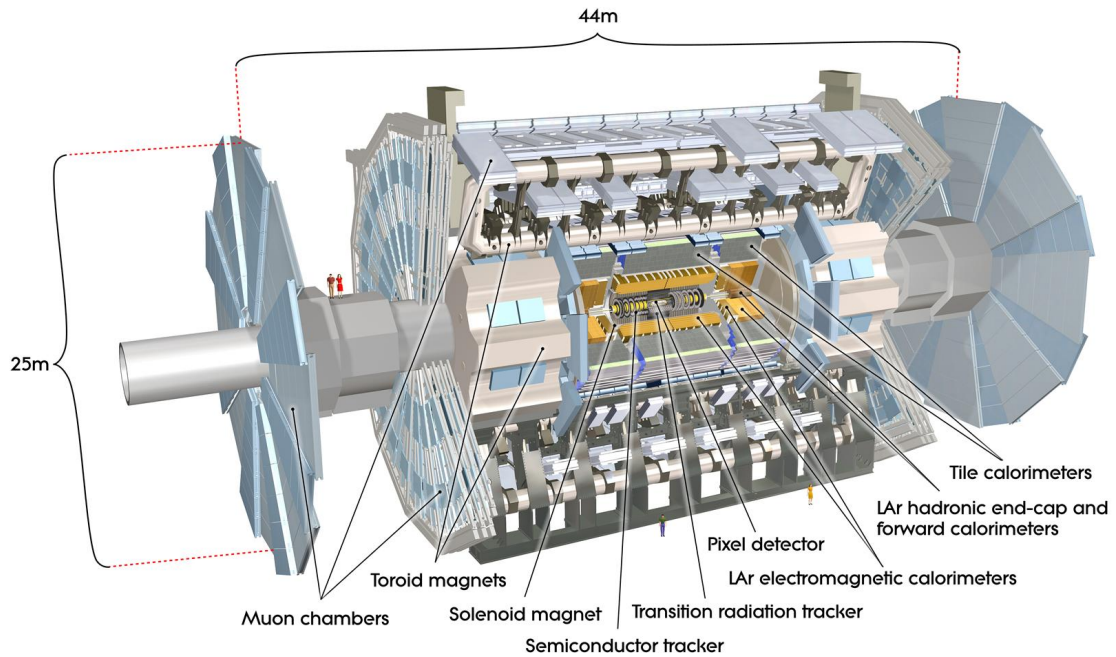


Figure 0.2.: Schéma du détecteur Atlas indiquant la localisation de tout ses sous-systèmes [34]

Un de ces sous système plus particulièrement étudié dans cette thèse, est le calorimètre à échantillonnage à argon liquide d'ATLAS (LAr), représenté en figure 0.3. Il est utilisé par ATLAS pour déterminer l'énergie de particules issues de la collision. Il détermine l'énergie des particules électromagnétique et des jet dans toutes les directions. Le calorimètre avant détermine aussi l'énergie des hadrons. Il est composé de quatre sous-détecteurs : tonneau électromagnétique, bouchons électromagnétiques, bouchons hadroniques, et calorimètre avant.

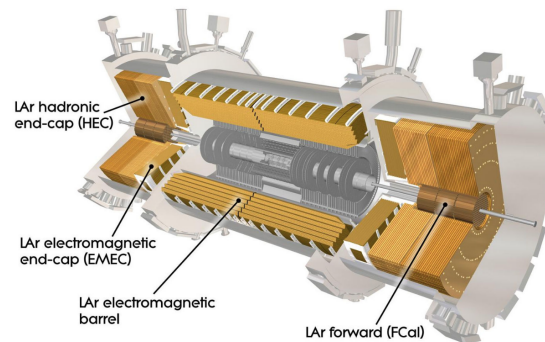


Figure 0.3.: Vu du calorimètre à argon liquide d'ATLAS

Le système d'acquisition du calorimètre LAr est composé de plusieurs cartes comme représenté sur la figure 0.4. Deux systèmes reçoivent les données du calorimètre LAr, l'acquisition d'ATLAS et le déclencheur de niveau 1. La transmission des données à l'acquisition d'ATLAS est réalisé par deux cartes : la carte frontale (FEB), installée sur le châssis frontale et la carte de lecture (ROD) installé dans la salle de comptage USA15. La carte frontale amplifie, met en forme et numérise les signaux laissés dans l'argon liquide par les particules chargées traversant le calorimètre pour chaque cellule du détecteur. La numérisation est effectuée uniquement pour les événements sélectionnés par le déclencheur. Les signaux sont ensuite envoyés à la carte ROD qui calcule les énergies et le temps à partir des signaux numérisés et envoie les énergies et, pour certains signaux à haute énergie, les énergies, le temps et x^2 au système d'acquisition de données.

La transmission au système de déclenchement est réalisée par deux cartes : la carte de sommation de couches (LSB) et la carte de construction de tours de déclencheur (TBB). La carte de sommation de couches est une carte mezzanine installée sur la FEB. Elle additionne les signaux amplifiés de plusieurs cellules d'une même couche et les met en forme. Les sommes analogiques sont envoyées par le LSB à la carte TBB installée dans le même châssis frontale. La TBB additionne les signaux pour créer une région d'intérêt pour le système de déclenchement. La tour de déclenchement (TT) regroupe les cellules dans une région de $\Delta\eta \sim 0,1$ et $\Delta\phi \sim 0,0982$. Ce signal est envoyé par des câbles analogiques au système de déclenchement de niveau 1 installé dans la salle de comptage USA15. La décision de déclenchement est envoyé aux châssis frontaux par le système TTC. Sur chaque châssis frontal, la décision de déclenchement est reçue par la carte contrôleur. La carte contrôleur propage cette décision à tous les FEB afin qu'ils puissent numériser le signal correspondant échantillonné dans les mémoires analogiques.

Pour conserver de bonnes performances dans les conditions du HL-LHC avec un empilement accru, le système de calorimètre LAr doit être mis à niveau. Sa mise à niveau est divisée en deux phases suivant celles du LHC. Au cours de la première phase, le système de lecture du déclencheur a été mis à niveau. Le système de déclenchement analogique utilisé lors de la dernière période de prise de données a été remplacé par un système numérique. En outre, la granularité de ce système de déclenchement a été multipliée par dix. Lors de la deuxième phase, les cartes de l'acquisition principale seront remplacées.

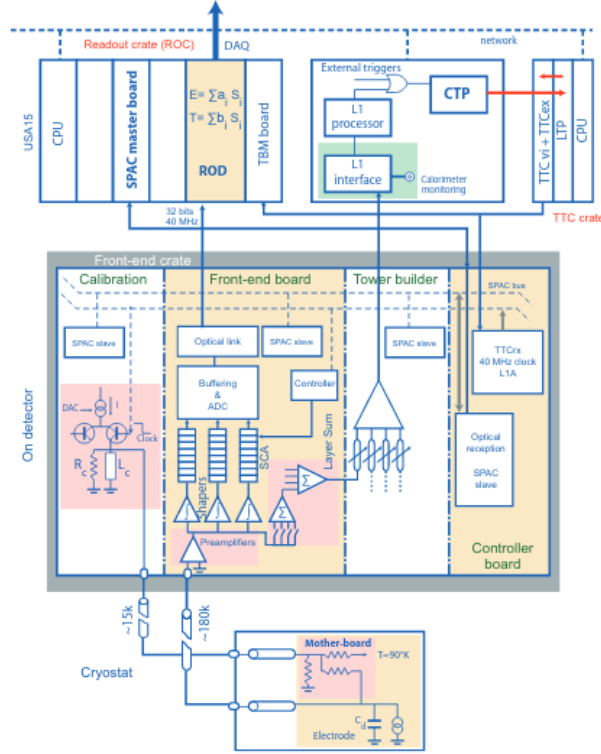


Figure 0.4.: Schema block du système de lecture du détecteur ATLAS[1]

La mise à niveau de la phase 1 du déclencheur décuple la granularité des données envoyées au déclencheur L1. Les tours de déclenchement seront divisées en 10 super-cellules. Les super cellules (SC) regrouperont 4 cellules de détection pour les couches 0 et 2, et 8 cellules de détection pour les couches 1 et 3. Pour créer ces signaux de supercellules, la carte de sommation des couches a été remplacée et une nouvelle carte frontale, appelée Lar Trigger Digitization Board (LTDB) a été installée dans le châssis frontal. La LTDB numérise les signaux des super cellules envoyés par la carte de sommation et les envoie au système de traitement numérique LAr (LDPS). Ce système est composé de plusieurs cartes nommées LAr Digital Processing Board (LDPB). Le LDPS calcule les énergies de la super cellule à partir des valeurs numériques échantillonnées provenant de la LTDB et les envoie au premier niveau du système de déclenchement.

Au cours de la mise à niveau phase II, les cartes frontales seront remplacées par la carte frontale 2 (FEB2). Les cartes ROD seront remplacées par la nouvelle carte LASP (LAr Signal Processing). La FEB2 numérise à 40Mhz les signaux avec une précision accrue (2x16bits) par rapport à la FEB (100kHz 3x12 bits) et le LASP calculera et reconstruira les énergies à partir du signal numérisé. La mise en mémoire tampon

analogique des données présente sur le FEB ne sera pas présente dans le FEB2. La mise en mémoire tampon sera numérique et sera effectuée par la carte du LASP.

0.2. Contrôle et surveillance des cartes avants de la phase I de mise à niveau

La figure 0.5 montre les différents composants matériels et logiciels du nouveau système de la Phase I. Ces composants font partie de deux systèmes quasiment indépendants : Le système de contrôle et surveillance numérique (DCS) et le système de déclenchement et d'acquisition de données (TDAQ). Le DCS gère l'alimentation de base et la surveillance du système. Le système TDAQ a plusieurs objectifs : configuration et distribution du Timing and Trigger Control (TTC), configuration des cartes, déclenchement et enfin acquisition des données. Les deux systèmes sont indépendants à l'exception des serveurs OPC-UA (Open Platform Communications Unified Architecture) de la LTDB qui sont partagés entre les deux.

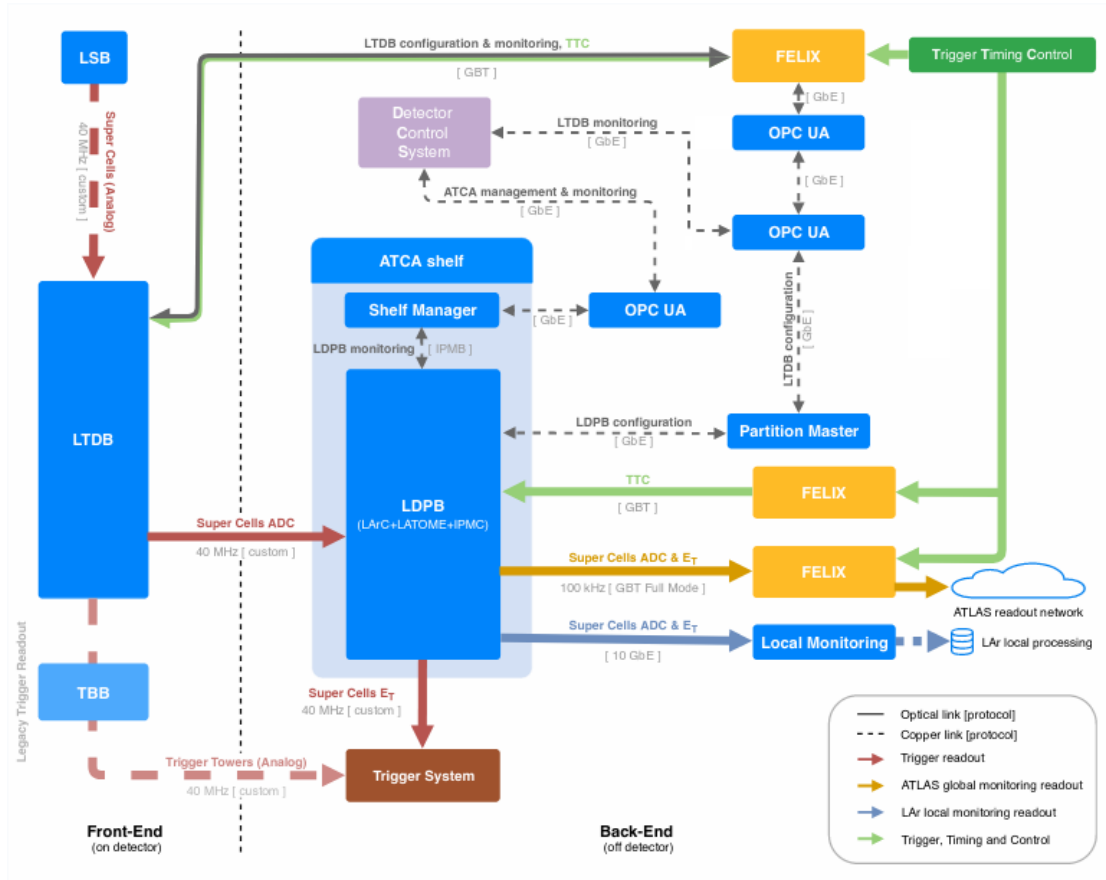


Figure 0.5.: Schéma du système de déclenchement numérique du LAr après la phase I d'amélioration. Ce système est composé de deux cartes , la LTDB et la LDPB. L'interface de ces cartes avec les systèmes TDAQ et DCS est faites au travers de plusieurs cartes FELIX et de serveurs OPC-UA.

0.2.1. LTDB

Un schéma simplifié de la LTDB est présenté dans la figure 0.6. La carte est composée de deux chemins, le chemin de contrôle et le chemin de données.

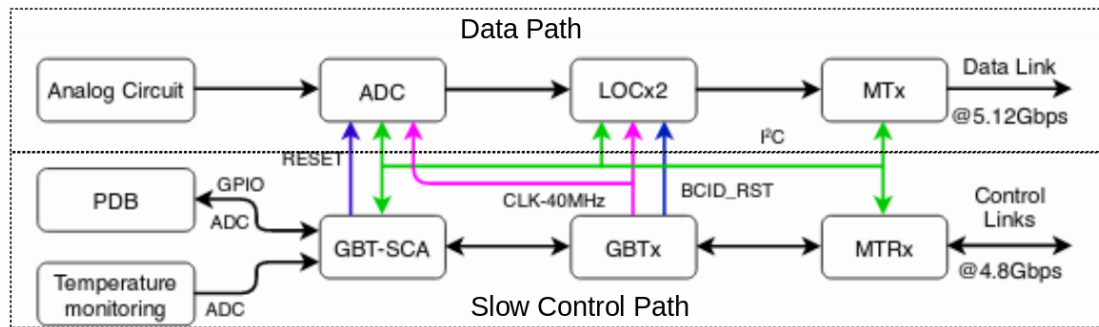


Figure 0.6.: Schema conceptuel de la LTDB. La figure montre les différents composants de son chemin de données et de celui de contrôle.

L'entrée du chemin de données est constituée des signaux des super cellules analogiques provenant du circuit analogique. Ces signaux sont numérisés dans l'ADC. Après avoir été numérisés, les données correspondantes sont transmises au composant LOCx2. Le LOCx2 encode le signal numérique qui est envoyé par le transmetteur optique (MTx) au LDPB. Ce chemin utilise les données provenant du système TTC (horloge et BCID_RST), crée et traite les données nécessaires au système de déclenchement et au système de lecture et est configuré par le système TDAQ.

Le chemin de contrôle reçoit les données de configuration des serveurs OPC-UA, les signaux de synchronisation du système TTC, et envoie les valeurs de contrôle aux serveurs OPC-UA. Les données du chemin de contrôle sont reçues et transmises par le protocole GBT. Les trames GBT sont reçues/transmises par le composant optique MTRx et sont décodées/encodées par le composant GBTx. La trame GBT contient trois types de données : TTC, configuration et surveillance. Les données TTC sont reçues par le GBTx qui les décode et les transmet aux différents composants LTDB qui en ont besoin. L'horloge est transmise aux ADCs et aux composants LOCx2. Le signal Bunch Crossing Reset est transmis uniquement au LOCx2. Les commandes de contrôle OPC-UA sont envoyées au GBT-SCA. Le GBT-SCA distribue les commandes aux différents composants de la carte. Ces composants font partie du chemin de contrôle et de données mais aussi du Power Distribution Board (PDB). La carte PDB est une carte mezzanine composée principalement de régulateurs. La PDB fournit toutes les tensions et la puissance nécessaires à la LTDB.

0.2.2. Logiciel de contrôles

Plusieurs outils logiciels ont été développés pour pouvoir surveiller, contrôler et configurer les LTDBs et sont présentés dans cette thèse ainsi que les performances obtenues. Les outils sont divisés en 3 catégories : des outils de bas niveau fournis par la collaboration ATLAS qui assurent la communication générique avec le matériel via l'interface FELIX, des outils logiciels LAr spécifiques qui mettent en œuvre les besoins réels du matériel LAR, et des clients utilisateurs qui mettent en œuvre l'interface avec

les utilisateurs DCS et TDAQ. Plusieurs logiciels sont nécessaires pour contrôler les LTDB :

- **Felixcore** : Felixcore est le logiciel faisant le lien entre la communication Ethernet et la carte FELIX via son pilote. Felixcore est un client tcp-ip fournissant des méthodes pour utiliser la communication GBT des cartes pour accéder aux LTDBs. Cette couche est développée par l'équipe Felix.
- **le serveur OPC-UA SCA** : Ce serveur OPC-UA est développé par l'équipe DCS d'ATLAS afin de fournir un serveur unifié pour gérer la communication avec les SCA. Le SCA est principalement utilisé par la nouvelle petite roue et le calorimètre à argon liquide pour leurs mises à niveau de phase I.
- **OPC-UA LAr LTDB Server** : Le serveur OPC-UA LAr LTDB est un serveur OPC-UA spécifique conçu pour couvrir les besoins du sous-détecteur LAr de la LTDB. Il fournit des méthodes de haut niveau pour surveiller, configurer et calibrer le LTDB.
- **Clients et interfaces utilisateur** : Deux clients OPC-UA fournissent des interfaces utilisateur graphiques pour contrôler et surveiller la carte : un client pour le système DCS et un pour le système TDAQ.

0.2.3. Résultats

Au total, 124 LTDB ont été installés sur le détecteur. Leur système de contrôle repose sur plusieurs serveurs basés principalement sur le protocole OPC-UA. Le logiciel le plus complexe de cette chaîne est le serveur OPC-UA LAr LTDB. Le développement de ce serveur a été un défi car il doit être robuste et doit atténuer plusieurs problèmes matériels. Le serveur OPC-UA LAr LTDB possède plusieurs interfaces avec divers composants. Ce serveur a été optimisé pour réduire la consommation de ressources et le temps de configuration des LTDBs. Il comprend également plusieurs procédures de récupération pour gérer les erreurs matérielles. Toutes ces fonctionnalités ont été testées avec succès lors de l'essai pilote du LHC en octobre 2021 et le serveur est utilisé sans problème notable depuis lors. Ce serveur est entièrement finalisé et intégré au système ATLAS. Il est déployé et prêt pour l'exploitation des nouveaux composants de la phase 1 pendant le cycle 3 du LHC qui doit commencer en juin 2022.

0.3. Réseau neuronaux récursif embarqués dans les FGPA pour la mise à niveau phase 2

Pendant la phase 2 de la mise à niveau du calorimètre à argon liquide d'ATLAS, l'électronique de lecture complète sera remplacée. Le FEB et le ROD seront retirés. Le nouveau chemin de données de lecture sera composé de deux nouvelles cartes. La

carte frontale version 2 (FEB2) sera installée dans les châssis frontaux en remplacement de la FEB. Le ROD sera remplacé par une nouvelle carte appelée LAr Signal Processing board (LASP). La FEB2 sera chargée d'amplifier, de mettre en forme et de numériser les signaux analogiques des cellules détectrices. La FEB2 transmettra les données numérisées à la carte LASP installée dans la caverne de comptage USA15. La carte LASP calculera les énergies à partir des valeurs des convertisseurs analogiques numériques (ADC) intégrés dans la carte FEB2. Les valeurs ADC seront transmises au système de déclenchement. Les énergies calculées seront transmises au système de déclenchement et au système d'acquisition de données ATLAS.

La reconstruction de l'énergie pour le calorimètre LAr est effectuée sur les FPGA de la carte LASP en utilisant les données numériques provenant des cartes frontales. Des algorithmes de filtrage numérique sont nécessaires pour réduire le bruit en combinant plusieurs échantillons ADC. Le filtrage numérique est également chargé de trouver le pic de l'impulsion puisque l'échantillonnage peut ne pas être au pic. Au cours du run 3, le filtrage optimal (OF) sera utilisé. Cet algorithme calcule les amplitudes de l'impulsion et son temps (phase) à partir de quatre échantillons. L'algorithme max finder peut être utilisé pour déterminer le BCID de l'impulsion [6]. L'algorithme OF est sensible à la présence de bruit d'empilement. Le bruit d'empilement est le bruit causé par les dépôts d'énergie passés dans la cellule du calorimètre. Un signal d'impulsion causé par l'énergie déposée dans le détecteur est plus long que la période entre deux collisions. Si une énergie est déposée dans une cellule LAr avant la fin de l'impulsion d'un dépôt antérieur, les deux impulsions se chevauchent. L'algorithme de filtrage optimal montre de bonnes performances pour corriger le bruit électronique uniquement dans des conditions de faible empilement. Le filtre optimal (OF) est actuellement utilisé pour reconstruire les énergies LAr et montre de bonnes performances pour corriger le bruit électronique et le pile-up. La mise en œuvre de l'algorithme OF utilise des filtres FIR.

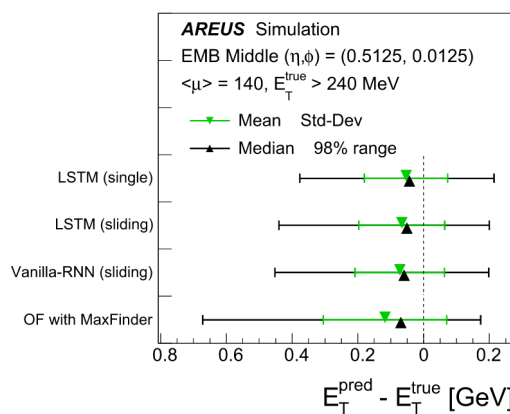


Figure 0.7.: Résolution en énergies des différents algorithmes de réseau neuronaux comparé à l'algorithme OF. Cette figure utilise les données simulées d'AREUS[23] dans les conditions HL-LHC. AREUS est le logiciel simulant le comportement du détecteur.

Avec l'augmentation de l'empilement au HL-LHC, de nouveaux algorithmes sont nécessaires pour reconstruire les énergies déposées dans le calorimètre LAr. Avec les capacités de calcul accrues des nouvelles cartes de lecture du LASP, l'utilisation d'algorithmes d'intelligence artificielle est présentée et discutée dans cette thèse. Plusieurs réseaux d'algorithmes RNN ont été développés au CPPM pour répondre à ce besoin. La comparaison entre les performances de ces différents réseaux est visible en figure 0.7, chacun de ces réseaux est plus performant que l'algorithme OF. La structure interne du réseau est étudiée en détail pour construire des architectures RNN avec des besoins en ressources de calcul réduits tout en maintenant les performances en termes de résolution énergétique. Le bloc de la fonction d'activation, le bloc de la multiplication de la matrice et la quantification du réseau complet ont été optimisés. Des études supplémentaires sont encore nécessaires pour optimiser le bloc d'addition des biais et le bloc de la couche dense. Avec les optimisations actuelles, le réseau vanilla est capable de calculer l'énergie dans un maximum de 370 canaux et peut tenir dans un FPGA Stratix 10. Les réseaux LSTM sont trop grands pour répondre aux exigences de la phase 2 du calorimètre LAr en termes de latence et de nombre de canaux par FPGA. Cependant, les réseaux LSTM peuvent être utilisés à une étape ultérieure pour affiner le calcul de l'énergie à une fréquence plus basse, par exemple pour les événements sélectionnés par le premier niveau du système de déclenchement. Des optimisations supplémentaires pour le réseau vanilla sont nécessaires pour réduire encore sa taille afin qu'il puisse tenir dans le FPGA dans des conditions réelles d'exploitation. Le FPGA doit effectuer des tâches supplémentaires et toutes les ressources ne sont pas disponibles pour le calcul de l'énergie. Les ressources exactes disponibles pour le calcul de l'énergie dépendront du nombre final de canaux traités par un FPGA, du choix exact du FPGA et des ressources nécessaires aux autres tâches. Ces éléments ne sont pas entièrement connus au moment de la rédaction de ce document. Cependant, l'implémentation RNN vanilla présentée dans ce document est un candidat prometteur pour l'utilisation de l'intelligence artificielle pour le calcul de l'énergie déposée dans le calorimètre LAr au HL-LHC.

Contents

Affidavit	2
Liste de publications et participation aux conférences	3
Résumé	4
Abstract	6
Remerciements	7
Résumé en français	8
0.1 Contexte	8
0.2 Contrôle et surveillance des cartes avants de la phase I de mise à niveau	13
0.2.1 LTDB	14
0.2.2 Logiciel de controles	15
0.2.3 Résultats	16
0.3 Réseau neuronaux récursif embarqués dans les FGPA pour la mise à niveau phase 2	16
Contents	19
List of Figures	22
List of Tables	26
Introduction	27
1 ATLAS	30
1.1 CERN and accelerator complex	30
1.1.1 CERN	30
1.1.2 The accelerator complex	31
1.1.3 Large Hadron Collider	32
1.1.4 High-Lumi LHC	33
1.2 ATLAS	35
1.2.1 ATLAS Inner Detector	37
1.2.2 ATLAS Calorimeters	39
1.2.3 ATLAS Muon Spectrometer	49
1.2.4 ATLAS trigger and data acquisition system	50
1.2.5 Control and monitoring of ATLAS	53

2	Control and Monitoring of the front-end boards of the phase-I upgrade	56
2.1	Introduction	58
2.1.1	Overview of the phase-I electronic system	58
2.1.2	ATLAS Detector Control System	60
2.2	Hardware Components	62
2.2.1	LTDB	62
2.2.2	Front-End FELIX	73
2.2.3	LAr Digital Processing Blade	77
2.3	Software	79
2.3.1	Felixcore	80
2.3.2	OpCuaScaServer	80
2.3.3	OpCuaLarLtdbServer	83
2.3.4	The DCS ATLLARLTDB Wincc OA project	92
2.3.5	Trigger and Data acquisition system	97
2.4	DCS Monitoring of the LTDB boards	98
2.5	Control of the board	101
2.5.1	Control by the TDAQ Partition	103
2.5.2	Control by DCS panels	105
2.6	Calibration of the LTDBS boards	106
2.6.1	ADC calibration	107
2.6.2	Link calibration	107
2.6.3	Mapping	109
2.6.4	Error mitigation in the software	111
2.7	Board Installation and validation	112
2.7.1	Results	113
2.8	Conclusion	114
3	Embedded Recursive Neural Network in FGPAs for the phase-II upgrade	116
3.1	Hardware components of the LAr phase 2 upgrade	116
3.2	Reconstruction of the energy deposited in the LAr calorimeter	119
3.3	Neural network theory	121
3.4	Neural Network for LAr energy reconstruction	124
3.4.1	Recurrent Neural Networks	124
3.5	Neural network firmware considerations	129
3.5.1	Internal neural network operations	129
3.5.2	Multiplexing and pipeline	130
3.5.3	Latency	130
3.5.4	Multiplications	131
3.6	Firmware Implementation	132
3.6.1	Intel FPGA blocks	133
3.6.2	The Intel High Level Synthesis	135
3.6.3	Implementation of Matrix Multiplication	136
3.6.4	Activation functions	143

3.6.5	Fixed point representation	146
3.7	Firmware performance	153
3.7.1	HLS resource estimate	154
3.7.2	Implementation and testing on hardware	154
3.8	Conclusion	158
Conclusion		159
Bibliography		161
Acronyms		167
ANNEXES		175
A	SCA channels	175
B	List of all pannels of the ATLLARLTDB project	176
C	OpcUaScaServer Full Device Classes Diagram	177

List of Figures

0.1	Calendrier du HL-LHC en Janvier 2022	9
0.2	Schéma du détecteur Atlas indiquant la localisation de tout ses sous-systèmes [34]	10
0.3	Vu du calorimètre à argon liquide d'ATLAS	10
0.4	Schema block du système de lecture du détecteur ATLAS[1]	12
0.5	Schéma du système de déclenchement numérique du LAr après la phase I d'amélioration. Ce système est composé de deux cartes , la LTDB et la LDPB. L'interface de ces cartes avec les systèmes TDAQ et DCS est faites au travers de plusieurs cartes FELIX et de serveurs OPC-UA.	14
0.6	Schema conceptuel de la LTDB. La figure montre les différent composant de son chemin de donnée et de celui de contrôle.	15
0.7	Résolution en énergies des différents algorithmes de réseau neuronaux comparé a l'agorithme OF. Cette figure utilise les donnée simulée d'AREUS[23] dans les conditions HL-LHC. AREUS est le logiciel simulant le comportement du détecteur.	17
1.1	CERN accelarator complex in 2019[27].	32
1.2	LHC tunnel and Schematic.	32
1.3	HL-LHC Plan in January 2022.	34
1.4	Atlas Detector[34].	35
1.5	Schematic view of the underground ATLAS installation.	36
1.6	Coordinates system of the ATLAS experiment.	36
1.7	Inner detector schematic	37
1.8	d0 (left) and z0 resolution (right) as a function of p_T in ATLAS Run-1 (without IBL) and Run-2(with IBL) [41].	38
1.9	Hit efficiency of the SCT detector.	38
1.10	Position residual width as a function of track P_T for the TRT barrel.	39
1.11	LAr detector view.	40
1.12		41
1.13	LAr barrel cells schematic	41
1.14	Schematic of the HEC cell and module	42
1.15	Shape of the pulse.	43
1.16	Crates position.	44
1.17	Block schematic of the readout of the ATLAS LAr detector[1].	45
1.18	Schematic of the LAr readout after the different upgrade phases.	46
1.19	Geometrical representation of an electromagnetic trigger tower(TT) and geometrical representation of super cells proposed for the Phase I upgrade.	47

1.20	Tile calorimeter concept: scintillating tiles set perpendicular to the beam axis [36].	48
1.21	Three-dimensional view of the muon spectrometer instrumentation indicating the areas covered by the four different chamber technologies.	49
1.22	Muon reconstruction efficiency	50
1.23	Schematic overview of the Trigger and DAQ system in run 1 and 2.	51
1.24	Schematic overview of the Trigger and DAQ system after phase 1 upgrade.	52
1.25	Overall baseline design of the Phase-II TDAQ system	53
1.26	Atlas DCS user interface.	54
1.27	Atlas run control application.	55
2.1	Schematic of the LAr digital trigger system after the phase-I upgrade.	59
2.2	DCS hierarchy schematic.	61
2.3	Overview of the Quasar components[39].	62
2.4	Schematic of the concept of the LTDB.	63
2.5	Picture of an LTDB.	64
2.6	Schematic of the slow control components of the LTDB.	65
2.7	Schematic of the LTDB Control path components and signals used for the configuration between them.	66
2.8	Schematic of the LTDB Data path.	67
2.9	Block diagramm of a Nevis ADC [9].	68
2.10	Reconstructed output of the MDAC stages inside the Nevis ADC.	69
2.11	Schematic of the ADC serializer.	69
2.12	LocIc Frame.	70
2.13	Schematic of the LOCx2 component.	70
2.14	Schematic of the locIc Encoder in the LOCx2.	71
2.15	Chronograph of LOCx2 synch FIFO	72
2.16	Photo of the FELIX-712 PCI-E board.	74
2.17	The GBT frame	75
2.18	The GBT frame which is specific to LAr.	75
2.19	The HDLC frame [12]	76
2.20	Block schematic of the LATOME firmware.	78
2.21	Schematic of the quasar architecture for OpcUaSca server.	80
2.22	Diagram representing the Device Logic Hierarchy of the OpcUaScaServer.	81
2.23	Schematic of quasar architecture for the OPC-UA LAr LTDB server.	84
2.24	Schematic of the quasar generated C++ classes	87
2.25	Schematic of the implementation of the call of a method using address space variables.	88
2.26	Diagram representing the Device Logic Hierarchy of the OpcUaLarLtdb-Server and it Address-Space.	89
2.27	Schematic of the connection between the Wincc OA data-points and the OPC-UA LAR LTDB server address-space.	93
2.28	ATLLARLTDB panel used to create, modify, configure, and delete the data points of the ATLLARLTDB project.	96

2.29	Tree view of the LAr side C partition in the TDAQ system.	97
2.30	FSM of the ENDCAP A LTDBs.	99
2.31	Panel of the status of the ENDCAP side A LTDBs.	100
2.32	Monitoring panel of the LTDB.	101
2.33	LTDB transitions and states.	102
2.34	GUI of the TDAQ system representing a partition with its different run control applications in running state	104
2.35	Expert main panel seen with a non expert account.	105
2.36	Expert panel for a special crate containing 3 LTDBs	106
2.37	View of the results of the LOCx2 clock phase shifter scan inside a ATL-LARLTDB dedicated panel.	108
2.38	View of the results of the CLKDES clock phase shifter scan inside a ATLLARLTDB dedicated panel.	109
2.39	Schematic of the connections between the different hardware and software of the LTDB system.	112
3.1	Picture of the LASP demonstrator board. The two FPGAs are under the two copper heat sinks. The optical emitters/receivers are under the aluminium heat sinks.	118
3.2	Schematic of the block organisation of the LASP firmware and the data flow between these blocks.	119
3.3	Resolution of the transverse energy reconstruction.	120
3.4	A neuron as described by McCulloch and Pitts.	121
3.5	MCP-Neuron implementing the logical AND function.	122
3.6	Perceptron as described by F. Rosenblatt.	123
3.7	Neuron described by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams.	123
3.8	Neural network architecture with several layers.	124
3.9	Difference in architecture between a RNN and a Feed-Forward NN. The RNN cell output is taken as input of the same cell at the next iteration.	125
3.10	Schematic of the different architectures possible for an RNN.	125
3.11	Architectures of recursive cells.	126
3.12	Schematic of a single cell network for an LSTM cell.	127
3.13	Schematic of a sliding windows network for an LSTM cell.	128
3.14	Transverse energy resolution for the optimal filtering and the various RNN algorithms.	128
3.15	Schematic of the computation blocks inside an RNN gate. n is the size of the state vector.	129
3.16	Number of channels fitting into Stratix 10 SX multiplying blocks for a 5 cell LSTM network in sliding windows algorithm as a function of the multiplexing.	132
3.17	Number of channels fitting into Stratix 10 SX multiplying blocks for a 5 cell Vanilla-RNN network in sliding windows algorithm as function of the multiplexing.	132

3.18	Block design of the Stratix 10 SX FPGA[37].	133
3.19	Schematic of a Stratix 10 DSP block.	134
3.20	Schematic of an ALM of the Stratix 10 FPGA.	134
3.21	Design flow for an HLS component and an HDL component.	135
3.22	Schematic of the data interfaces of an HLS streaming component.	136
3.23	Operations implemented by the HLS compiler for two designs of a dot 8 operation.	139
3.24	Resource usage of the ACC19 and the ACC37 designs depending on the target frequency.	140
3.25	Schematic block of a bit shift implementation.	141
3.26	Schematic block of a bit shift implementation with its direction fixed.	142
3.27	Schematic block of a bit shift implementation with its direction fixed and a constant operand value known at compilation time.	142
3.28	RTL view of the Relu activation function.	144
3.29	Approximation of a sigmoid function using a look up table.	145
3.30	Resolution of the Look up table of the sigmoid function(1024 entries between -8 and 8).	145
3.31	Resolution of the Look up table of the sigmoid function(1024 entries between 0 and 8).	146
3.32	Schematic of a vanilla cell with three different data types.	148
3.33	Energy Resolution plot of the output of a vanilla sliding windows RNN with 5 timestamps.	149
3.34	Latency and resource usage of a vanilla RNN sliding windows with 5 timestamps depending of the quantization mode.	150
3.35	Resolution plot of the output of a vanilla RNN sliding windows with 5 timestamps.	151
3.36	Resources usage and latency of a 5 cell vanilla network depending on the quantization mode of its data types.	151
3.37	Resolution of the network depending of the number of bits and of the quantization of the internal fixed point representation.	152
3.38	Resources usage and latency of a 5 cell vanilla network depending on the size and rounding of its internal data type.	152
3.39	Transverse energy reconstruction performance for the optimal filtering and the vanilla RNN implemented both in software (keras) and firmware (HLS).	154
3.40	The Stratix 10 GX devkit with the connected Intermezzo board which is used to test the firmware containing the neural networks.	155
3.41	Schematic of the firmware used to test the network produced by the HLS compiler.	156
3.43	Maximum frequency as a function of the number of implemented networks in one FPGA depending on the choice of the speed grade of the FPGA.	157
.1	Servers and boards	177

List of Tables

2.1	Correspondence between the fiber number in the MTP cable and the LTDB components.	65
2.2	List of the LTDB flavors with the corresponding number of chanel for each flavor.	73
2.3	SCA frame for a single write on 7bit address slave I2C transaction of the first I2C master interface of the SCA.	77
2.4	List of the LATOME Mapping flavors and the corresponding and the corresponding numbers of channels.	79
2.5	The OPC-UA Device classes implemented in the OPC-UA LAr LTDB server.	90
2.6	List of all dpTypes of the ATLLARLTDB project and corresponding number of DPs and DPEs.	93
2.7	List of all events that can start a scripts on panels.	94
2.8	LTDB and LATOME connections.	110
2.9	List of LTDBs replaced during the validation period	114
2.10	List of errors on the LTDBs installed in the detector. These errors are mitigated by a dedicated configuration on the control software.	114
3.1	Truth table of the logical AND function describe as an MCP neuron with $T = 2$ and $g = X_1 + X_2$	122
3.2	Minimum latency of the basic computation block used in the RNNs.	131
3.3	Resource usage for three different implementations of a vector multiplication and accumulation.	138
3.4	Core maximum frequency of the Stratix 10 DSP with and without the chain in/out for Stratix 10 FPGAs with two different speed grades.	140
3.5	Resources usage of the implementation of a bit shift depending on the operand types.	142
3.6	Resolution and resource usage for two different LUT implementations of a 5 cell sliding window LSTM network.	146
3.7	Quantization modes for a fixed point[7].	147
3.8	Parameters of the Vanilla RNN network implementation chosen for hardware testing.	153
3.9	HLS estimated resource usage of the vanilla RNN implementation.	154
3.10	Comparison between the HLS and Quartus resource usage.	157
.1	List of all the channels of the SCA chip	175
.2	List of all panels of the ATLLARLTDB project	176

Introduction

Research in high energy physics seeks to identify and study the fundamental constituents of matter and uncover the laws that govern their interactions. Several approaches are followed to pursue this goal such as: precision measurements of the standard model of interactions, constraints on beyond standard models scenarios, and search for new decay processes of known particles. Several experiments have been designed for this purpose. These are mainly based on particle colliders producing the high energies collisions needed by these experiments.

To share the increasing complexities and costs of these high energy physic experiments, huge international collaboration have been created. One of the main host for such collaborations and experiments is the European organization for nuclear research (CERN) located in Geneva at the Swiss/France boarder. Since its founding in 1952, it has been at the origin of many discoveries in high energy physics. CERN hosts the biggest and most powerful accelerator complex in the world. The accelerator complex provides high energy particle beams to several experiments. The largest and most powerful accelerator at CERN is the Large Hadron Collider (LHC). It accelerates two particle beams circulating in opposite direction at an energy of up to 7 TeV per beam. The beams are composed of many bunches of particles that collide at four interaction points along the accelerator and create collision events at a 40MHz frequency. Each collision creates a flux of high energy particles which decay into many secondary particles that eventually end in detector surrounding the collision point. The detection of such created particles allows to reconstruct the full event. One of the detector at one of the LHC interaction point is the ATLAS detector.

Since its first operation in 2011, the LHC allowed physicists to perform many studies. These led to the discovery of the Higgs boson in 2012. In ten years of operations, the recorded LHC collision data are at the origin of more than 2800 scientific publications, with 1025 from ATLAS. After the Higgs boson discovery and many precise standard model measurements and limit on new particle searches, the physicist work is nowadays focusing to search for hint of new physics and very rare events. To achieve this goal, much more collision data are needed. To fullfill this need, CERN announced in 2013 an upgrade program of the LHC and the related experiments. A huge luminosity upgrade is planned as well as a slight increase of the particle beam energy. The luminosity is linked to the amount of particles per bunch and the bunch size. The increase of the bunch density will cause more collisions during bunch crossing. In standard LHC conditions, up to 80 collisions occur on average per bunch crossing,

at the HL-LHC, up to 200 collisions per bunch crossing are foreseen. This average number of collision per event is named pile-up(μ). The LHC upgrade to the HL-LHC will be made during two long shutdowns of several years. The first shutdown lasted 3 years and finished at the beginning of this year. The second shutdown is scheduled for 2026.

The ATLAS collaboration is in charge of the ATLAS detector. This collaboration regroups physicists, engineers, technicians, students and support staff. This is one of the largest scientific collaboration in the world with 3000 scientific authors for a total of 5500 members from one hundred eighty one institutions in forty two countries. The ATLAS experiment is composed of several layers of detectors which contribute to locate and measure the energies and charge of the created particles in the bunch collision as well as identify and track some of them. Two calorimeter systems are responsible for the energy measurements. The Liquid Argon (LAr) calorimeters target electromagnetic particles as well as hadronic particles for its endcap section. The tile calorimeters only target hadronic particles. Chapter 1 gives an overview of the LHC accelerator complex and of ATLAS detector as well as their performances and the future upgrade programs.

To keep good performances at HL-LHC conditions with increased pile-up, the LAr calorimeter system must be upgraded as well. Its upgrade is split into two phases following the LHC ones. During the first upgrade phase, its trigger readout system, which contributes to select interesting calorimeter events, has been upgraded. The analog trigger system used in the past data taking period has been replaced by a digital one. Furthermore the granularity of this trigger system has been increased tenfold. For this new trigger system, 124 on-detector boards, named LTDB, have been installed and validated in addition to needed modifications on the standard readout on-detector electronics and crates. Each of these LTDB boards digitizes the data from up to 320 readout channels of the LAr detectors. To control, calibrate and configure these boards, a robust hardware and software system has been developed based on Open Platform Communication Unified Architecture (OPC-UA) servers and dedicated application-specific integrated circuit (ASIC) named Slow Control Adapter (SCA) located on the LTDBs. This system and its performances is fully described in chapter 2.

During the phase 2 upgrade, the full electronic readout of the LAr calorimeter will be upgraded. The Marseille Particles Physics Center (CPPM) is in charge of the conception and development of the processing boards of this new electronics readout system named LAr Signal Processor (LASP). This board is composed of two huge field programmable gates array (FPGA) processing the signals from the detector cells into deposited energies. Each LASP will handle several hundred of cells.

Because of the increased HL-LHC pileup, the performances of the actual algorithms, used to reconstruct the cell energies, will decrease. With the increased computation power offered by the LASP board, it becomes possible to implement neural networks on the LASP FPGAs. These possibilities are studied as well as the expected performances. The chapter 3 presents in details this study and describes the LASP board.

1. ATLAS

Summary

1.1	CERN and accelerator complex	30
1.1.1	CERN	30
1.1.2	The accelerator complex	31
1.1.3	Large Hadron Collider	32
1.1.4	High-Lumi LHC	33
1.2	ATLAS	35
1.2.1	ATLAS Inner Detector	37
1.2.2	ATLAS Calorimeters	39
1.2.2.1	ATLAS Liquid Argon Calorimeter	39
1.2.2.2	Phase I upgrade of the liquid argon calorimeter	46
1.2.2.3	Phase II upgrade of the liquid argon calorimeter	47
1.2.2.4	ATLAS Tile Calorimeter	47
1.2.3	ATLAS Muon Spectrometer	49
1.2.4	ATLAS trigger and data acquisition system	50
1.2.5	Control and monitoring of ATLAS	53

1.1. CERN and accelerator complex

CERN hosts the biggest accelerator complex dedicated to high energy physics. The biggest accelerator of this complex is the Large Hadron Collider (LHC), a circular accelerator of 27 km circumference located underground in the Geneva area at the Swiss/France border at a 100 m depth, and designed to accelerate two proton beams at 6.5 TeV and collide them in four interaction points. Each of these collision points host a detector build and operated by a collaboration of institutes.

1.1.1. CERN

The European Council for Nuclear Research (CERN) was founded after the second world war when several European countries wanted to create a common nuclear physics laboratory to unify European physicists and share the increasing cost of research in nuclear physics [42]. The council was created in 1952 and the Geneva

area was chosen as the site to build the first accelerator of the complex thanks to its central position in Europe, to its neutrality during the war and to it already hosting several international institutions. Construction began in 1954 in Meyrin at the border between Switzerland and France, and the European Organisation for Nuclear Research began. This organisation kept the CERN acronym throughout time.

Since its creation CERN was at the origin of several major discoveries and new technologies. Among them are the discoveries of the W and Z bosons in 1983, the world wide web in 1989 and the Higgs boson in 2012. CERN research is mainly based on its accelerator complex. The accelerator complex provides high energy beams for collision in LHC experiments as well as in several experimental areas.

1.1.2. The accelerator complex

The CERN accelerator complex is able to accelerate protons, electrons/positrons, and heavy ions. Figure 1.1 show its actual composition. The beams are composed of many bunches of particles.

The proton beam is created by a linear accelerator, called LINAC 4. It accelerates H^- ions, created out of a bottle of di-hydrogen, at an energy of 160 MeV[43]. At the output of LINAC 4, the electrons of the H^- ions are stripped to create a proton beam. The beam is then sent to the Proton Synchrotron Booster (PSB) which increases its energy up to 2 GeV[16]. The beam coming out of the PSB is sent to the Proton Synchrotron (PS) which accelerates it to 25 GeV, followed by the Super Proton Synchrotron (SPS) to reach 450 GeV, and finally to the LHC where it reaches up to 7 TeV (and 6.5 TeV up to now).

The ion beam is created in the LINAC 3 accelerator. From the plasma sources the accelerator creates a beam of lead ions. These ions are prepared in the Low Energy Ion ring to be sent to the LHC by the PS and SPS accelerators.

1. ATLAS – 1.1. CERN and accelerator complex

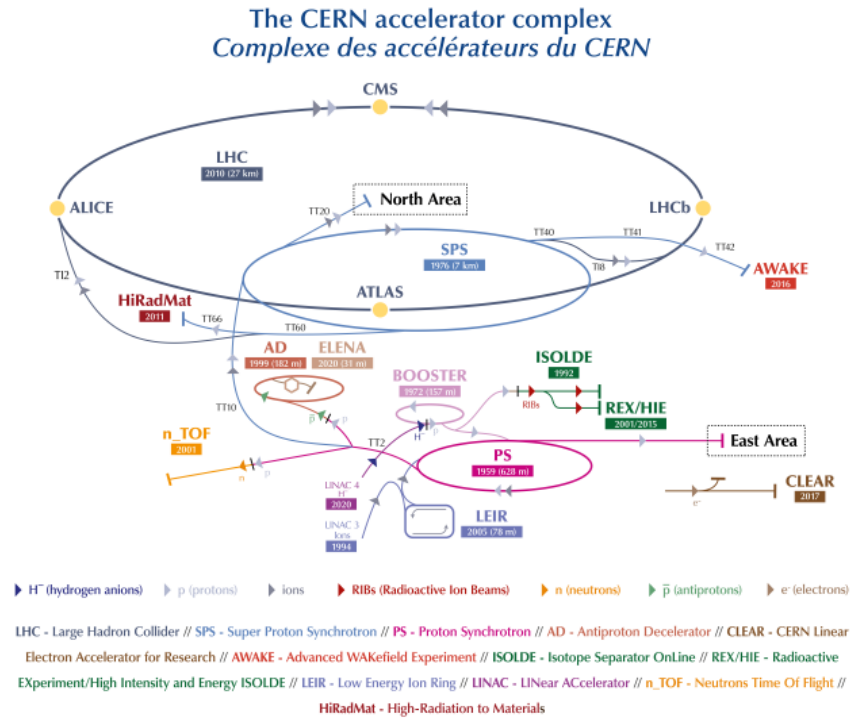


Figure 1.1.: CERN accelerator complex in 2019[27].

1.1.3. Large Hadron Collider



(a) Photo of the LHC.

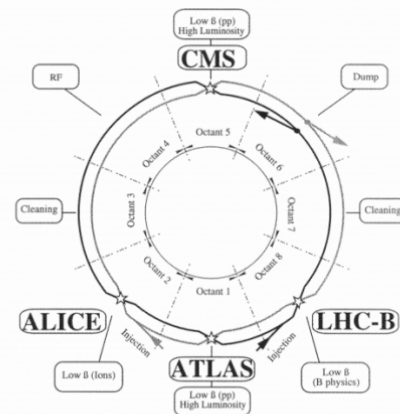


Figure 1: Schematic layout of the LHC

(b) Schematic of the LHC and location of its collision points.

Figure 1.2.: LHC tunnel and Schematic.

The large hadron collider(LHC) is a 27 km length circular accelerator situated underground at 100 m at CERN straddling the Swiss/France boarder. Figure 1.2 is a photo of it and its schematic. The LHC is composed of several parts: Supra-conductive magnets bending the beam to follow the beam pipe, Radiofrequency cavities accelerating the beam, four interaction points, insertion point, and dumping point. At each four collision points is installed an experiment . The four experiments are:

- ATLAS: general purpose detector that investigates a wide range of physics including searches for Higgs boson, dark matter and beyond standard model physics.
- CMS : general purpose detector, its physics program is the same as ATLAS but the detector design use different technologies.
- LHCb : specialized detector that investigates the slight differences between matter and antimatter by studying a type of particle called the "beauty quark", or "b quark".
- ALICE : specialized detector dedicated to heavy-ion physics that investigates physics with extreme energy density in quark-gluon plasma forms.

The LHC can contain up to 3564 bunches of particles. In its actual nominal configuration, 2808 bunches are filled with particles the others are empty bunches. The two beams are crossed in four points. At these crossing points collisions of particles composing the beams occur. The collisions frequency is 40 MHz. Many collisions occur per bunch crossing and an average of 80 collisions was obtained at the 2018 luminosity. This number will increase to up to 200 collision per bunch crossing with the planned increase of luminosity. These are called pile-up collision and the mean number of collisions per bunch crossing is named μ .

The LHC operation is organised in runs separated by long shutdowns. A run last several years. During the runs several small stops are done during 2 month at the end of the year for technical maintenance. Runs can also have Extended Year End Technical Stops (EYETS) for large maintenance. Between the runs, long shutdown of two or three years are scheduled to upgrade the LHC and its experiments. These upgrades are steps to upgrade the LHC to increase its luminosity.

1.1.4. High-Lumi LHC

Since 2010 were it was fully commissioned for proton-proton collision with a 6.5 TeV center of mass (c.o.m) energy, the LHC was constantly upgraded to reach in 2016 its nominal luminosity of $1 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and in 2022 its nominal energy is expected to be at 13.6TeV [4]. A peak luminosity at two times the nominal one ($2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$) was achieved in 2018. With this increasing luminosity the LHC goal is to reach an integrated luminosity of 350 fb^{-1} for an energy of 14 TeV at the end of run 3. This luminosity and energy are the maximum that the LHC can provide. To fully exploits

1. ATLAS – 1.1. CERN and accelerator complex

the physics potential of the machine, the upgrade to High Luminosity LHC (HL-LHC) was decided by CERN in 2010. The target of the HL-LHC is to increase the luminosity to reach for a center of mass of up to 14 TeV:

- A peak luminosity of $5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$
- An integrated luminosity of 250 fb^{-1} per year
- Over 12 years an integrated luminosity of 3000 fb^{-1}

The actual calendar of the HL-LHC upgrade is shown in figure 1.3. Two long shutdowns will be needed, the long shutdowns 2 and 3. During the second long shutdown that just finished, all the civil engineering needed for the upgrade was done. In the third long shutdown the HL-LHC will be fully installed.

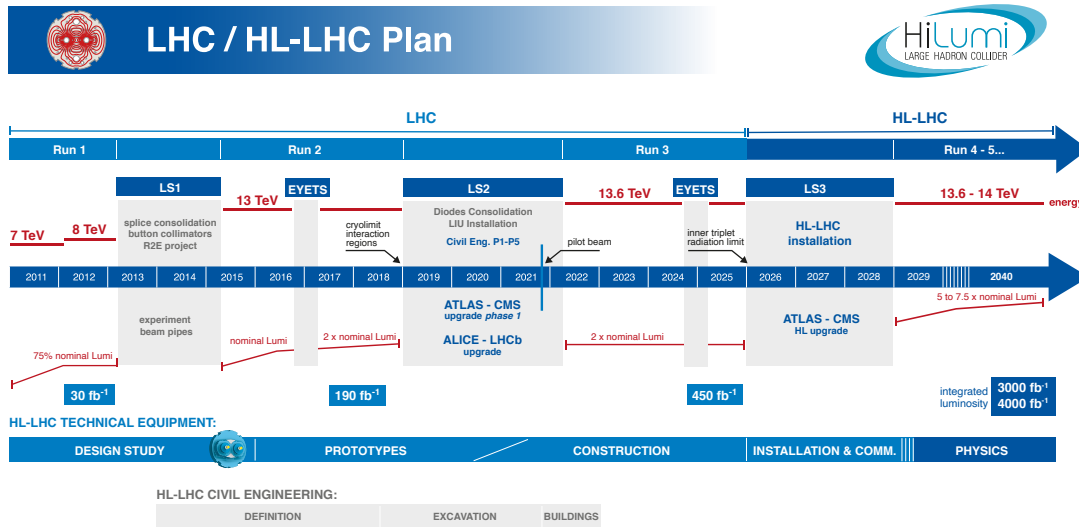


Figure 1.3.: HL-LHC Plan in January 2022.

1.2. ATLAS

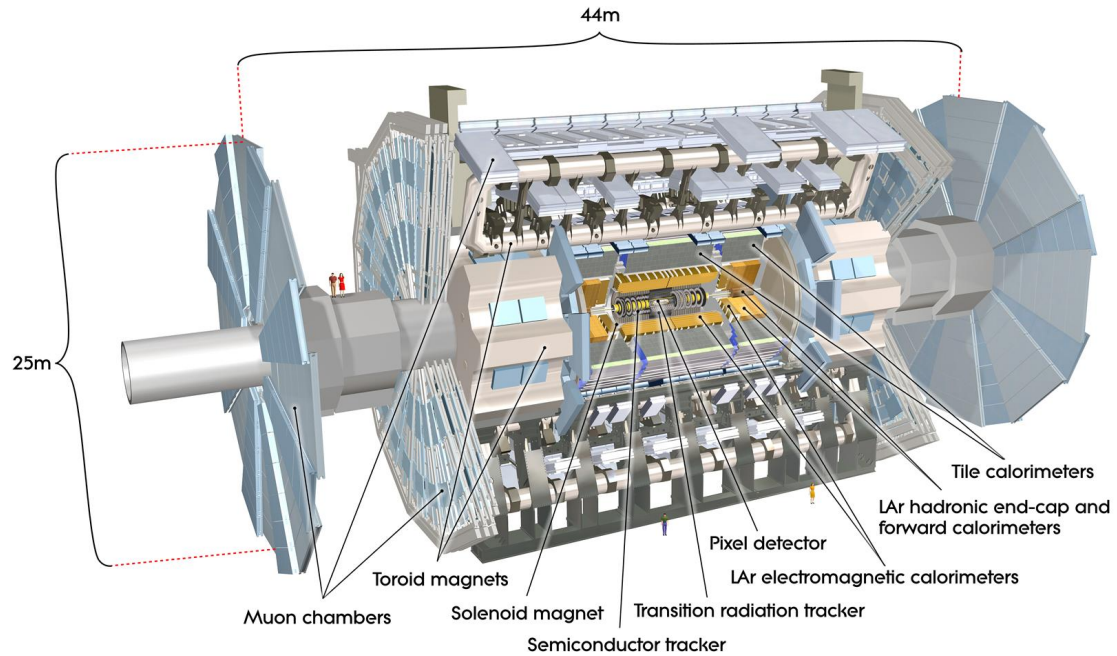


Figure 1.4.: Atlas Detector[34].

The ATLAS experiment shown in figure 1.4 is a 46 m long, 25 m diameter, 7000 tons cylinder. It is composed of six layers of sub systems wrapped around the interaction point inside a huge magnet field bending the charged particles trajectory. These sub detectors allow to record and reconstruct the trajectory, the momentum, and the energies of some of the particles created in each bunch collisions. These recorded information are the basic elements used to study several types of physics including standard model, the Higgs boson, top quark, search for new physics, and dark matter.

The ATLAS experiment is located in a cavern at 100 m underground. Figure 1.5 shows the underground caverns. The detector is installed in the UX15 cavern and occupies almost the entire cavern. Two pits were used to lower the detector parts and are still used during the opening of the detector to bring back and forth the elements of ATLAS such as the muon small wheels and its replacement, the new small wheels, to surface. The UX15 cavern contains the counting room. The readout electronics installed on the detector in UX15 is named the front-end (or on-detector) electronics and have to sustain the radiation level in the cavern. The UX15 cavern and the front-end electronics are subject to radiation, thus access to this cavern during operations is impossible. The USA15 cavern contains the back-end (or off-detector) electronics, this cavern is not subject to radiation.

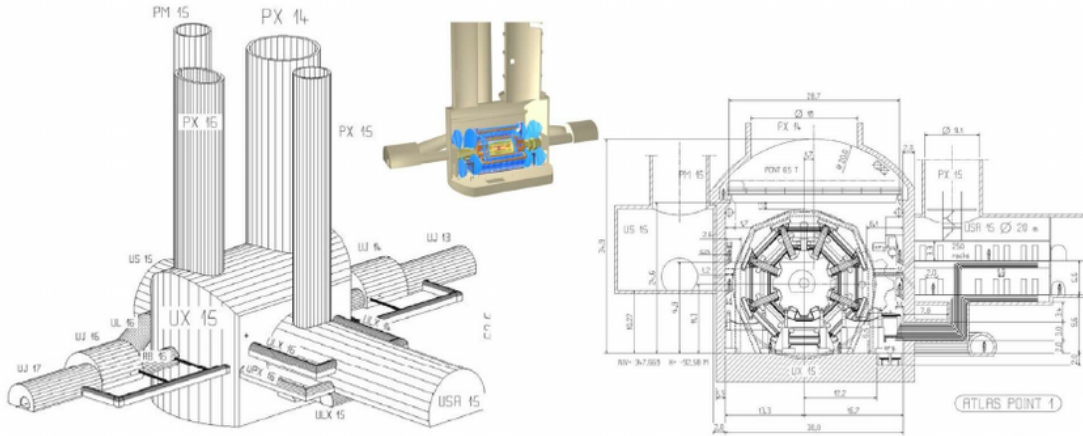


Figure 1.5.: Schematic view of the underground ATLAS installation. The detector cavern is UX15, the adjacent service caverns are USA15 and US15. PX14 and PX16 are the installation shafts for surface access, PM15 and PX15 are the two elevators. The detector is almost as large as the UX15 cavern.

A common coordinates system (shown in figure 1.6) is used by all ATLAS sub systems. The origin is the center of the detector, which is also the nominal interaction point. The Z axis is the beam axis, the positive X axis points to the center of the LHC and the positive Y axis to the earth surface. The positive Z side is named the A side and the negative one the C side of the detector. Cylindrical coordinates are defined along the Z axis, the azimuthal angle ϕ start from the X axis and a radial position r from the beam axis.

Spherical coordinates are defined adding a polar angle Θ from the positive x-axis. often reported in terms of pseudo-rapidity, defined as $\eta = -\ln \tan \Theta/2$.

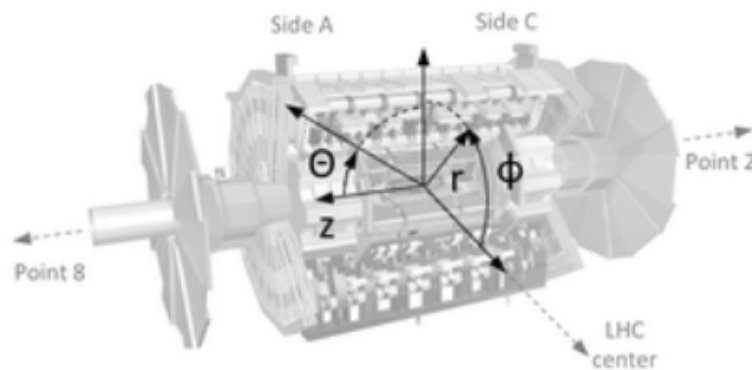


Figure 1.6.: Coordinates system of the ATLAS experiment.

1.2.1. ATLAS Inner Detector

The ATLAS Inner detector is composed of several tracking systems which measure the direction, the momentum and the charge of electrically charged particles. It is surrounded by a solenoid bending the particles trajectories with a two tesla magnetic field. Its innermost parts is the pixel detector, followed by a silicon micro-strip detector (SCT) and finally a transition radiation detector (TRT).

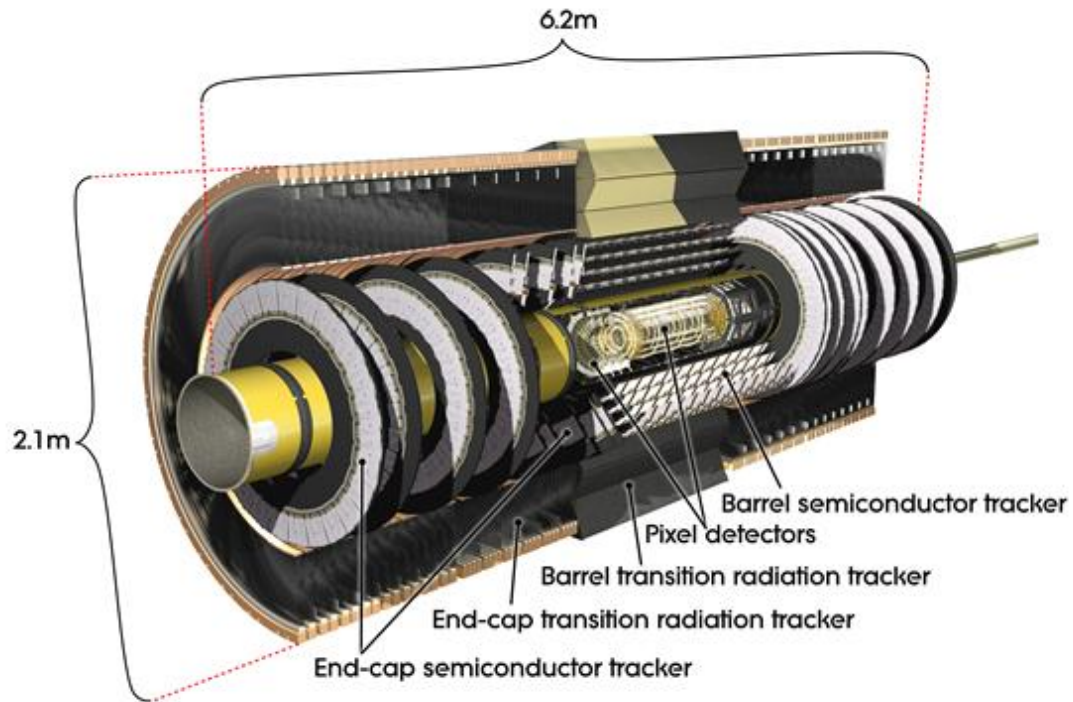


Figure 1.7.: Inner detector schematic[33].

The Pixel Detector [2] is the innermost element containing 80 million channels on four barrel layers and three disk layers at each end of the barrel region named end-caps. The innermost layer is named insertable B-Layer (IBL) and was installed at the end of run 1. The detection in these layers is made of silicon CMOS chips containing 2880 pixels cell covering $50 \times 400 \mu\text{m}^2$. The pixel chip is hybrid. It contains a detection part and an analog part. The analog part amplifies the signal of the detection and compare it to a threshold. The digital part of the chip transfers the address and the timestamp of the pixel above threshold to a buffer waiting a trigger signal to transmit it or discard it. Figure 1.8 shows the resolution of the pixel detector in both directions.

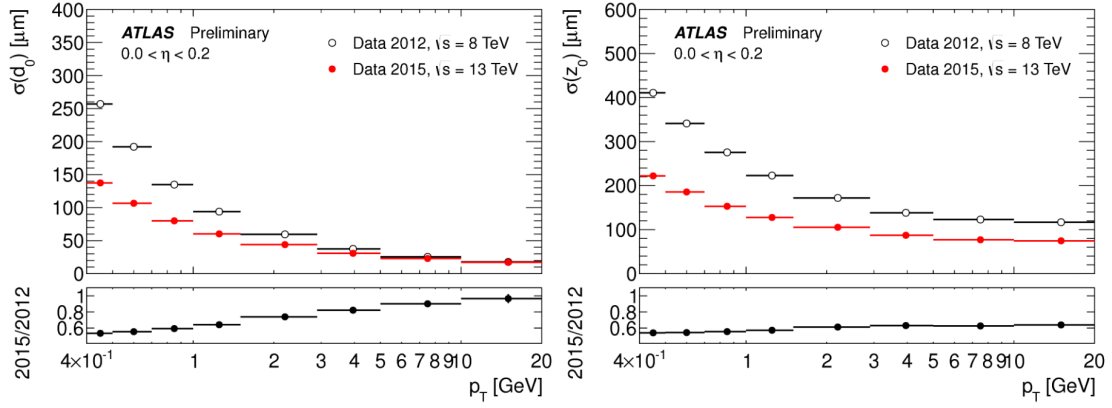


Figure 1.8.: d_0 (left) and z_0 resolution (right) as a function of p_T in ATLAS Run-1 (without IBL) and Run-2 (with IBL) [41].

The Semi Conductor Tracker (SCT) [8] contains 6 million channels on four barrel layers and 9 end-cap disks at each end. The detection is done by 4088 modules composed of two pairs of single-sided silicon micro strip sensors. Each sensor contains 765 strips. An external radiation-hard chip, named ABCD amplifies, digitizes, and compares the signal to a threshold. The signal is also saved in a 132 cell pipeline waiting a trigger signal to transfer it or discard it. Figure 1.9 shows the hit efficiency for each sector of the semi conductor tracker.

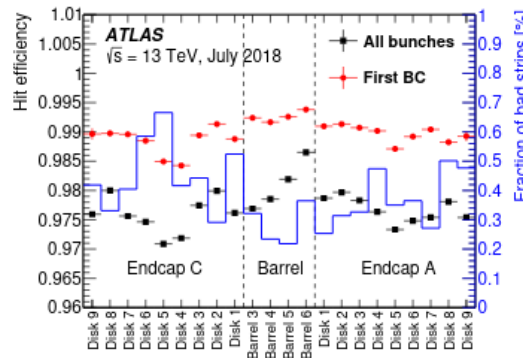


Figure 1.9.: Hit efficiency of the SCT detector measured in a typical pp collision run in July 2018. Values measured using all bunches (only first BC) are shown by the black squares (red dots). In only first bunches there is no pile-up effect from a past bunches collision. The hit efficiency averaged over each SCT barrel layer and endcap disk. The fraction of bad strips is superimposed [14]

The Transition Radiation Tracker (TRT) [11] detector contributes to the tracking system and is also designed to differentiate mainly electrons from pions. It contains

350 read-out channels composed of a 0.03 mm diameter gold-plated tungsten wire inside a straw tube of 4 mm diameter. 50000 straws of 144 cm, read on both side, compose the barrel and 250000 straws of 39 cm compose the two end-caps. A first front-end chip, the ASDBLR(Amplifier-Shaper-Discriminator with Baseline Restoration) amplifies, shapes and transmits signals to a DTMROC(Digital Time Measurement Read-Out Chip) digitization chip. This chip digitizes the 75 ns signals to triggered events with a 3 ns time resolution and compares it to two threshold to produces the timing data. The hit resolution of the TRT is shown in figure 1.10.

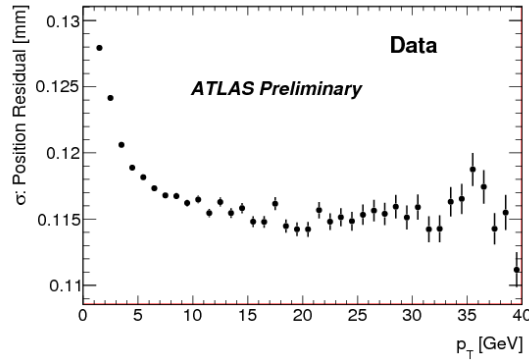


Figure 1.10.: Position residual width as a function of track P_T for the TRT barrel. Residuals rise at lower P_T due to increased scattering of the track.

During the Phase 2 upgrade the actual inner detector will be fully replaced by a new tracker, the new Inner Tracker (ITk) The ITk will be composed of a Silicon Pixel Detector close to the beam and interaction point surrounded by a silicon strip detector. This new detector will reuse many of the services (low and high voltages, cooling and dry gas system) of the existing inner detector.

1.2.2. ATLAS Calorimeters

The calorimeters are dimensioned to stop all particles but muons and neutrinos and are in charge of measuring their deposited energies. Two different types of calorimeters are installed in ATLAS, the Liquid Argon (LAr) calorimeters and the Tile calorimeter. The LAr calorimeters, with the electromagnetic layers, identify electrons and photons and measure their energies. The combination of the Tile and LAr calorimeters allow to identify and measure the energies of the jets and to measure the missing energy.

1.2.2.1. ATLAS Liquid Argon Calorimeter

Surrounding the tracking system is the The ATLAS Liquid Argon (figure 1.11) sampling calorimeter for the electromagnetic, jet, and the forward calorimetry. It is composed of four sub-detectors: electromagnetic barrel, electromagnetic end-caps, hadronic end-caps, and forward calorimeters.

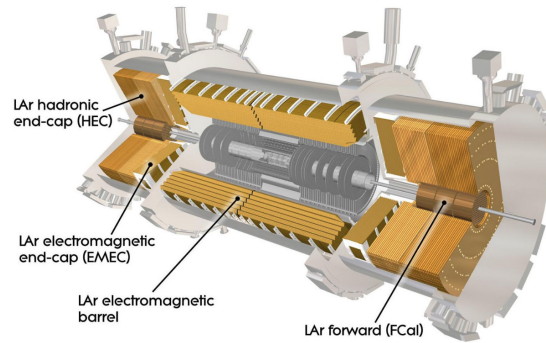


Figure 1.11.: LAr detector view.

All these detectors use liquid argon as active medium and are placed inside three cryostats. The end-caps cryostats are able to move for maintenance purposes to access the barrel sections. For the electromagnetic detectors a lead absorber is used, while copper is used for the hadronic end-cap, and copper/tungstene is used for the forward.

The electromagnetic detectors are composed of lead accordion-shaped absorber with 3 layers in depth from the interaction point as shown for the barrel in figure 1.13. The first layer has a strip geometry, the strip direction following the ϕ direction, the second a square geometry and the third layer cells are rectangle shaped, the bigger in the η direction. On the photo of the end-cap module the inner wheel of the module is clearly seen since the accordion folding angle is different.

A presampler detector is located in front of the barrel detector as well as in front part of the end-cap precision region. These presampler layers (layer 0) allow the detection of the signals induced by particles starting to deposit their energy in the matter (cryostat wall, solenoid...) in front of the calorimeter layer (1,2,3).



(a) Photo of an electromagnetic end-cap module, (b) Schematic of a cell of electromagnetic calorimeter [1]

Figure 1.12.

As shown in figure 1.12, two absorbers are separated by two honeycomb paper separators around the electrode. This electrode is composed of three layers of copper separated by kapton insulators. The two outer layers are at high voltages and the inner layer is for the readout. A capacitive coupling exists between the high voltage layers and the readout layer, mirroring the AC signals of the high voltage layer onto the readout layer.

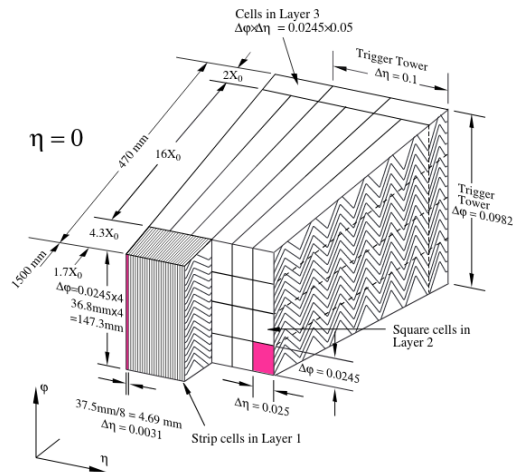


Figure 1.13.: LAr barrel cells schematic

The Hadronic end-cap calorimeters (HEC) is a sampling calorimeter using copper as absorber and liquid argon as active medium. The HEC is composed of four wheels, two in both end-cap cryostats. The two wheels are named front and rear HEC wheel, the front being close to the interaction point. Each wheel is composed of 32 modules.

The front modules are composed of 25 copper absorber plates separated by several honeycomb separators and electrodes forming readout cells. The rear modules follow the same structure with 17 absorber plates. The schematic of a HEC module and of its readout cells is shown in figure 1.14.

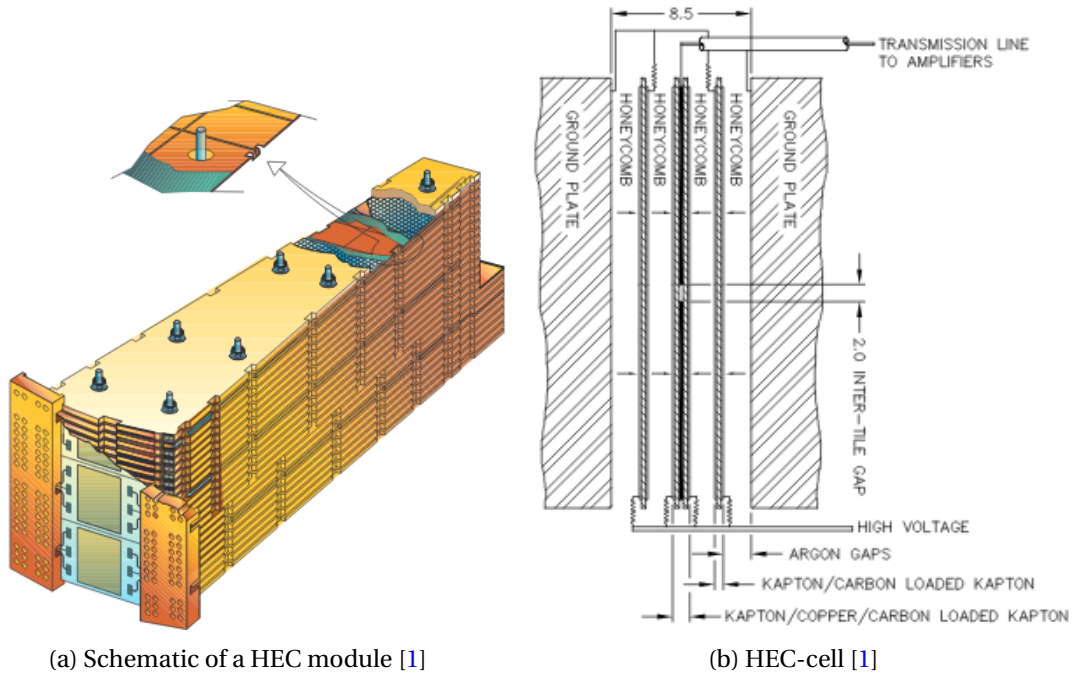


Figure 1.14.: Schematic of the HEC cell and module

All those detector cells are based on the same principle. Particles crossing the liquid argon ionize it. The amount of ionized particles depends on the energy of the primary particle which by passing through the absorbers creates a shower of secondary particles. Each of the charged secondary particles will ionize the liquid argon along its track. The ionized particles (mainly e^- , Ar^+) drift along the electric field produced by the high voltage. This drift creates an electric current on the high voltage layers. This current is mirrored on the readout electrode layer. The signal read on the readout layer has a pulse shape shown in figure 1.15. The height of the pulse is linked to the amount of ionized particles, so to the energy of the primary particle.

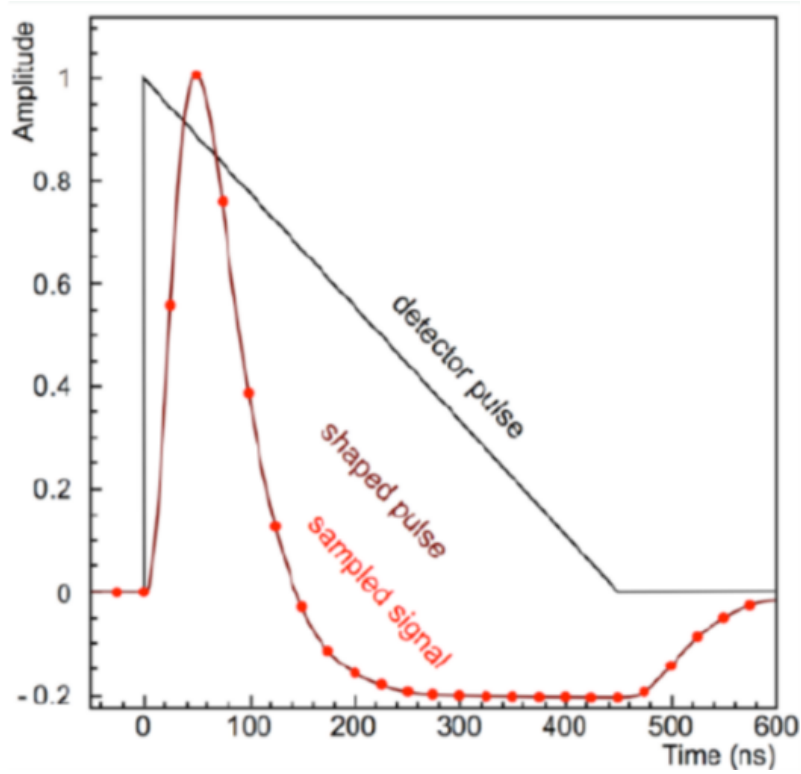
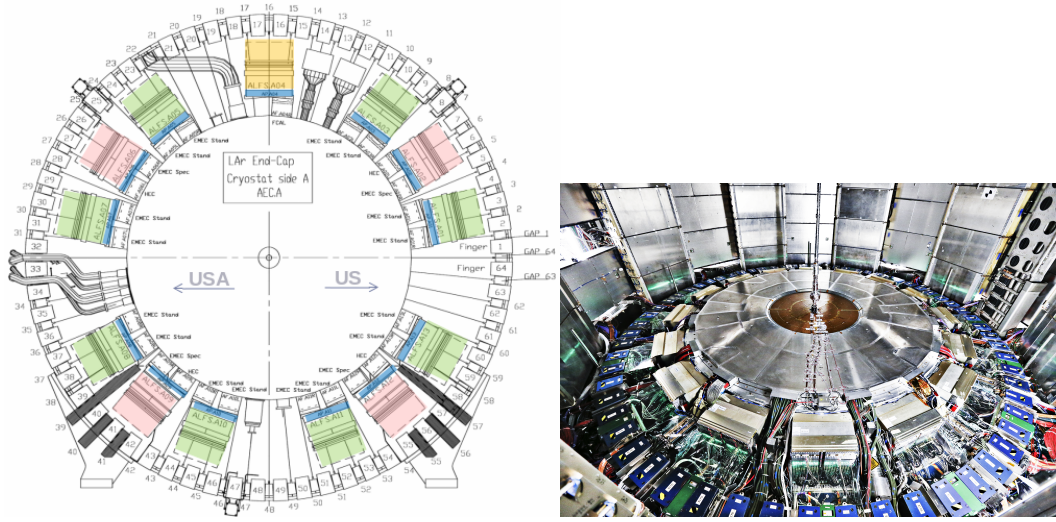


Figure 1.15.: Shape of the pulse.

The signals coming from the readout electrode are propagated to the front-end crates through copper cables. The Front end boards (FEB) receive these signals routed out of the cryostat through feedthrough flanges to the base plane of the crate. The crates are installed for the barrel at the two end of the cryostat . The end-caps and forwards crates are located on the end-cap cryostats, on the side farthest from the interaction point. A wheel is formed with, on top the FCAL crates, the end-caps crates forming a wheel with four special crates receiving both (outer and inner) electromagnetic and HEC signals and standard crates receiving only the signals from the precision region of the outter wheel. The crates position and organisation is shown in figure 1.16. A wheel is separated into 4 regions named quadrants, each of them covering a $\pi/2$ region.



(a) Schematic of the crates on the endcap. The top crate (in orange) is the FCAL. On each quadrant two standard crates (in green) surround the special one (in pink).

(b) Photo of the crates of the barrel[25]

Figure 1.16.: Crates position.

The LAr readout is composed of several boards as represented in figure 1.17. Two systems receive data from the LAr calorimeter, the ATLAS readout and the level 1 trigger.

The ATLAS readout path is composed of two boards: the Front end board (FEB), installed on the front end crate and the Readout driver (ROD) installed in the USA15 counting room. The front end board amplifies, shapes and digitizes the ionization signals for each readout cell. The digitization is done only for event selected by the trigger. Signals are then sent to the ROD board that computes the energy and time from the digitized signals. The computed time and energy are finally sent to the data acquisition system.

The trigger paths contain two boards: the Layer Summing board (LSB) and the Tower builder board (TBB). The layer summing board is a mezzanine board installed on the FEB. It sums the amplified signals of several cells of the same layer and shapes them. The analog sums are sent by the LSB to the TBB board installed in the same Front end crate. The TBB sums the signals to create a region of interest for the trigger system. The trigger tower (TT) regroups cells in a $\Delta\eta \sim 0.1$ and $\Delta\phi \sim 0.0982$ region. This signal is sent through analog cables to the level 1 trigger system installed in the USA15 counting room. The trigger decision is sent to the front end crate by the TTC system. On the crate, the trigger decision is received by the Controller board. The controller board propagates this decision to all FEB so they can digitize the corresponding signal sampled in analog memories.

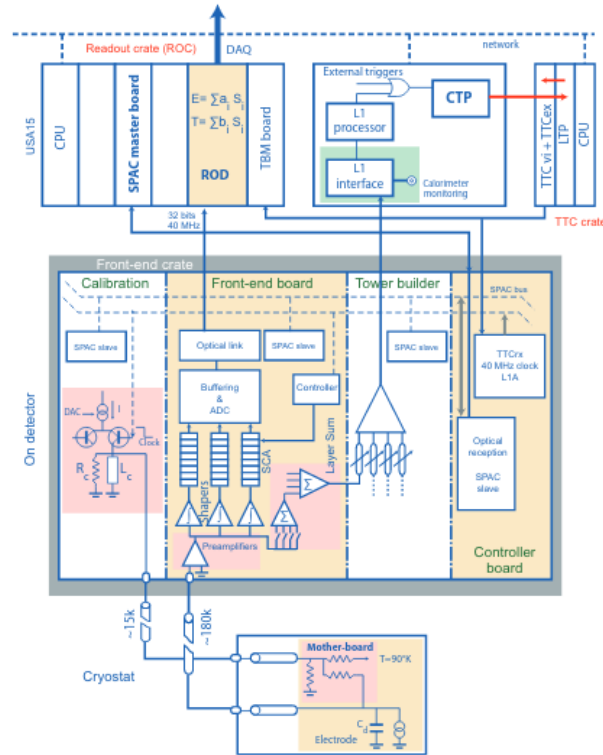


Figure 1.17.: Block schematic of the readout of the ATLAS LAr detector[1].

To keep good performance after the upgrade of the LHC to the HL-LHC, the LAr calorimeters need to be upgraded. The detector itself is not modified but the full electronics readout will be exchanged and improved. Figure 1.18 shows the upgraded electronics. The upgrade is done in two phases, the phase 1 is currently being commissioned and the phase 2 will be installed in 2025. The phase 1 focuses on the upgrade of the trigger readout path. The phase 2 upgrade will improve the main readout.

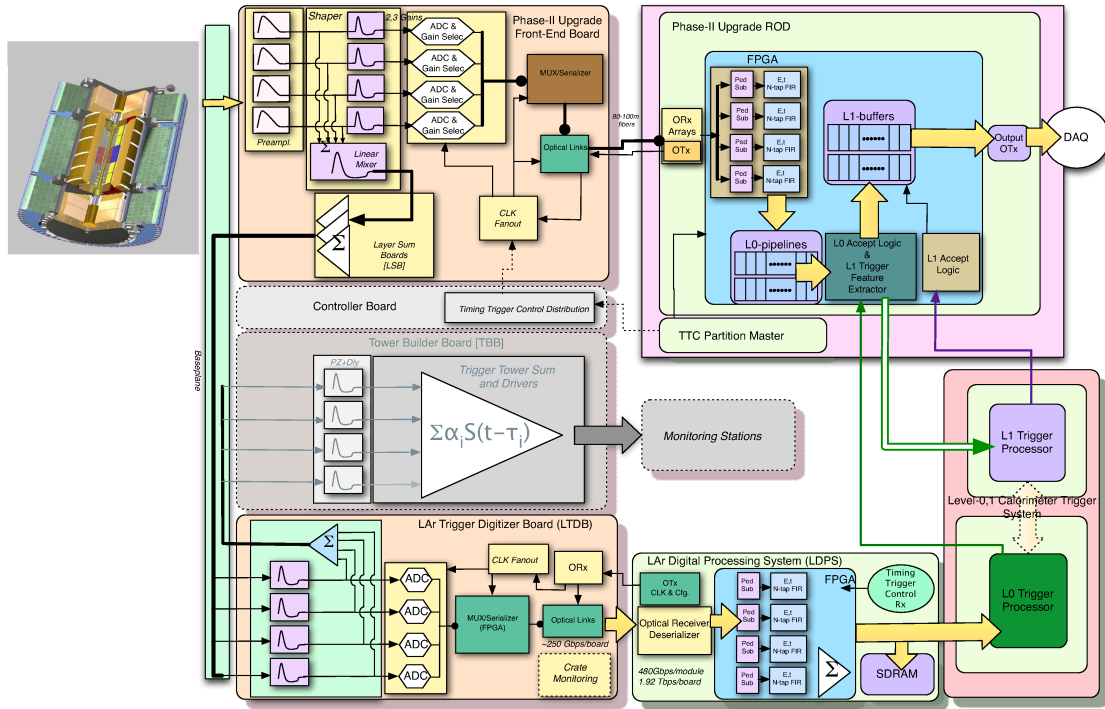


Figure 1.18.: Schematic of the LAr readout after the different upgrade phases.

1.2.2.2. Phase I upgrade of the liquid argon calorimeter

The phase 1 trigger upgrade increases the granularity of the data sent to the L1 trigger ten-fold. The trigger towers are splitted into 10 super-cells. The super cells (SC) regroup 4 detector cells for layers 0 and 2, and regroup 8 detector cells for layers 1 and 3. To create these super cell signals, the layer summing board have been exchanged and a new front end board, named LAr Trigger Digitization Board (LTDB) have been installed in the front end crate. The LTDB digitizes the super cells signals sent by the summing board and sends them to the LAr Digital Processing System (LDPS). This system is composed of several boards named LAr Digital Processing Board (LDPB). The LDPS computes the energies of the super cell from the sampled digital values coming from the LTDB and sends them to the first level of the trigger system. Chapter 2 details the LTDB and its control and monitoring system.

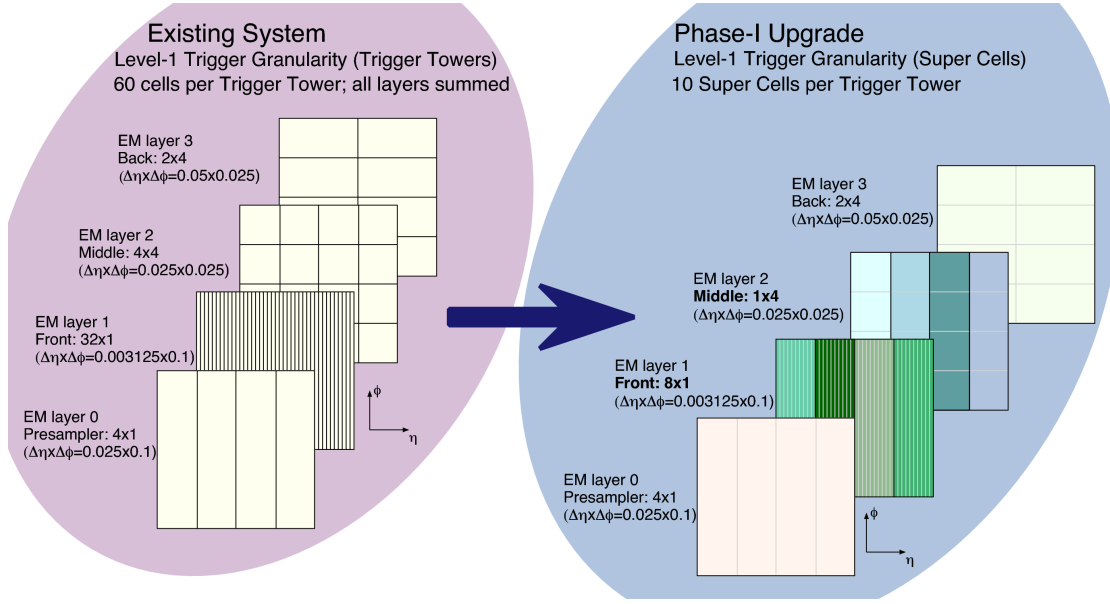


Figure 1.19.: Geometrical representation in η, σ space of an electromagnetic trigger tower (TT) in the current system, where the transverse energy in all four layers are summed (left) and of the super cells proposed for the Phase I upgrade, where the transverse energy in each layer is retained in addition to the finer granularity in the front and middle layers (right). Each square represents an area of size $\Delta\eta \times \Delta\Phi = 0.1 \times 0.1$.

1.2.2.3. Phase II upgrade of the liquid argon calorimeter

During the phase 2 upgrade, the Front end boards will be exchanged and replaced by the Front end board two (FEB2). The backend ROD boards will be exchanged by the new LAr Signal Processing (LASP) board. The FEB2 will digitizes at 40 MHz the signals with a increased precision (2x16bits) compared to the FEB (100 kHz 3x12 bits) and the LASP will compute and reconstruct the energies from the digitized signals. Chapter 2 details the energy reconstruction in the LASP. The analog buffering of the data present on the FEB will not be present in FEB2. The buffering will be digital and will be done by the LASP board.

1.2.2.4. ATLAS Tile Calorimeter

The ATLAS tile calorimeter is located around the LAr barrel and end-caps. It measures the energy deposited by the particles not stopped by the LAr calorimeter and contributes to the jet and missing energy reconstruction. These particles are the hadronics ones. The detector is composed of steel absorbers and plastic scintillators. In total 420 000 tiles of plastic scintillators compose the detector. Particles passing trough the scintillators emit photons. The photon emmitted in the scintillators are readout by photo-multiplier tubes creating the transmitted electronic signal of the detectors. Figure 1.20 show the concept of the tile calorimeter.

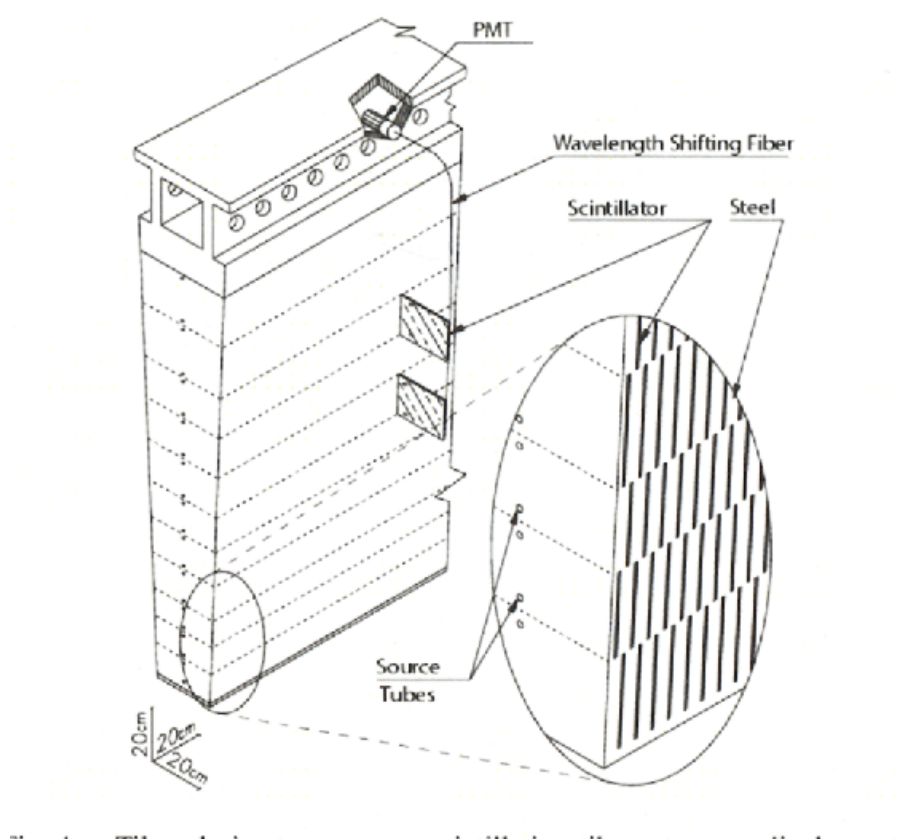


Figure 1.20.: Tile calorimeter concept: scintillating tiles set perpendicular to the beam axis [36].

1.2.3. ATLAS Muon Spectrometer

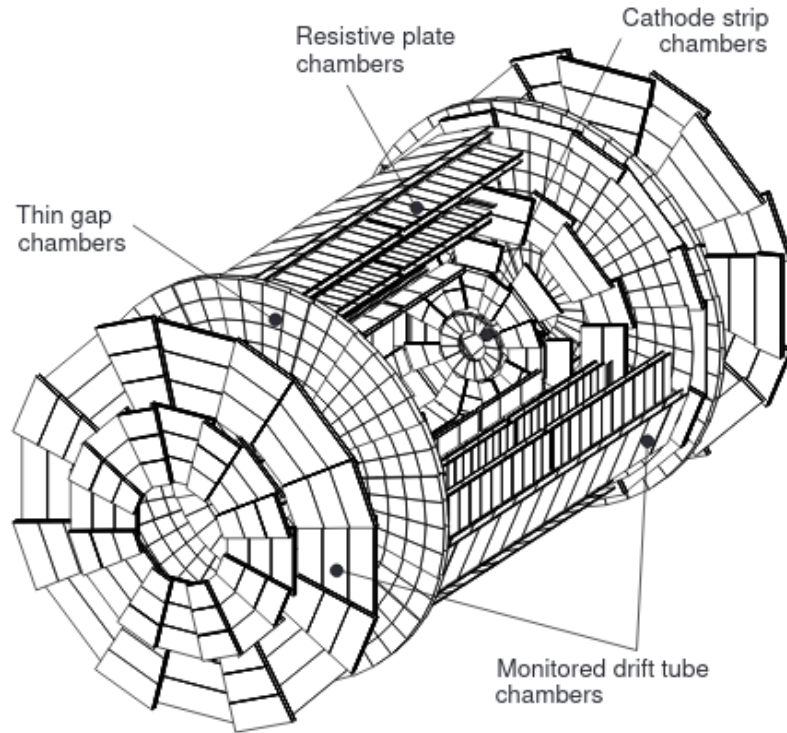


Figure 1.21.: Three-dimensional view of the muon spectrometer instrumentation indicating the areas covered by the four different chamber technologies.

The ATLAS muon spectrometer reconstructs the track and momentum of the muons going through it. It is composed of several muon chambers and a toroid magnet. The magnet provides a field that bends the muon trajectories in the r - z plane allowing to measure their momentum. The muon spectrometer is located around the calorimeters. Figure 1.21 shows this system. Each muon emitted during the collision crosses three layers of muon detectors. Four different technologies of muon detectors are used in ATLAS: resistive plate chambers, thin gap chambers, cathode strip chambers and monitored drift tube chambers. Thin gap chamber and resistive plate chamber detectors are used in the trigger. The efficiency of the muon reconstruction by the muon system is shown in figure 1.22. This plot compares the efficiency of the detector data with the Monte Carlo simulation of the muon system.

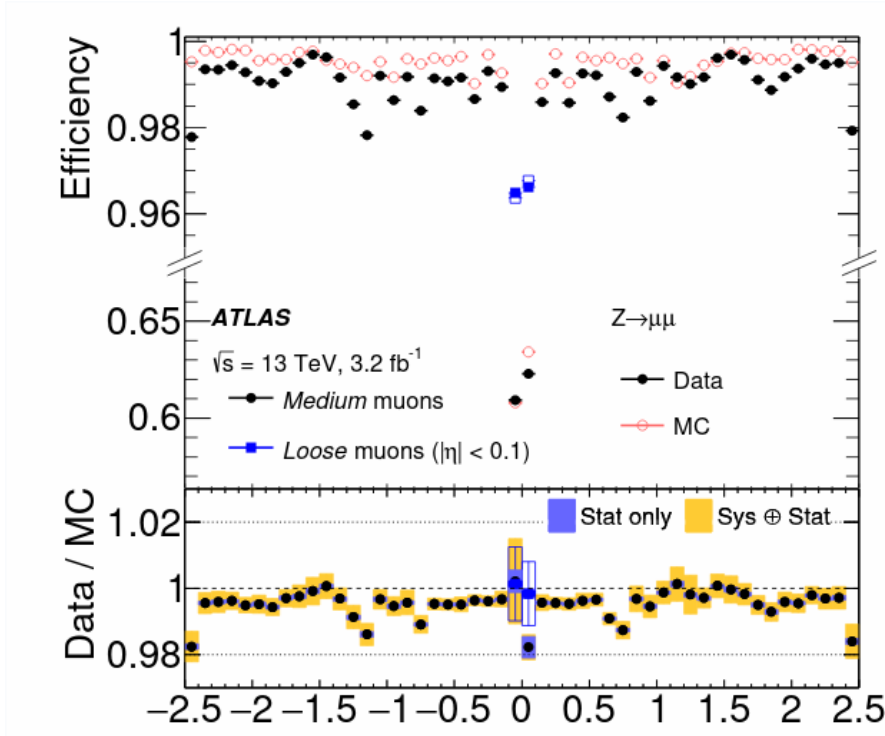


Figure 1.22.: Muon reconstruction efficiency as a function of η measured in $Z \rightarrow \mu\mu$ events for muons with $p_T > 10 \text{ GeV}$ shown for Medium muon selections. In addition, the plot also shows the efficiency of the Loose selection (squares) in the region $|\eta| < 0.1$ where the Loose and Medium selections differ significantly. The error bars on the efficiencies indicate the statistical uncertainty. The panels at the bottom shows the ratio of the measured to predicted efficiencies, with statistical and systematic uncertainties [13].

1.2.4. ATLAS trigger and data acquisition system

With a 40 MHz frequency of collision the amount of data generated by ATLAS represent several hundreds of Tera Bytes per seconds ($40 \text{ MHz} \times 1.4 \text{ MB} = 5.6 \cdot 10^{13} \text{ B/s} = 56 \text{ TB/s}$). Because of cost and technology limits it is not possible to store such amount of data. The trigger system is in charge of reducing the data rate by selecting events with physics interest. The selected events are saved on disks by the data acquisition system. These systems are named Trigger and Data Acquisition system (TDAQ). Figure 1.23 represents the different components of the ATLAS TDAQ system during run 2. There are two levels of trigger: Level-1 trigger (L1) and High Level Trigger (HLT). The L1 trigger processes data from the calorimeters and from the muon detectors. It searches for large electromagnetic energy deposits or high- p_T muon tracks. This level of the trigger system is made of built-in custom electronics. The central trigger processor (CTP) aggregates the data extracted from the L1 calorimeter and L1 muon processors. It computes the trigger decision of the Level-1 trigger named Level one accept (L1

Accept). The L1 accept maximum rate is one hundred kHz and is propagated to all sub-detectors of ATLAS. During the computation of the L1 accept, the data are buffered for all the detectors. The amount of data that can be buffered is small so the trigger decision has to be made in less than $2.5 \mu\text{s}$. The accepted events are sent by each sub-detector to the read out drivers. The readout drivers send the data to the readout system. On the readout system the event is built into data fragments for all sub-detectors. Then the High Level Trigger, based on software algorithms, treats the event fragments in a farm of computers and reduces the rates of selected events to one kilo Hertz.

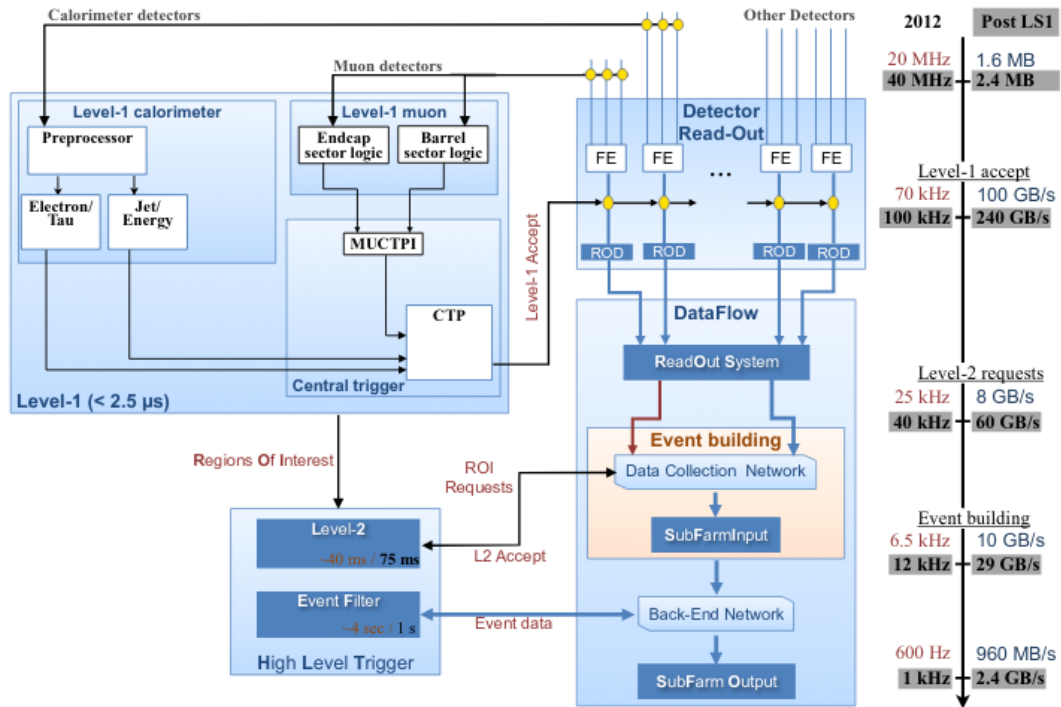


Figure 1.23.: Schematic overview of the Trigger and DAQ system in run 1 and 2.

The TDAQ system will be upgraded during the phase 1 upgrade of the ATLAS detector. Figure 1.24 shows the new system. The Level-1 calorimeter trigger system will be updated with feature extractors (FEX); the FEX will use the new digital trigger data sent by the liquid argon calorimeter upgraded readout. The Level-1 muon system will be upgraded to include a layer of the tile calorimeter in the logic of the endcap. The readout flow will be slightly modified to include the muon new small wheel(NSW). On the NSW readout, the ROD will be replaced by a Front ENd Link EXchange (FELIX) system. The High Level Trigger will be upgraded with the merge of the Level-2 Trigger and the event filter.

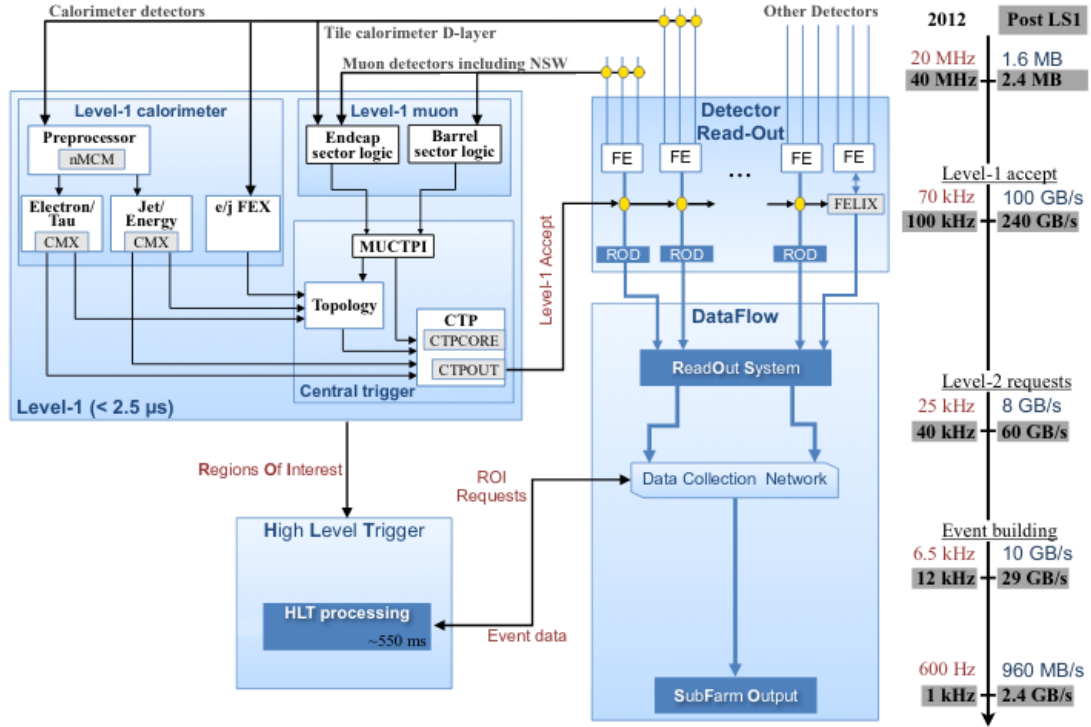


Figure 1.24.: Schematic overview of the Trigger and DAQ system after phase 1 upgrade.

For the ATLAS phase-II upgrade, the full TDAQ system will be changed to achieved an output rate of 10 kHz. Figure 1.25 is the overview of this new system. The readout of all subdetectors will be made by FELIX system. These FELIXs send the data to the data flow through data handlers. The data flow will be composed of an event builder, a storage handler and an event aggregator. The Level 1 trigger of the phase 1 will be renamed level 0. The level 0 trigger will reduce the event rate from 40 MHz to 1 MHz before the FELIX. The Event Filter trigger will reduce this 1 MHz rate to 10 kHz.

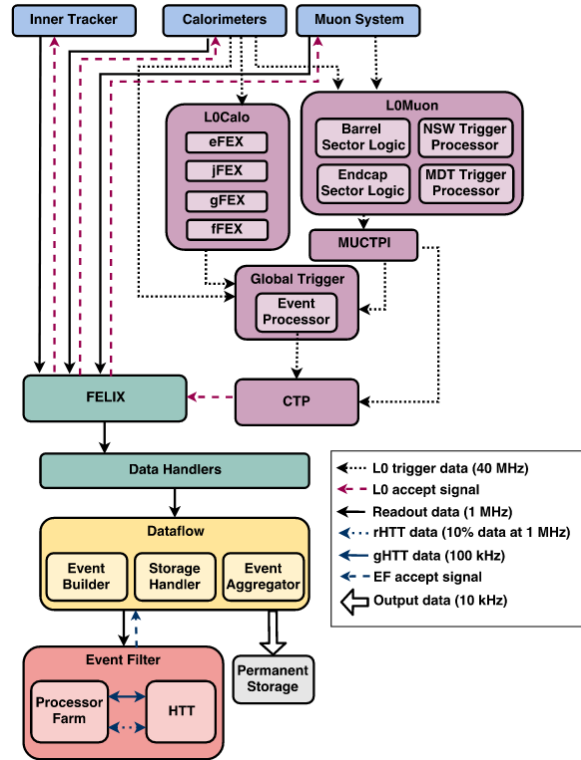


Figure 1.25.: Overall baseline design of the Phase-II TDAQ system. The black dotted arrows indicate the Level-0 dataflow from the detector systems to the Level-0 trigger system (composed of the L0Calo, L0Muon, MUCTPI, Global Trigger, and CTP) at 40 MHz, which must identify physics objects and calculate event-level physics quantities within $10\mu\text{s}$. The result of the Level-0 trigger decision (L0A) is transmitted to the detectors as indicated by the red dashed arrows. The resulting trigger data and detector data are transmitted through the DAQ system (made up of FELIX, the Data Handlers, and the Dataflow subsystem) at 1 MHz, as shown by the black solid arrows. Direct connections between each Level-0 trigger component and the Readout system are suppressed for simplicity. The EF system is composed of a processing farm and a hardware-tracking subsystem (HTT) that will reduce the event rate to 10 kHz. Events that are selected by the EF trigger decision are transferred for permanent storage

1.2.5. Control and monitoring of ATLAS

Three systems control and monitor the ATLAS detector: the Detector Security System (DSS), the Digital Control System (DCS) and the Trigger and Data Acquisition (TDAQ) system.

The DSS system is in charge of the safety of the detector. It reacts on events that can damage the detector. This purely hardware system is based on a bulk of captors and actuator installed in all critical systems of the detector. The DSS is in charge for example of ramping down the high and low voltage in case of cooling failure. It is based on interlocks taking actions without human intervention. The human intervention is needed after the problem is resolved to rearm the DSS.

The DCS is in charge of basic controls and monitoring of the detector. The monitoring is focused on temperatures, currents, fluid flow, pressure, and voltages values. As the DSS system it computes alarms but at a less critical level. The DCS alarms can have several criticality levels and are propagated to system experts by mail or SMS. The DCS system, in charge of the basic configuration of the detector, includes all sub detectors and also the power system, the magnet system, and the cryogenic system. The DCS system is always running and provides a graphical user interface(GUI). Figure 1.26 represents this graphical interface. It is designed to provide a complete, easy to read and clear status of the detector. This GUI that has several layers is always displayed in the control room of ATLAS.

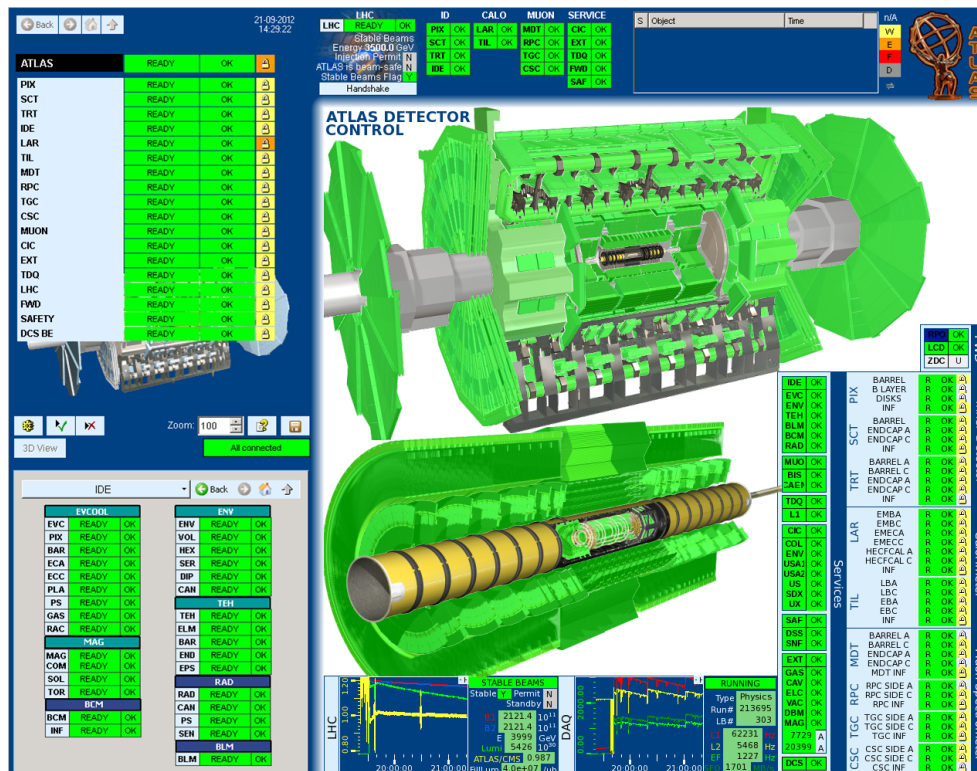


Figure 1.26.: Atlas DCS user interface during a regular LHC fill for luminosity production with p-p-collisions. It shows an overview of the state of all detector components.

The TDAQ system has its own user interface which control and monitor the data-taking and event-selection of the detector. The application configures all the sub detectors and systems for the data acquisition. The data application is organised in runs. These runs can include the full detector or part of it. A part of the detector able to autonomously take and save data is called a partition. Four types of runs are defined for ATLAS: Physics, detector calibration with beam, detector calibration without beam, commissioning runs. Depending on the type of run, the run control application configures the detector acquisition and trigger systems. The run control application is controlled by a graphical interface shown in figure 1.27. This interface is used by the run coordinator in the control room to configure the detector. The interface also displays monitoring of the data acquisition system.

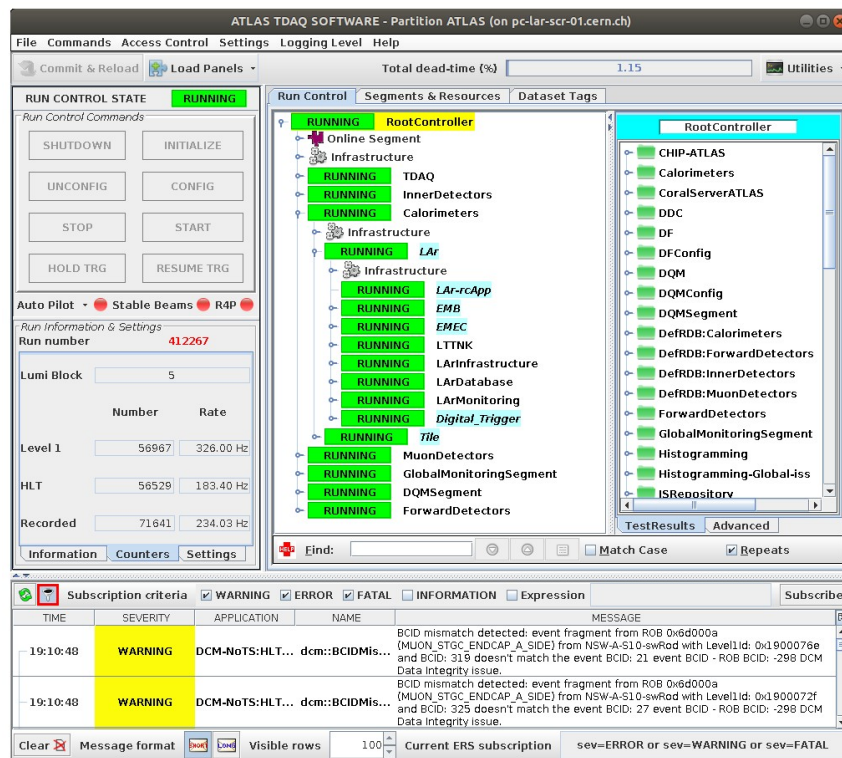


Figure 1.27.: Atlas run control application during a run. On the top left the button to control the state of the run. In the main window the ATLAS tree is developed for the LAr detector. In the bottom the error window shows the different errors and warnings of the detector.

2. Control and Monitoring of the front-end boards of the phase-I upgrade

Summary

2.1	Introduction	58
2.1.1	Overview of the phase-I electronic system	58
2.1.2	ATLAS Detector Control System	60
2.2	Hardware Components	62
2.2.1	LTDB	62
2.2.1.1	Slow control path	64
2.2.1.2	Data path	67
2.2.1.3	LTDB flavors	72
2.2.2	Front-End FELIX	73
2.2.2.1	The GBT-Protocol	73
2.2.2.2	The GBT Protocol for the LAr GBTx component	75
2.2.2.3	HDLC-Protocol Layer and SCA Channel command Layer	76
2.2.3	LAr Digital Processing Blade	77
2.3	Software	79
2.3.1	Felixcore	80
2.3.2	OpcUaScaServer	80
2.3.3	OpcUaLarLtdbServer	83
2.3.4	The DCS ATLLARLTDB Wincc OA project	92
2.3.5	Trigger and Data acquisition system	97
2.4	DCS Monitoring of the LTDB boards	98
2.5	Control of the board	101
2.5.1	Control by the TDAQ Partition	103
2.5.2	Control by DCS panels	105
2.6	Calibration of the LTDBS boards	106
2.6.1	ADC calibration	107
2.6.2	Link calibration	107
2.6.2.1	LOCx2 phase shift delay calibration	108
2.6.2.2	LOCx2 input phase calibration	109
2.6.3	Mapping	109
2.6.3.1	Control fiber mapping between the FELIX and the LTDB	110

2. Control and Monitoring of the front-end boards of the phase-I upgrade –

2.6.3.2	Data fiber mapping between the LTDB and the LATOME	110
2.6.3.3	Channel mapping between ADC channels and LATOME channel	111
2.6.4	Error mitigation in the software	111
2.7	Board Installation and validation	112
2.7.1	Results	113
2.8	Conclusion	114

2.1. Introduction

The phase-I upgrade of the LAr calorimeter focuses on the development and the installation of the new digital trigger readout path. With a better granularity, this new trigger improves the performance of the trigger system.

2.1.1. Overview of the phase-I electronic system

Figure 2.1 shows the different hardware and software components of the new phase-I system. The components are parts of two largely but not completely independent systems: The Digital Control System(DCS) and the Trigger and Data Acquisition (TDAQ) System. The DCS handles basic powering and monitoring of the system. The TDAQ system has several purposes: configuration and distribution of the Timing and Trigger Control(TTC), configuration of the boards, triggering and finally data acquisition. The two systems are independent with the exception of the LTDB's Open Platform Communications Unified Architecture (OPC-UA) servers which are shared between the two.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.1. Introduction

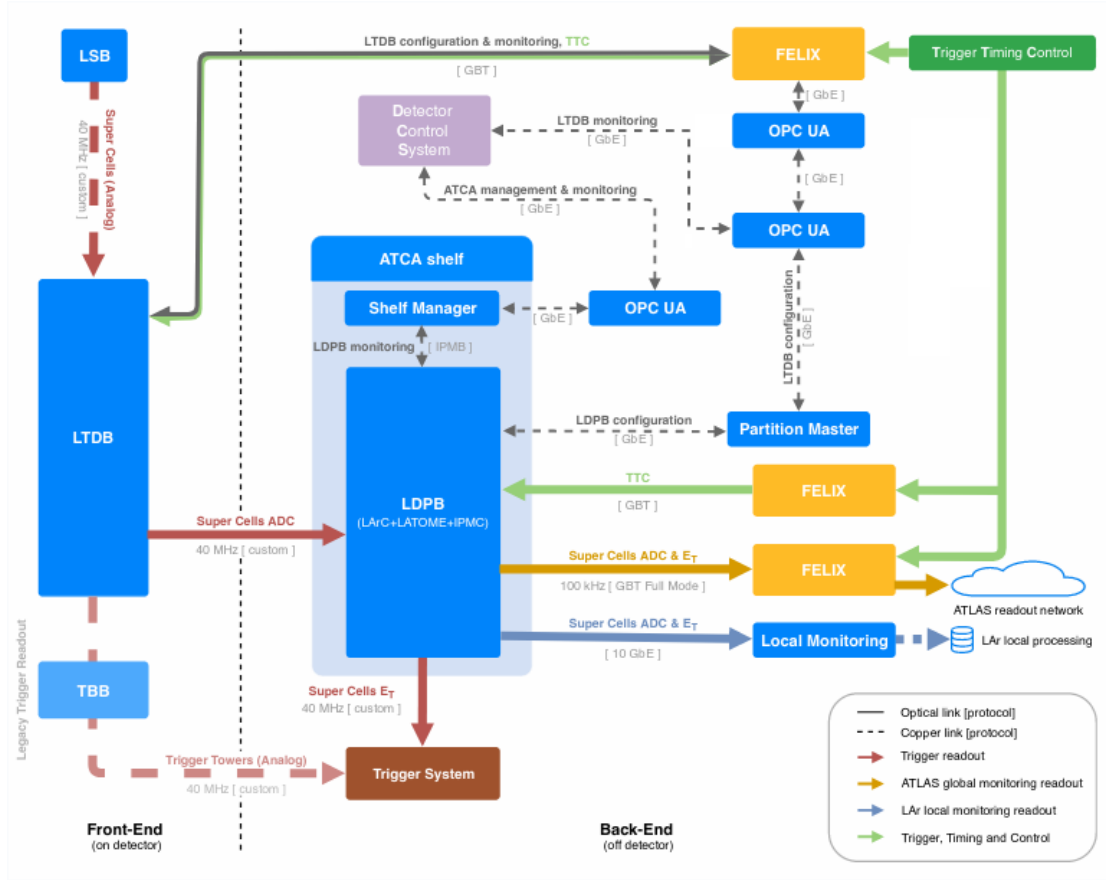


Figure 2.1.: Schematic of the LAr digital trigger system after the phase-I upgrade. This system is composed of two boards: the LTDB and the LDPB. The interfaces of these boards with the TDAQ and DCS systems are done through different FELIX boards and OPC-UA servers.

The DCS of the phase-I upgrade is divided into two sub-parts: the LDPB-DCS and the LTDB-DCS. For the LDPB-DCS, powering commands are sent to the LDPB through an OPC-UA server. The server is connected to the LDPB through an Advanced Telecom computer architecture (ATCA)[15] shelf manager. Monitoring values coming from the LDPB and from the shelf itself use the same OPC-UA server. OPC-UA protocol will be detailed in section 2.1.2.

For the LTDB-DCS the commands are sent through two OPC-UA servers. The two servers are chained in a server client structure with specific functionalities. The second server uses the FELIX-FE (FELIX Front End) board to propagate the data to the LTDB through optical links. The monitoring data of the LTDB use the same path and are propagated through the FELIX-FE to the OPC-UA servers connected to the DCS system. The OPC-UA servers, the FELIX board and the optical links are shared with the TDAQ system.

The TDAQ system configures the LTDB, the LDPB, and the TTC. The LTDB configuration is done through the two OPC-UA servers shared with DCS. The LDPB configuration uses a direct Gb ethernet connection with the LDPB boards. At the frequency of the collisions, the LTDB digitizes the super-cells electric pulses coming from the LSB and sends the digitized value to the LDPB. The LDPB computes the deposited energies using the digitized samples. These energies are sent to the first level of the trigger system and stored in a buffer of the LDPB. The level one trigger is composed of features extractor (FEX) searching for interesting objects. The objects are detected using the data from the calorimeters but also from the muon system. If such objects are found in an event, the trigger system accepts it and sends a level one accept (L1 Accept) signal through the Trigger Timing Control (TTC) system. When the LDPB receives the L1 Accept, it extracts from the buffer the ADC data and energies data of the corresponding event and sends them to the backend FELIX (FELIX-BE). The FELIX-BE sends the data to the ATLAS readout to be stored on disk.

The TTC system provides the trigger decision, the base clock of the LHC and the timing data. The base clock frequency is 40 MHz equal to the frequency of the collisions. The main timing data is the bunch crossing reset (BCR) signal. This signal is used to compute the identifier of the collision event. This identifier is added to the data sent by the LTDB boards to the LDPB boards.

The TTC data are sent to the LTDB using the FELIX-FE. The FELIX-FE also uses internally the clock and timing data from the TTC. The GigaBit Transceiver (GBT) protocol is synchronous to the 40 MHz clock and sends a packet at this frequency. This packet of data contains the TTC data, the monitoring data and the configuration from the OPC-UA server. It creates a dependency between the DCS and the TDAQ systems: the TDAQ configuration of the TTC is needed by DCS to receive data through GBT. Details of the GBT protocol and how it propagates the different types of data are available in section 2.2.2.1.

The TTC data are sent to the LDPB using a FELIX board named FELIX-TTC using the GBT protocol through an optical link.

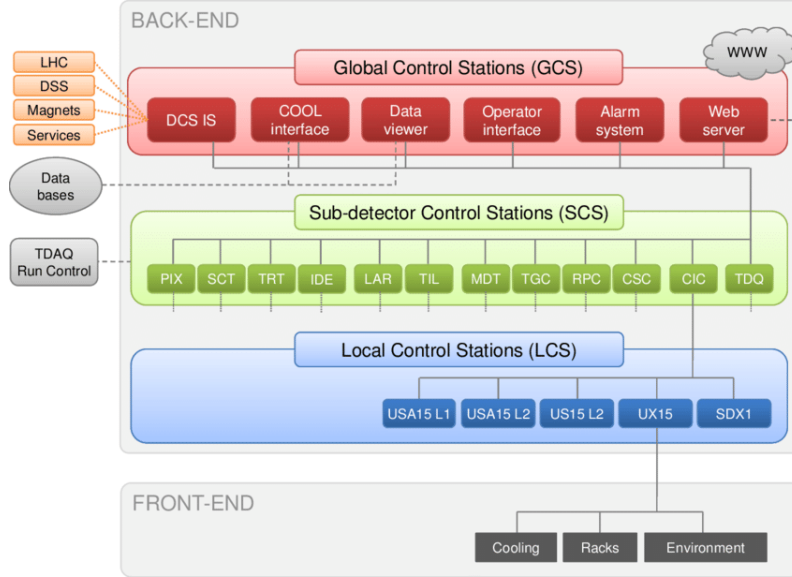
2.1.2. ATLAS Detector Control System

The ATLAS DCS supervises the full detector and provides to non experts a simple graphical interface to operate the detector. For this purpose, a large amount of procedures were implemented to control the detector, monitor its status, and alert in case of problems. The DCS is based on timestamped slow control data that monitor the status of the system. The DCS system is also able to control and configure parts of the detector components, for example high and low voltage, power of the different boards.

As seen in figure 2.2 the ATLAS DCS is organised as a hierarchy with top level commands being propagated to all sub-systems. The errors from sub-systems are propagated to the upper class hierarchy. This system relies on on a Supervisory Control

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.1. Introduction

And Data Acquisition (SCADA) software from siemens named Wincc Oa (for Windows control command open architecture).



(a) Figure A

Figure 2.2.: DCS hierarchy schematic.

The different sub-systems being heterogeneous with different hardware, a middleware software is needed to do the link between the ATLAS generic SCADA and the sub systems specifics library or protocols. The middleware consists of a transition layer, ATLAS choose the OPC-UA standard to build it. The OPC-UA standard was proposed by the OPC Foundation [24] and standardized as the international Standard IEC 62541 [19]. The object oriented design and the platform independence of this standard correspond to the need of the ATLAS architecture.

To facilitate the development, maintenance and standardisation of this middleware ATLAS DCS developed a framework for OPC-UA servers generation. This framework is called quasar [31, 39] for **Quick opc-ua Server generation framework**. This framework handles the creation of the server, logging , configuration loading, and OPC-UA address space access based on a description in the "Design" xml files. The address space is the part of the server accessible for clients. The only part that is not generated by quasar is the code specific to the sub-systems named device logic. The device logic access the specific electronics of the system through libraries named device access layer. A schematic of the quasar server components is available in figure 2.3

To read or write into the quasar server address space, clients need to connect to it through Ethernet connection and OPC-UA protocols. To facilitate the development of such clients, quasar is able to generate a library with an API named UaoForQuasar. The C++ generated library handles all the low level ethernet connection and the OPC-UA

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

protocol. The library provides Ua objects [32] (UaO) that represent the different part of the server address space. These C++ objects have methods to read or write into the server address-space. This tool is mainly used in the clients developed in C++.

Some of the OPC-UA clients, mainly DCS clients, are developed in Wincc Oa and cannot use the UaoForQuasar generated library. For these clients a specific API named fwQuasar is included in the standard framework, named JCOP, of the CERN Wincc OA team. The quasar generated servers contain several components. Figure 2.3 is a schematic of the quasar server and its components. The components are separated into 3 layers: the commercial OPC-UA toolkit, the common quasar layer, and the device specific layer. A detailed description of the quasar based OPC-UA servers used in the control of the LTDB is available in section 2.3.

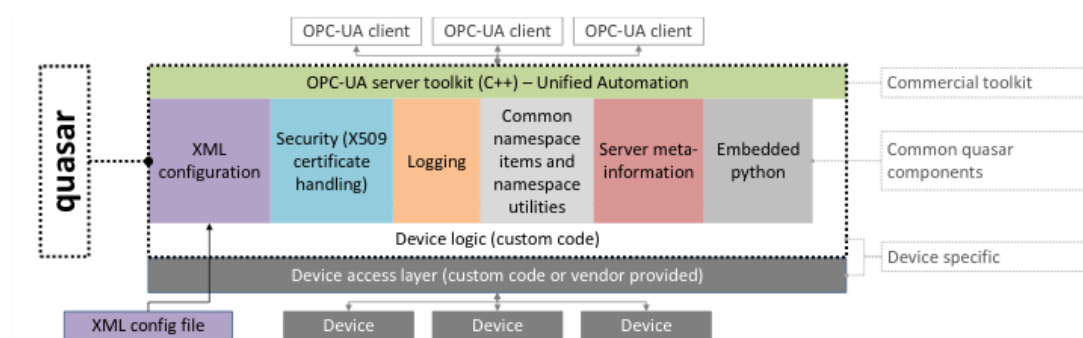


Figure 2.3.: Overview of the Quasar components[39].

2.2. Hardware Components

Several hardware components are needed to control and calibrate the LTDB boards: the LTDB board itself, the FELIX-FE, and the LDPB. The FELIX-FE does the translation between the control software and the LTDB. The LDPB is used to validate the configuration of the LTDB output links and the data set by the LTDB.

2.2.1. LTDB

A simplified schematic of the LTDB is shown in figure 2.4. The board is composed of two paths, the control path and the data path.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

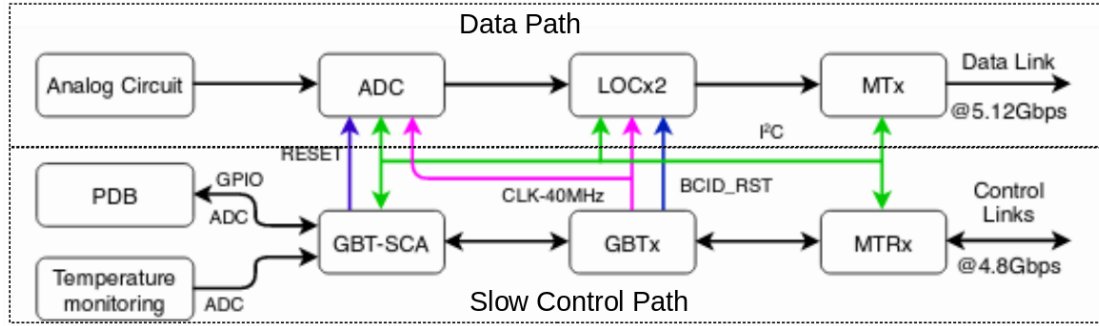


Figure 2.4.: Schematic of the concept of the LTDB. The figure shows the different components of the data path and the control path.

The data path input is the analog super cell signals coming from the analog circuit. These signals are digitized in the ADC. After being digitized the corresponding data are transmitted to the LOCx2 component. The LOCx2 encodes the digital signal which is sent by the optical transmitter (MTx) to the LDPB. More details about the data path are available in [2.2.1.2](#).

The control path, detailed in [2.2.1.1](#), receives configuration data from the OPC-UA servers, timing signals from the TTC system, and sends monitoring values to OPC-UA servers. The control path data are received and transmitted through the GBT protocol. The GBT frame is received/transmitted by the MTRx optical component and is decoded/encoded by the GBTx component. The GBT frame contains three types of data: TTC, configuration and Monitoring. The TTC data are received by the GBTx which decodes them and transmits them to the various LTDB components that needs them. The clock is transmitted to the ADCs and the LOCx2 components. The Bunch Crossing Reset signal is transmitted only to the LOCx2. The OPC-UA control commands are sent to the GBT-SCA. The GBT-SCA distributes the commands to the different components of the board. These components are part of the control and data path but also part of the Power Distribution Board (PDB). The PDB is a mezzanine board composed mainly of regulators. The PDB provides all the needed voltages and power to the LTDB.

The LTDB is shown in figure [2.5](#). The board is composed of an analog section, a digital section and a PDB. The digital section is composed of five identical parts. Each one of these parts is composed of one control path and four data paths. The four data paths are controlled by the control paths of the corresponding part. The three white connector on the bottom of the board are the rear connectors of the board. They receive the analog signals from the LSB. To provide legacy trigger tower signals, the analog signals received by the LTDB are summed in the analog part of the LTDB. The summed signals are sent through the back connectors to the tower building board (TBBs). The LTDB digital parts lie between the front plate and the analog part. The blue connectors in the front-end plate are MTP connectors containing 12

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

optical fibers each. The left connector is used for the control paths while the 4 other connectors are used for the data paths.

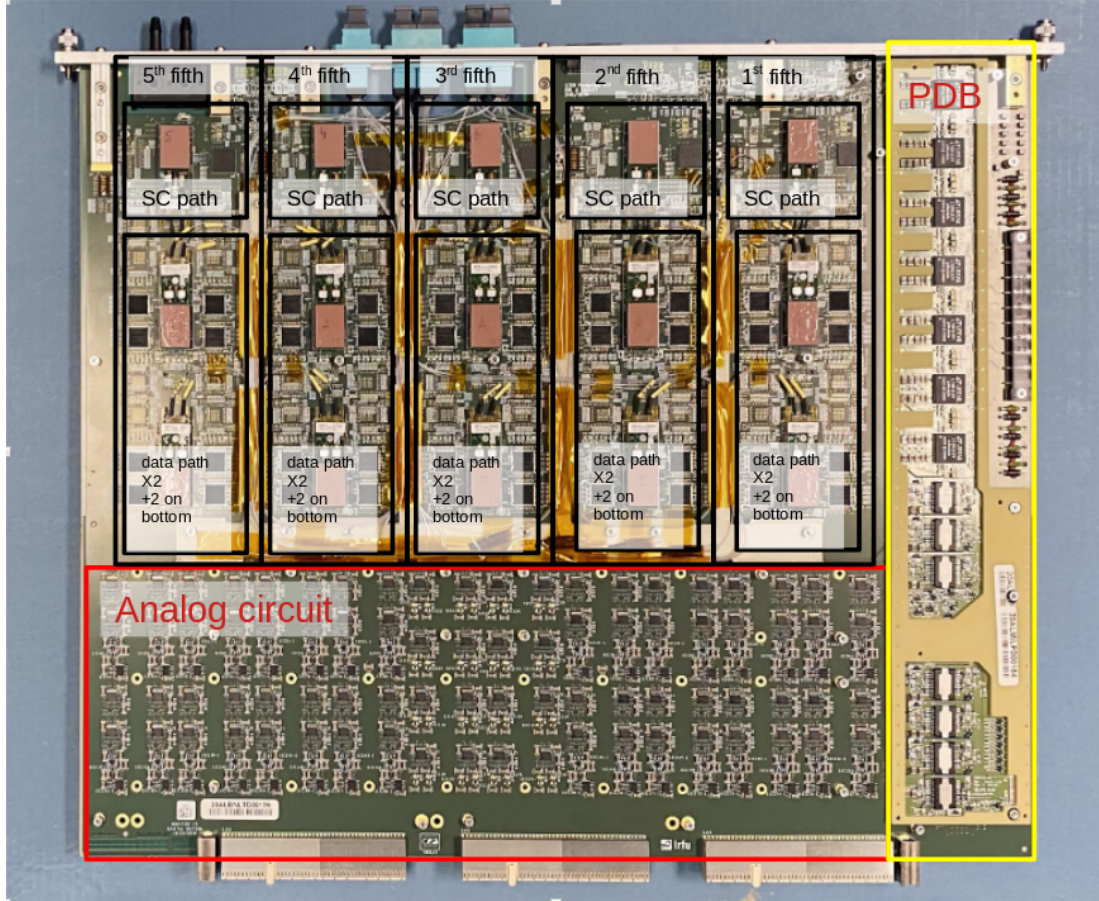


Figure 2.5.: Picture of an LTDB. On the bottom the analog part receives the signals from the LSB. These signals are propagated to the digital section. The digital section is composed of 5 identical parts, each of them composed of a control path and four data path. The board is powered by the different regulators on the PDB mezzanine board.

2.2.1.1. Slow control path

The slow control schematic is shown in figure 2.6. The communication with the FELIX is done through 5 pair (Tx/Rx) of optical fibers linked to the optical transceivers (MTRx) of the board [46]. Each pair of fibers connects a slow control path to the FELIX. The fibers are grouped in MTP 12 connectors. Details of the correspondence between the fibers in the MTP and the MTRx can be found in table 2.1. To resist radiation, these links use GBTx component and GBT protocol [28] which are radiation hard. Five GBTx components decode the GBT frames on each board. More details about the GBT protocol and its usage for the LTDB control system are available in section 2.2.2.2

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

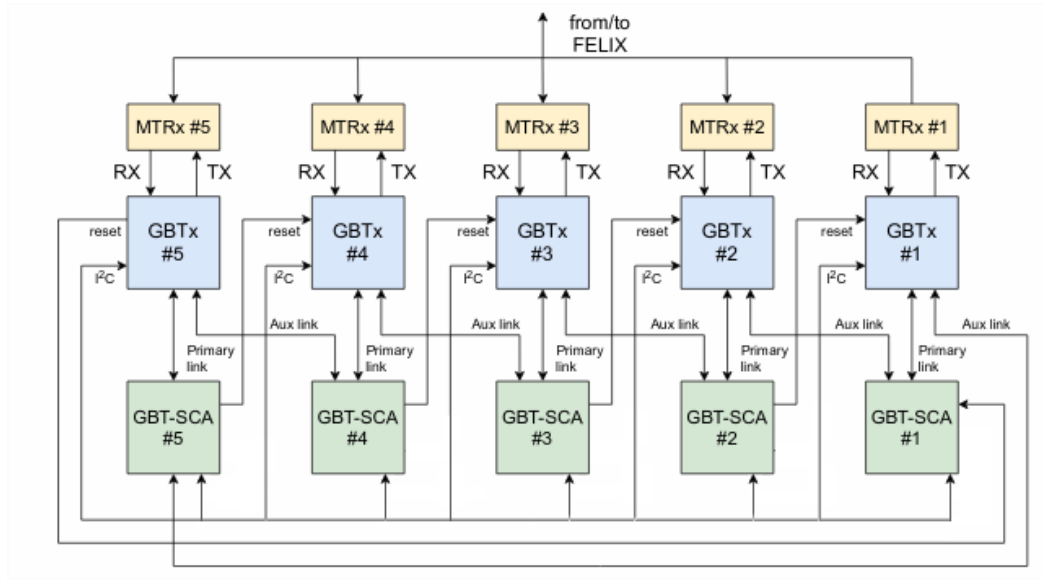


Figure 2.6.: Schematic of the slow control components of the LTDB. Each GBTx is connected to two SCAs to provides redundancy in case of issue in one control fiber or MTRx component.

Fiber in MTP	MTRx(Barrel LTDBs)	MTRx(Endcap/hec/fcal LTDBs)	GBTx	SCA (DIR)	SCA(AUX)
1	MTRx 1 Tx	MTRx 1 Tx	1	1	5
2	MTRx 1 Rx	MTRx 1 Rx			
3	MTRx 2 Tx	MTRx 2 Tx	2	2	1
4	MTRx 2 Rx	MTRx 2 Rx			
5	MTRx 3 Tx	MTRx 3 Tx	3	3	2
6	MTRx 3 Rx	MTRx 3 Rx			
7	MTRx 4 Tx	MTRx 4 Tx	4	4	3
8	MTRx 4 Rx	MTRx 4 Rx			
9	MTRx 5 Tx	MTRx 5 Tx MTRx 5 Rx	5	5	4
10	MTRx 5 Rx				
11					
12					

Table 2.1.: Correspondence between the fiber number in the MTP cable and the LTDB components. The DIR SCA is the GBT-SCA of the same part. The AUX SCA is the SCA of the consecutive part used for redundancy.

The LTDB has 5 Slow Control Adapter components (SCA or GBT-SCA)[12] , each one controlling four data paths. The SCA uses the HDLC protocol[20] received from the GBTx, detailed in 2.2.2.3. The GBT-SCA represents the only slow control access of the LTDB, it provides several interfaces for all the heterogeneous components of the board. For this it implements several general purpose input/output channels(GPIO protocol) for simple binary signals, several I2C buses for reading/writing into different registers, and several analog to digital converter (ADC) for analog input.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

The access to the SCA being critical to operate the board, redundancy has been installed to not lose the control if one optical link is not working. Figure 2.6 shows the different links between the different slow control components. The GBTx is connected to two SCAs, one through a direct connection and the second through an auxiliary connection. The SCA of the same control path is named direct SCA (SCA DIR) and the second auxiliary SCA (SCA AUX). The auxiliary SCA is for the consecutive slow control path. It creates a cyclic redundancy with each SCA accessible from two GBTx.

To access the SCAs two components are critical and need to be configured: the MTRx and the GBTx. The GBTx configuration is fixed and loaded during reset. To operate this reset the SCA must be accessible and the GBTx must not be in the path of the reset command to avoid a blocking situation. To have the GBTx not in the path of its own reset, the AUX SCA is used to control the reset.

The MTRx configuration is done through an I2C connection with the SCA of the same control path. If the direct link between the GBTx and the SCA is used, the MTRx is in the path of its own configuration. If the MTRx is not well configured, the communication with the SCA can be lost and the reconfiguration of the MTRx becomes impossible. To solve this issue the auxiliary link between the SCA and the GBTx is used by default. In this case, the MTRx is not in the path of its own configuration.

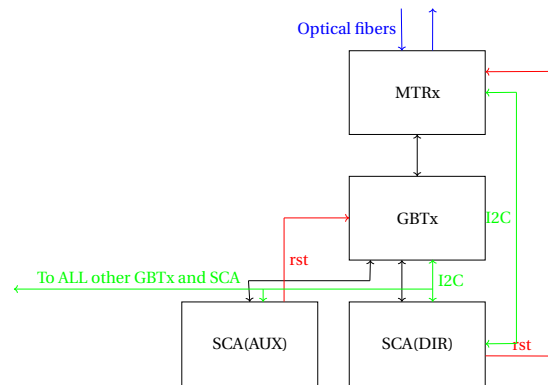


Figure 2.7.: Schematic of the LTDB Control path components and signals used for the configuration between them. The MTRx reset signal and its I2C bus is handle by the DIR SCA. The GBTx reset is handled by the AUX SCA. All the GBTx use the same I2C buses were all SCA are connected as master.

As shown in figure 2.7 there is a unique GBTx I2C bus that serves all SCAs and GBTx. On this bus the GBTx are slaves and the SCAs are masters. It is important to notice that the SCA does not support multi master mode so only one SCA has to talk to the bus at a given time. By default only the first SCA is used as master to avoid clashes between SCAs on this bus . The I2C master interface of the other SCAs is disabled.

2.2.1.2. Data path

Figure 2.8 shows the schematic of the LTDB data-path. Each data path of the LTDB is composed of four ADCs, one LOCx2 and one MTx chip. The ADCs digitize the signals of four channels of the board. The LOCx2 encodes the ADC data into two LOCic frames. On each frame the LOCx2 component adds timing data to identify the collision event corresponding to the ADCs data. The encoded frames are transmitted on two optical fibers by the MTx component.

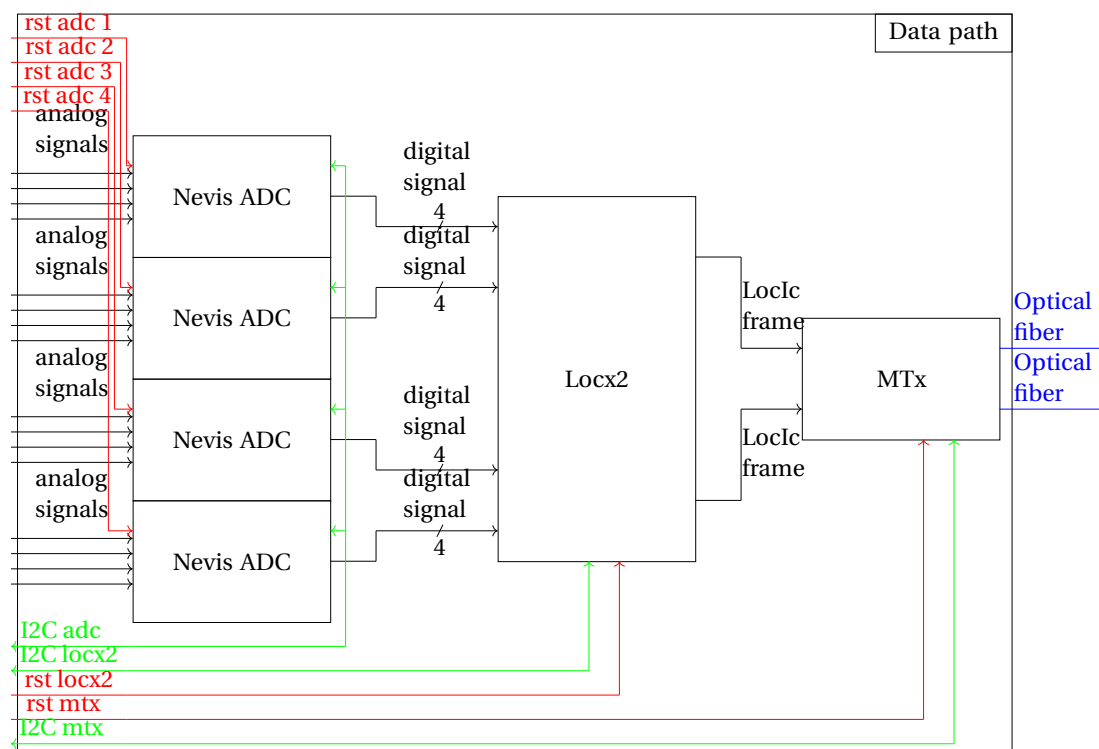


Figure 2.8.: Schematic of the LTDB Data path. Four ADC digitize the sixteen analog inputs. The LOCx2 component encodes these data into a frame that is sent to optical fibers by the MTx component.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

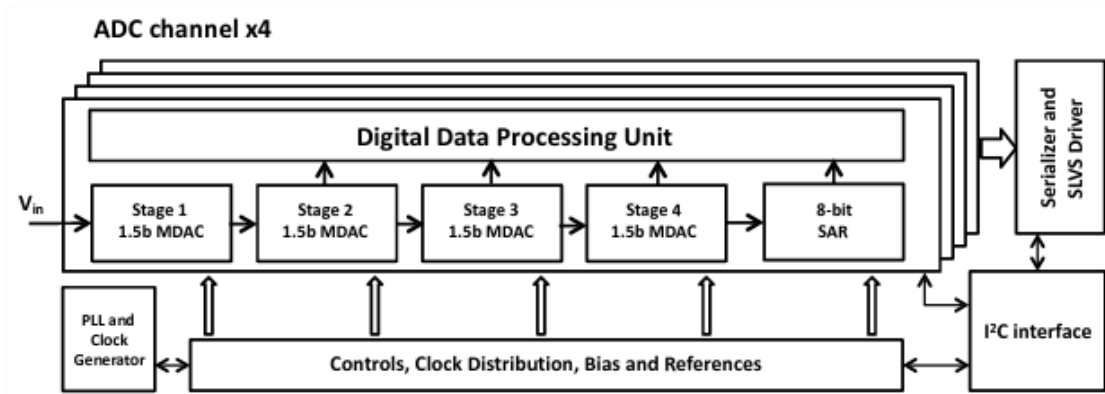


Figure 2.9.: Block diagramm of a Nevis ADC [9].

The Nevis ADC is composed of four identical channels as seen in figure 2.9. These channels implement four stages of 1.5 bits Multiplying Digital-to-Analog Converter (MDAC) followed by 8-bit Successive Approximation Register (SAR) analog digital converters [22]. The results of each ADC stage are sent to the Digital Data Processing Unit (DDPU). The DDPU aggregates the output of the different stages to form a 12 bits output. The reconstructed output of each MDAC stage, represented in figure 2.10, is not perfect and contain jumps. The origin of the jumps is a capacitor mismatch inside the MDAC stage. To correct this effect, the DDPU applies a digital correction. The coefficients of this correction are obtained through a calibration. The DDPU can be configured to send to the serializer four different types of data: the raw value without correction, the corrected output, a fixed pattern or the data needed for calibration. An I2C interface is used for the configuration of the ADC and a PLL provides the different needed clocks.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

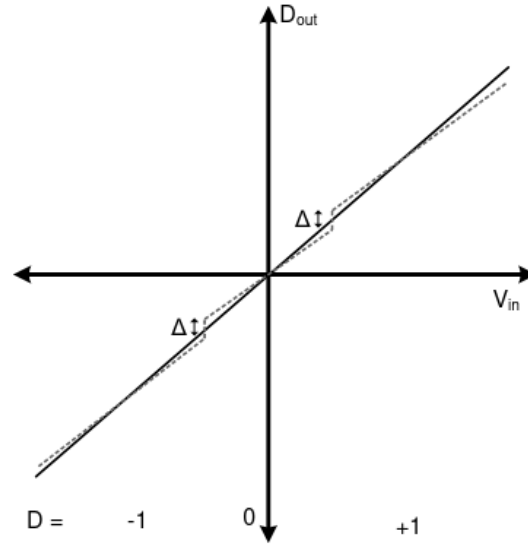


Figure 2.10.: The reconstruct output of the MDAC stage are shown in dashed lines. The solid line represents a perfect reconstruction. The real reconstruction is not perfect, the slope is lower indicating a lower gain and there is jumps in the curve.

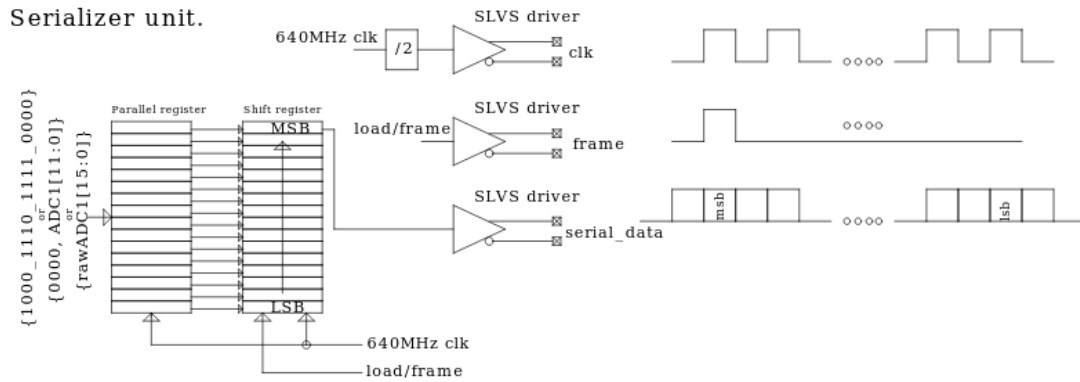


Figure 2.11.: Schematic of the ADC serializer[22]. The parallel to serial conversion is done through parallel and shift registers.

The serializer of the ADC, shown in figure 2.11, is composed of a parallel register filled at 40 MHz with 16 bits words by the DDPU. The 16-bits words are composed of the twelve bits output of the DDPU and four bits at 0. This register is transferred to a shift register that is read at 640 MHz for serialisation. In total three signals are transferred to the LOCx2:

- serial_data(DATA): One serial data bus of 1 bit width at 640 MHz for each of the channels of the ADC.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

- clk(SCK): Signal clock at 320 MHz. Both edges are being used to create a clock at 640 MHz.
- frame(FCK): Frame clock or frame delimiter indicating the MSB of the frame.

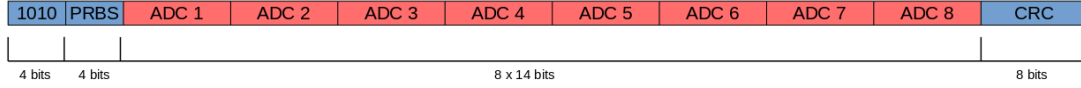


Figure 2.12.: LocIc Frame. The frame is composed of one header, one PRBS field, 8 ADC data field and one checksum field.

The LOCx2 [45] component is responsible of encoding the LOCic frame seen in figure 2.12. This frame has 128 bits composed of 4 bits of header, 4 bits of Pseudo Random Binary Sequence (PRBS), 112 bits of data and 8 bits of Cyclic Redundant Checking (CRC). The PRBS encodes the Bunch-Crossing IDentification (BCID) of the corresponding data. The data contain eight fourteen-bits ADC values coming from two different Nevis ADCs. The CRC is a checksum to detect errors of transmission based on polynomial division[38].

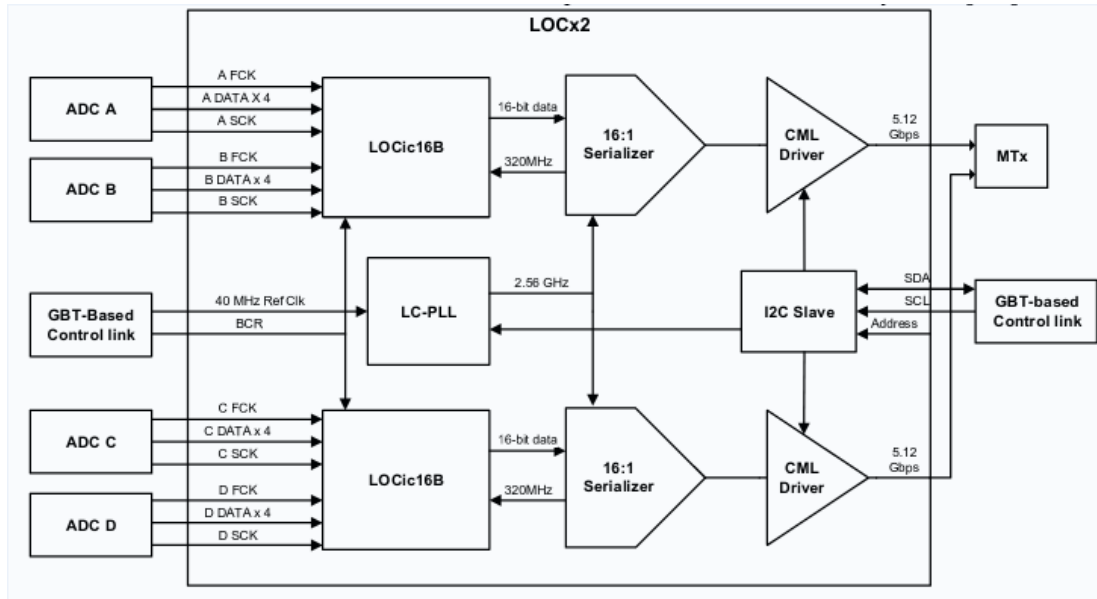


Figure 2.13.: Schematic of the LOCx2 component. Two LOCic encoder create the frame transmitted to the serializers. The CML drivers drive the bus to send this frame to the MTX at 5.12 Gbps. The components are controlled by an I2C slave interface. The LC-PLL provides all the clocks needed by the various components.

The LOCx2 (figure 2.13) is composed of two LOCic blocks, each one encoding one frame from two ADCs, a PLL providing the 2.56 MHz clock to the serializer, an I2C

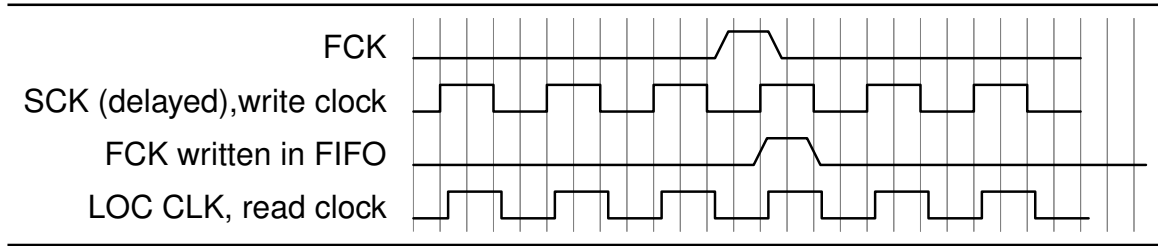


Figure 2.15.: Chronograph of LOCx2 synch FIFO

rising/falling time of the FCK and must mitigate the jitter of the LOC clock. The jitter is a Gaussian random variation from the true periodicity of the clock signals. If the FCK signal is not correctly latched, the PRBS generator is not able to correctly encode the BCID and the LOCic frame is corrupted.

The LOC clock is generated in the LOCx2 component in the serializer by a division of the 2.56 GHz clock generated in the LC-PLL seen in figure 2.13. The serializer clock division output phase is unknown and changes at each power up of the component. It causes a random delay between LOC and SCK clock edges at each power up of the board. To accommodate this issue a delay must be added at each power up to one of the LOC or SCK clocks. The added delay shifts one clock to have the edges of the two clocks separated by enough time.

Two calibration procedures have been created to find the correct values of this delay. The delay can be added to the FCK clock with the delay register at the LOCx2 input or by adding a delay in the GBTx to the 40 MHz clock going to the LOCx2. The two calibrations are detailed in 2.6.2.

Each LOCic frame built in the LOCx2 is sent to one channel of the MTx. The MTx is a dual channel optical transceiver. The fibers of different MTx are merged into MTP-12 optical fibers links and sent to the LDPB.

2.2.1.3. LTDB flavors

The LTDBs are created in 7 different flavors for the different parts of the detector, each one having a different number of channels used from the 320 available on the common PCB. The seven flavors are:

- EMB : Electro-Magnetic Barrel: covers the full barrel. 64 LTDBs of this type are needed.
- EMEC : Electro-Magnetic End Cap: covers the $1.6 < |\eta| < 2.4$ region of the electromagnetic end cap. 32 LTDBs of this type are needed.
- EMECSpec0 : Electro-Magnetic End Cap Special 0: covers the $1.4 < -|\eta| < 1.6$ region of the electromagnetic end cap. 8 LTDBs of this type are needed.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

- EMECSpec1 : Electro-Magnetic End Cap Special 1: covers the $2.4 < |\eta| < 3.2$ region of the electromagnetic end cap. 8 LTDBs of this type are needed.
- HEC : Hadronic end-cap : covers the hadronic end-cap. 8 LTDBs of this type are needed.
- FCAL0 : Forward CALorimeter 0: covers the first layer of the forward calorimeter. 2 LTDBs of this type are needed.
- FCAL1 : Forward CALorimeter 1: covers the second and third layer of the forward calorimeter. 2 LTDBs of this type are needed.

A summary of the LTDB flavors is available in table 2.2. All LTDB flavors are similar for the slow control path. However the special end-cap and HEC LTDBs have some missing components on the PCB since they are not needed due to the limited number of channels served by these LTDBs.

LTDB type	Detector	Channels per LTDB	Number of LTDBs
EMB	EMB	290	64
EMEC	EMEC	312	32
EMECSpec0	EMEC	240	8
EMECSpec1	EMEC	160	8
HEC	HEC	192	8
FCAL1	FCAL Layer 1	192	2
FCAL2	FCAL Layer 2 and 3	192	2

Table 2.2.: List of the LTDB flavors with the corresponding number of channels for each flavor.

2.2.2. Front-End FELIX

The Front-End Link eXchange [44] (FELIX) board is a PCIe board which functions as a router connecting on one side the TDAQ system, the DCS system, the TTC System and on the other side the LTDB.

The board, seen in figure 2.16, is based on a Xilinx Kintex UltraScale FPGA. It contains 8 Minipods transceiver, four of them being twelve fiber Tx, and four of them twelve fiber Rx for a total of 48 bidirectional links. The FELIX is connected to the LTDBs through two MTP 48 connectors. Since the LTDB is using 5 bi-directional links only 45 of the available links on the FELIX are used to communicate with up-to 9 LTDBs. These links are using the GBT protocol which is encoded by the FELIX in the firmware of the FPGA.

2.2.2.1. The GBT-Protocol

The GigaBit Transceiver protocol (GBT) is developed at CERN to increase the bandwidth between the experiment caverns and the counting rooms for the high luminosity

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

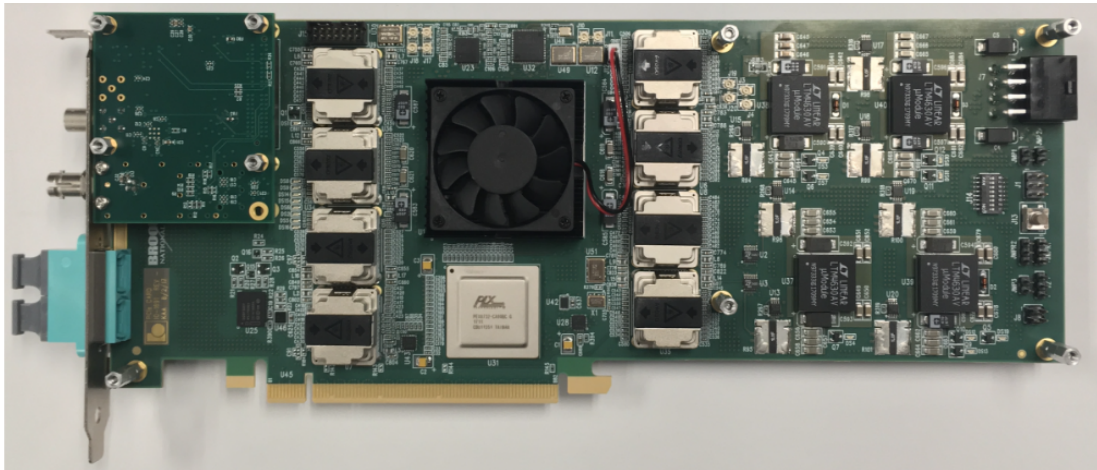


Figure 2.16.: FELIX-712 PCI-E board[44], the FPGA is located under the fan. 4 minipod can be seen on right of the FPGA (metal box with black plastic part) and four on the left of it

upgrade of the LHC. It is supported by point to point duplex link through optical fibers. It can transmit simultaneously physics, trigger and slow-control data.

The GBT frame (figure 2.17) length is 120 bits transmitted at 40 MHz, for a total throughput of 4.8 Gb/s. The frame contains a 4 bits header, 32 bits of forward error correction and 84 bits of User field. The forward error correction is important to handle the high dose of radiation present in the experiment. 4 bits of the user field are reserved for the slow-control, 16 for the Timing and trigger (TTC) and 64 for the data. To be able to fulfill the needs of the different experiments the user field is fully configurable and can follow a different pattern. On the slow control part the 2 first bits are called internal control (IC) and the others two bits are called External Control (EC). The IC is used for the slow control of the component decoding the GBT frame and the EC for slow control of components external to it. The 80 others bits are designed as five groups of 16 bits called E-groups, each one contains 8 E-paths of 2 bits. The E-paths can be merged and used as 2 bits, 4 bits, 8 bits, or 16 bits field. Each field is designated by a unique number called E-link. The E-link contains also the number of the Felix board on the server and the fiber number.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

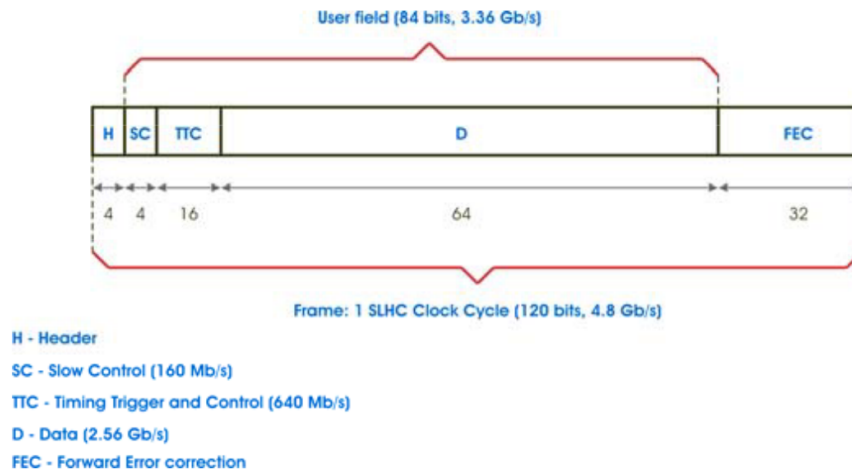


Figure 2.17.: The GBT frame [29]. The frame is composed of 5 fields: the header, the slow control field, the TTC field, the data field and the forward error correction field.

2.2.2.2. The GBT Protocol for the LAr GBTx component

field	hdr	IC	EC	Bank 4			Bank 3		Bank 2		Bank 1		Bank 0	FEC
width	4	2	2	2	12	2	14	2	14	2	14	2	16	31
use		IC	SCA	SCA-AUX		BCR		BCR		BCR		BCR		
bits	119-116	115-114	113-112	111-110	109-98	97-96	95-82	81-80	79-66	65-64	63-50	49-48	47-32	31-0

KC_LTDBbits_v01

Figure 2.18.: The GBT frame which is specific to LAr. Four 2-bit E-links are used for BCR, one for each LOCx2. Two 2-bits E-links are used to control the SCAs. One 2-bits E-links is used to directly configure the GBTx, this is used for debugging and not in the production software.

For the LAr LTDB needs, a special FELIX firmware has been developed to provide a modified GBT frame (figure 2.18). In this modified frame the TTC and Data fields are replaced by five new fields of two bits width and seventy unused bits. The EC control field is used for the direct SCA as the standard GBT protocol. In addition one of the new fields is reserved to control the auxiliary SCA. The other four new fields are used to transmit the TTC BCR data. Each of these fields transmit the BCR to one of the LOCx2 that is connected to the GBTx component. The BCR information is only used for the down-links. However the two slow control fields are also used for the up-link to send data from the SCA to the FELIX.

More than two bits are needed to send an instruction to the SCA. However only 2 bits are available in one GBT frame. The solution is to divide the needed bits by packets of two and send them in different GBT frames. Afterward the GBTx sends these packets bit by bit to the SCA. The aggregation of these bits form the High-level Data Link Control(HDLC) protocol frames [20] describes in the next section(2.2.2.3).

2.2.2.3. HDLC-Protocol Layer and SCA Channel command Layer

The HDLC data frame is shown in figure 2.19. It is composed of an 8 bit start of frame (SOF), an 8 bit address field, an 48 bit data field containing the SCA frame, a 16-bit checksum (FCS) and an end of frame delimiter (EOF).

On the control field of the HDLC frame three commands can be sent :

- TEST : Put the SCA in loopback for debugging, this is not used for the LTDB control system.
- CONNECT: Indicates to the SCA if it will be controlled by its direct or an auxiliary port. The SCA can be controlled by only one port and discards the data received on the other port. This command is used in the LTDB control system when the OPC-UA server loses the communication with the SCA and tries to recover it.
- RESET: Fully reset the SCA. This command resets the SCA and all its registers. The command also acts as the connect command. This command is used in the LTDB control system at the connection of the OPC-UA server and at expert request.

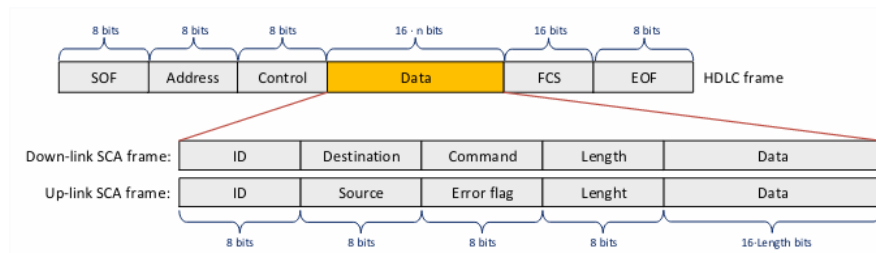


Figure 2.19.: The HDLC frame [12]

The HDLC data fields contain the SCA frame. This frame is different for the up-link and for the down-link. For the down-link, the SCA frame is composed by of an 8-bit identification (ID), an 8-bit destination channel identifier, an 8-bit command, 8 bits indicating the data length in number of bytes, and finally the data. On the up-link the destination field is replaced by the source field and the command field replaced by the error flag field. The channels (source and destination) can be for example the first I2C bus, the ADC, or the GPIO of the SCA. The full list of channels is available in annex A. For each down link frame received, the SCA sends an up-link frame. This frame contains the same ID as the down-link, and the source ID is the same as the destination. The error flag sent is the error code of the received command. An example of an SCA command frame and the corresponding SCA answer frame are described in table 2.3.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

	ID	Destination	Command	Length	Data
Request	12	I2C1	I2C_S_7B_W	4	I2C slave address + Data to write + 2 empty word
	ID	Source	Transaction Error Flag	Length	Data
Answer	12	I2C1	0x0 (no error)	4	I2C error code

Table 2.3.: SCA frame for a single write on 7bit address slave I2C transaction (I2C_S_7B_W) of the first I2C master interface of the SCA. The transaction with the SCA was successful and the decoding of the answered data tells the error code of the I2C itself.

2.2.3. LAr Digital Processing Blade

The LAr Digital Processing Blade (LDPB) is composed of four Advanced Mezzanine cards (AMC) called LATOME plugged into the LAr carrier (LArC) ATCA board. The LATOME is based on an Intel Arria 10 FPGA. It has 4 MTP-12 connectors receiving the fibers coming from the LTDB. The goal of the board is to compute the deposited energies in the detector from the values of the LTDB ADCs. A functional diagram of the LATOME firmware is shown in figure 2.20. The LTDB data are received by the input stage block. It decodes the LOCic frame and synchronises the data between fibers. The data from the input stage are sent to the configurable remapping block. The remapping block reorganizes the channels by trigger tower and region of interest for the trigger system and sends them to the user code. The user code computes the energies deposited in the detector and transmit them to the TDAQ monitoring and to the output summing block. The output summing block sums per region of interest the energies to be transmitted to the different features extractors (FEX) of the trigger system. The TDAQ/Monitoring block buffers the data until reception of the trigger decision and transmits them to the TDAQ system and to the monitoring system. The TTC block decodes the data from the TTC system and propagates them to the blocks that need them. The LATOME board needs to be configured and monitored by writing and reading into different registers distributed inside the firmware blocks. This configuration is done through the IPBus Ethernet protocol decoded by the IPBus controller block. The clients connected to this IPBus controller block are the TDAQ system and the OPC-UA server of the LTDB.

The LTDB OPC-UA server reads status registers inside the input stage block that monitors the data transmission between LTDB and LDPB. These status register are composed of different error flags for the LOCic frame decoding. An error counter also exists for each of the status. Three different errors are possible and one register aggregates all errors:

- CRC Error: The CRC contained on the frame does not correspond to the one computed from the received data.
- PRBS error: The PRBS decoding indicates an error. It is not possible to extract the BCID from what is received.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.2. Hardware Components

- Bitshift error: The LOCic frame header is not found.
- Channel ready error: at last one of the above error is found.

These errors are visible if the LOCx2 delays are not well selected on the LTDB as detailed in 2.2.1.2. The calibration of the LOCx2 delay is based on the reading of the channel ready error register to choose the correct delay.

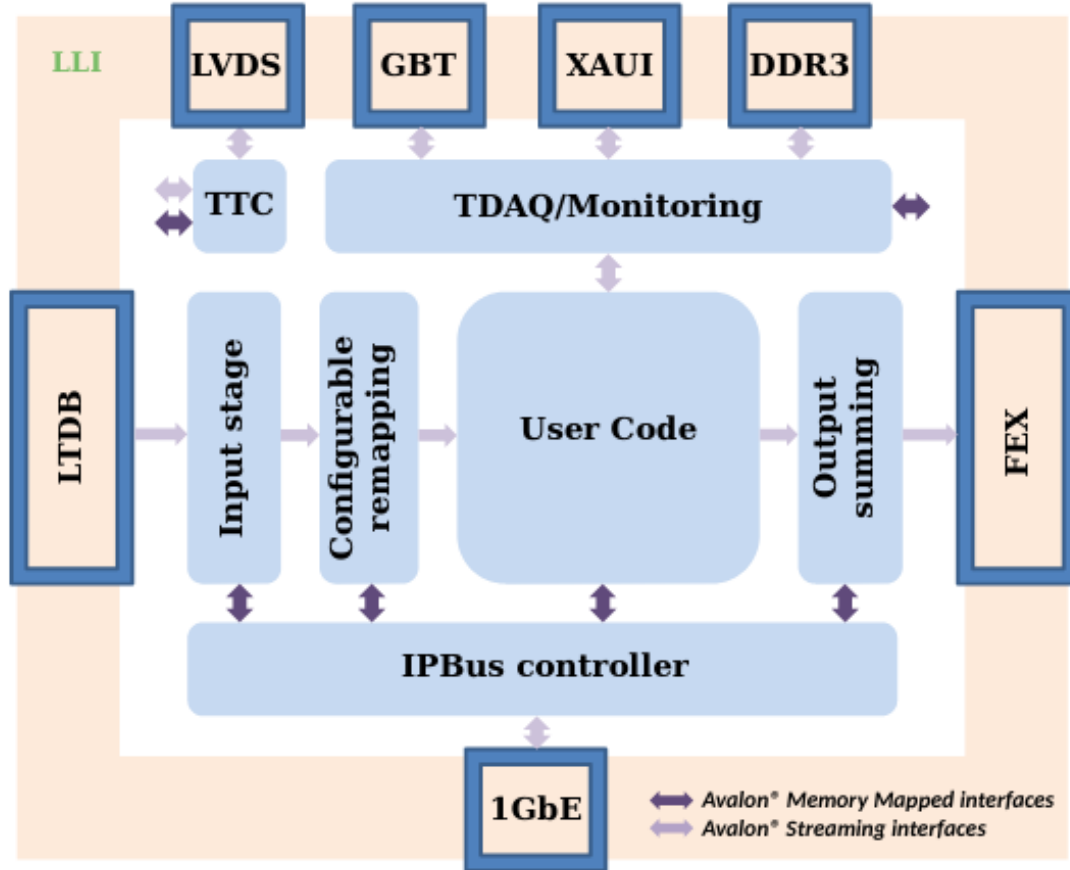


Figure 2.20.: Block schematic of the LATOME firmware.

The LATOME boards are organised in seven types with different mapping configuration covering the different parts of the Detector. The seven types are summarized in table 2.4 and described in what follows:

- EMB : Cover the full ElectroMagnetic barrel, 4 LATOMEs per quadrant.
- EMB/EMEC : Cover the region shared between the barrel and the endcap LTDBs, 4 LATOMEs per quadrant.
- EMEC: Covers the central region of the electromagnetic end-cap.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

LATOME type	Detector	Channels per LATOME	Number of LATOME
EMB	EMB	320	32
EMB/EMEC	EMB,EMEC	312	32
EMEC	EMEC	312	32
EMEC/HEC	EMEC	80	16
	HEC	96	
FCAL1	FCAL Layer 1	192	2
FCAL2	FCAL Layer 2 and 3	192	2

Table 2.4.: List of the LATOME Mapping flavors and the corresponding and the corresponding numbers of channels.

- EMEC/HEC: covers the Forward region of the end-cap and the HEC, 2 LATOMES per quadrant.
- FCAL1: Cover the first layer of the forward calorimeter. One LATOME per side.
- FCAL2 : Cover the second and third layer of the forward calorimeter. One LATOME per side.

2.3. Software

Several software tools are developed to monitor, control and configure the LTDBs. The tools are divided into 3 categories: low level tools provided by the ATLAS collaboration that ensure generic communication with the hardware through the FELIX interface, specific LAr software tools that implement the actual needs of the LAr hardware, and user clients that implement the interface with DCS and TDAQ users. Several software packages are needed to control the LTDBs:

- Felixcore: Felixcore is the software making the link between Ethernet communication and the FELIX board through its driver. Felixcore is a tcp-ip client providing methods to use the GBT communication of the boards to access the LTDBs. This layer is developed by the FELIX team.
- The OPC-UA SCA Server : This OPC-UA Server is developed by ATLAS Central DCS team to provide a unified server to handle communication with SCAs. The SCA is mainly used by the New Small Wheel and Liquid Argon Calorimeter for their phase-I upgrades.
- OPC-UA LAr LTDB Server: The OPC-UA LAr LTDB Server is a specific OPC-UA server made to cover the needs of the LAr sub detector for the LTDB. It provides high level methods to monitor, configure and calibrate the LTDB.
- Clients and User interfaces: Two OPC-UA clients provide graphical user interfaces to control and monitor the board: one client for the DCS System and one for the TDAQ System.

2.3.1. Felixcore

Felixcore is one of the software packages provided by the FELIX team to operate the FELIX boards. This TCP server provides an interface to send/receive data to/from the different E-links of the FELIX. In the case of the LTDB it provides an interface for the communication between the OPC-UA SCA server and the HDLC E-links connected to the SCAs. A dedicated library named Netio is developed by the FELIX team and implemented into OPC UA SCA server to communicate with felixcore.

2.3.2. OpcUaScaServer

Several sub-systems in ATLAS are using SCAs as main components for the control and monitoring of their Front-End boards. The main ATLAS users, LAr and NSW, are using respectively 620 and 6976 SCAs. A common quasar-based Server, the OPC-UA SCA server is developed by the ATLAS DCS team to standardize the communication with the SCAs from different sub-detectors [30].

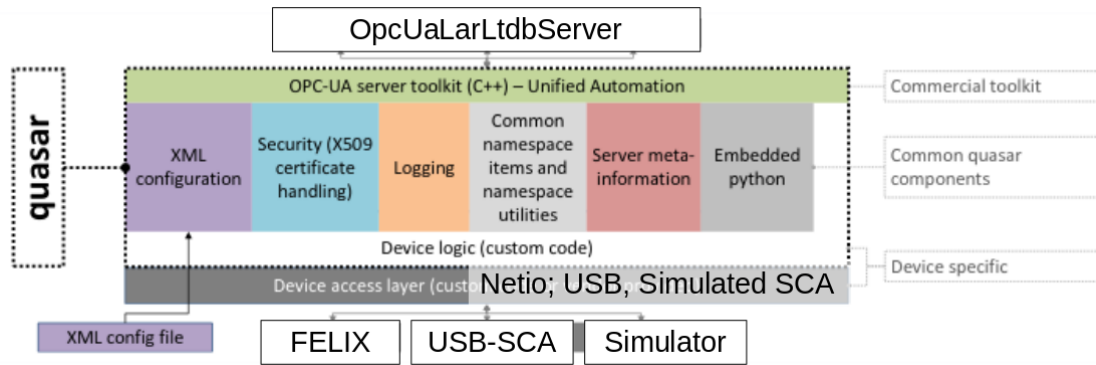


Figure 2.21.: Schematic of the quasar architecture for OpcUaSca server.

The OPC-UA SCA server is generated by quasar. This server is developed to communicate with SCAs with three available connections.

- USB-SCA: SCAs connected through USB protocol. This connection is mainly used for development and test with a reduced amount of SCAs.
- Simulator: The simulator creates virtual SCAs. These virtual SCAs emulate the behaviour of real ones and are mainly used for the development when hardware is not available.
- FELIX: The SCAs are connected through the FELIX board. This connection is used in production.

The OPC-UA SCA server implements a dedicated device access layer for each of these connections.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

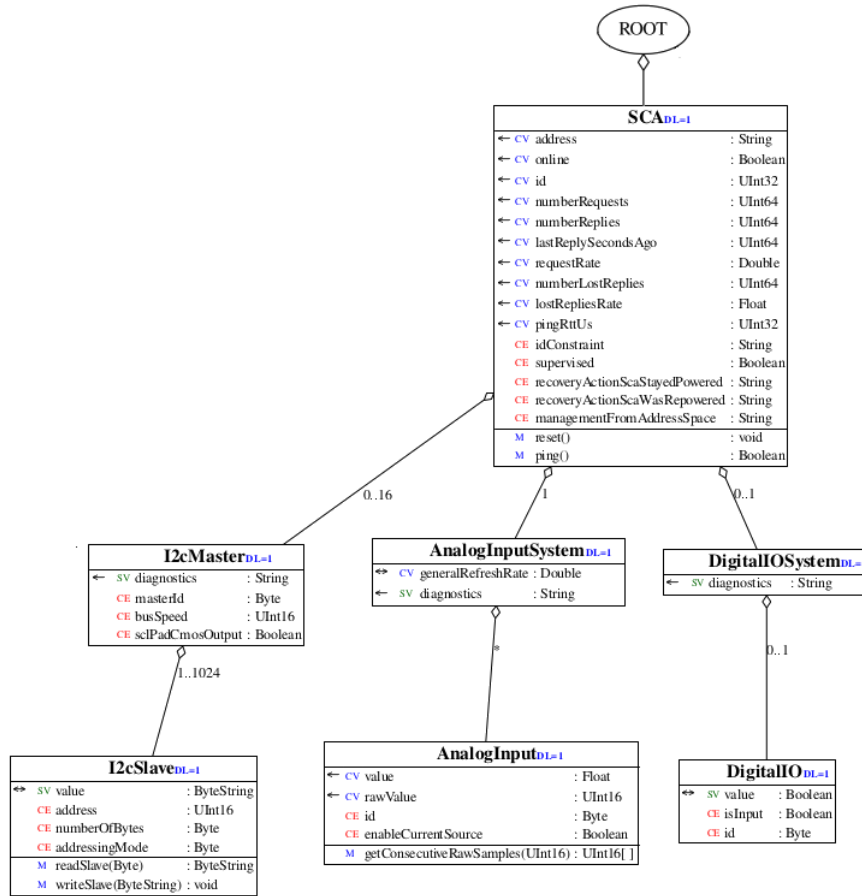


Figure 2.22.: Diagram representing the Device Logic Hierarchy of the OpcUaScaServer. Inside the classes the Config Entry (CE in red), Cache variables (CV in blue), Source Variables (SV in Green) and Methods (M in blue) are represented, their data types are also indicated.

A simplified schematic of the device logic of the OPC-UA SCA server is available in figure 2.22. Only the device logic classes used in LTDB configuration are shown. The device logic is hierarchically structured, the SCA is the top level and contains several child classes:

- **Digital IO System:** Class handling the GPIO system. Each of its children (DigitalIO classes) represents one GPIO of an SCA.
- **I2C Master:** This class represents a I2C bus of the SCA. On this bus SCAs are Master and connected devices are slaves. The slaves are represented by a child class of the I2C Master class.
- **Analog Input System :** Class handling the analog inputs and internal ADC of the SCA. The AnalogInput class is a child of the analog InputSystem and represents one analog input.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

The different classes are instantiated at the startup of the OPCUA SCA Server depending of the extensible markup language(XML) configuration file. This XML file declares how many SCAs must be instantiated. Each SCA is declared with several children and sub children depending of the hardware. For the LAr LTDB each SCA is declared with the same hierarchy, it includes one digital IO system, one analog input system and ten I2C masters.

The I2C slaves are the registers of the components controlled by I2C on the LTDB. The LTDB I2C components are separated in three groups depending on how their internal registers are addressed:

- Direct addressing components: On these components each register has its own I2C address. An I2C slave is declared in the XML config file for each register. The LTDB MTxs, MTRx and LOCx2 components use direct addressing.
- Indirect addressing components: On these components only one address is used. The data sent to this address contains the internal address of the targeted register. For these components only one I2C slave is declared in the configuration file. On the LTDB system only the GBTx uses indirect addressing.
- Mixed addressing: On the LTDB the ADC uses a mix of direct and indirect addressing registers. For each ADC there are three I2C addresses:
 - Control address: The control register of the ADC uses direct addressing. This register is used to configure the I2C link of the ADC.
 - Write address: This address is used to send command or configuration to the ADC, the first byte of the I2C frame containing the internal address of the desired register.
 - Read address: This address is used during read operation. This is a direct addressing register. Its value mirrors the value of an internal register after a read command is sent to this register via the write address of the ADC.

In total eighty-seven I2C slaves registers per SCA are declared in the configuration of the OPC-UA SCA Server.

Only one Digital IO system exists on the SCA. For the LTDB system it contains three types of DigitalIO children.

- Reset output: Twenty digitalIO classes represent SCA outputs controlling a reset signal. These signals control the reset of the GBTx , ADC , LOCx2, MTx and MTRx components.
- Control output: Five digitalIO classes represent SCA outputs controlling a control signal. These control signals are the ones of the PDB regulators powering the LTDB.

- Status input: Two digitalIO classes represent SCA inputs receiving a status signal. These status signals come from the two PDB regulators.

In total 27 Digital IO are declared in the configuration file.

Only one Analog Input System exists on the SCA. For the LTDB system it is used to monitor temperature and tension on the boards. Nineteen analog inputs are declared per SCA to cover all the monitoring channels.

The Devices classes can be composed of children but also by several entries of different types.

- Config entries: The value of the configuration are read from the XML configuration file when classes are instantiated. It is used to specify parameters for the components represented by the instantiated devices classes. For example the address of an I2C slave is defined as a configuration entry.
- Variables: The variables are defined in the address space of the server. The variables contain a value, a status and a timestamp. They are defined as input or output variables and can be of two types:
 - Cache variables: The values of these variable are stored in the memory of the server. Operations with these variable are fast.
 - Source variables: The values of these variable are not stored in the memory of the server. Operations accessing them, in write or read modes, are propagated to the LTDB directly. A source variable is defined as synchronous or asynchronous mode. In synchronous mode the write/read is done by the main thread of the server, it hangs all other address space access until the completion of the operation on the LTDB. In asynchronous mode the LTDB access is done in a dedicated thread.
- Methods: The methods are functions accessing the LTDB to act on it. It can be for example an I2C write on a I2C slave. As the source variables they can be defined as synchronous or asynchronous.

To facilitate the development and the integration of this server, a library named UaoClientForOpcUaSca is provided by central DCS. This library is generated by UaoForQuasar as described in 2.1.2. This client library is used in the OpcUaLarLtdbServer as a device access layer to access the LTDB through the OpcUaScaServer.

2.3.3. OpcUaLarLtdbServer

The LTDB being a complex board accessed by two different systems, DCS and TDAQ, it has been decided to develop a custom server to access it. I have been in charge of the design, the implementation, the test, and the integration on the DCS and TDAQ

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

systems of this server. It is designed to handle the concurrency between clients, to provide a simple interface to clients masking the complex operations on the board and to handle errors. The OPC-UA protocol was chosen to build this server which is called the OPC-UA LAr LTDB server. This choice provides a native way of communication with the OPC-UA SCA server, with the DCS system, and with the TDAQ system. The server is developed in C++ and is based on quasar. Figure 2.23 shows a schematic of the architecture of this server.

The server needs to access to the LTDB and to access the LATOME. This LATOME access is used during calibration of the LTDB. To communicate with these boards, two library are implemented in the Device Access Layer:

- IpBus: The communication with the LATOME is done through Ethernet connection and implement an IpBus protocol. The library called uHAL IpBus implements it.
- OPC-UA library: Two library to implement OPC-UA exists. A commercial library named UA-SDK is provided by Unified Automation and an open source library also exist named open-62541. A module named open62541-compat is provided in quasar to allow the switch between the two library without changing custom code. The OPC-UA LAr LTDB Server was developed initially with open62541 but switched to UA-SDK during a release. Open62541 do not support asynchronous source variables which are needed for this server.

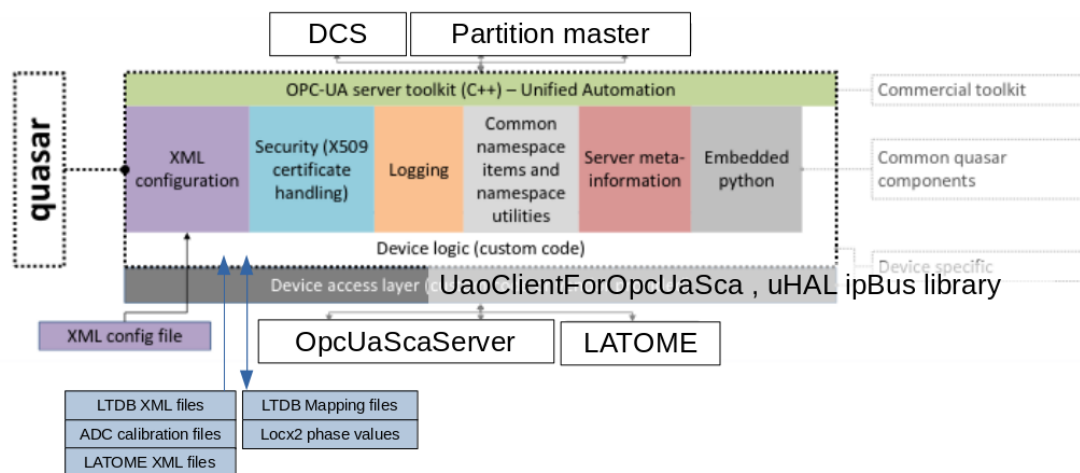


Figure 2.23.: Schematic of quasar architecture for the OPC-UA LAr LTDB server.

The configuration of the OPC-UA LAr LTDB server is made by several configuration files:

- Standard XML quasar configuration files: At its startup, the server instantiates the device classes depending on the XML configuration file.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

- XML configurations files for LTDB components: Two configuration files provide the value of the registers written in the LTDB components during configuration. A first file contains standard values used by default for all LTDBs. A second file allows to modify these values individually for a specific register of the LTDBs.
- XML LATOME description files: These files describe the LATOME and its IP-Bus registers.
- Mapping files: These files describe how the optical fibers are connected between the LTDBs and the LATOMEs.
- Calibrations files: These files contain configuration entries changing over time for the LTDB. They are generated during the calibration of the board.

The Standard XML quasar configuration file describes hierarchically the LTDB and is used at server startup to instantiate the different Device classes. The configuration entries defined in this file are the hardware fixed values, for example I2C addresses and components IDs. They are generated from the topology database and from the LTDB production database. The topology database describes how the FELIX boards and the LTDB boards are connected. The LTDB production database contains the values of all SCA IDs. These IDs are used to check that the right LTDB is addressed during configuration.

LTDB XML files are separated into one general file containing information which is common to all boards and one specific XML file per board. The general file contains all the default configuration values used for the LTDB configuration. The file per LTDB contains variations for each board. These configuration entries were separated from the standard XML quasar configuration file to allow a change and reload of the configuration without a restart of the server. The configuration change without server restart was needed in particular during the commissioning of the boards.

Some components of the LTDB are not perfect which leads to intermittent communication issues or configuration failures. For example one LOCx2, of the 2480 equipping the full system, has an issue in one register. One of its eight bits is stuck at 0. If the OPC-UA LAr LTDB server tries to configure this bit with a value at 1, the configuration read-back will read a 0 and the configuration fails. Because of cost, time and access issues, it is not possible to replace this component. A special value for the configuration of this register mitigates the issue. This special value is provided to the server in the LTDB XML configuration file of this LTDB.

The LATOME XML files contain what is needed by the IpBus library to access the different LATOME boards. The needed entries are LATOME IP addresses and LATOME registers hierarchy and internal addresses.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

One mapping file per LTDB defines the correspondence between the LTDB output fibers and the LATOME input fibers. This mapping data is used during the calibration of the board. The server is also able to generate this mapping. The mapping generation was used during installation to validate the cabling.

Several calibrations files are needed for the configuration of the LTDB. In contrary to configuration entries, the calibration values vary between identical components and are not fixed over time. For the LTDB two components need calibration and have dedicated files for calibration results:

- LOCx2 phase calibration: As describe in [2.2.1.2](#), the LOCx2 component needs a calibration to correctly encode the LOCic frame. A file per LTDB contains the values of the phase shift of the GBTx clock going to the LOCx2s . This file is generated during the calibration which is also done by the server.
- ADC calibration: As describe in [2.2.1.2](#), the NEVIS ADC contains a digital correction of the output. This correction is linked to the components production and temperatures condition during operation. The values of the digital correction is obtained through a calibration. The calibration files contain these correction values and are provided by the board manufacturer. The TDAQ System is also able to calibrate the boards in-situ through a specific interface with the server. Studies to compare the in-situ and manufacturer calibrations output are on-going with the full system.

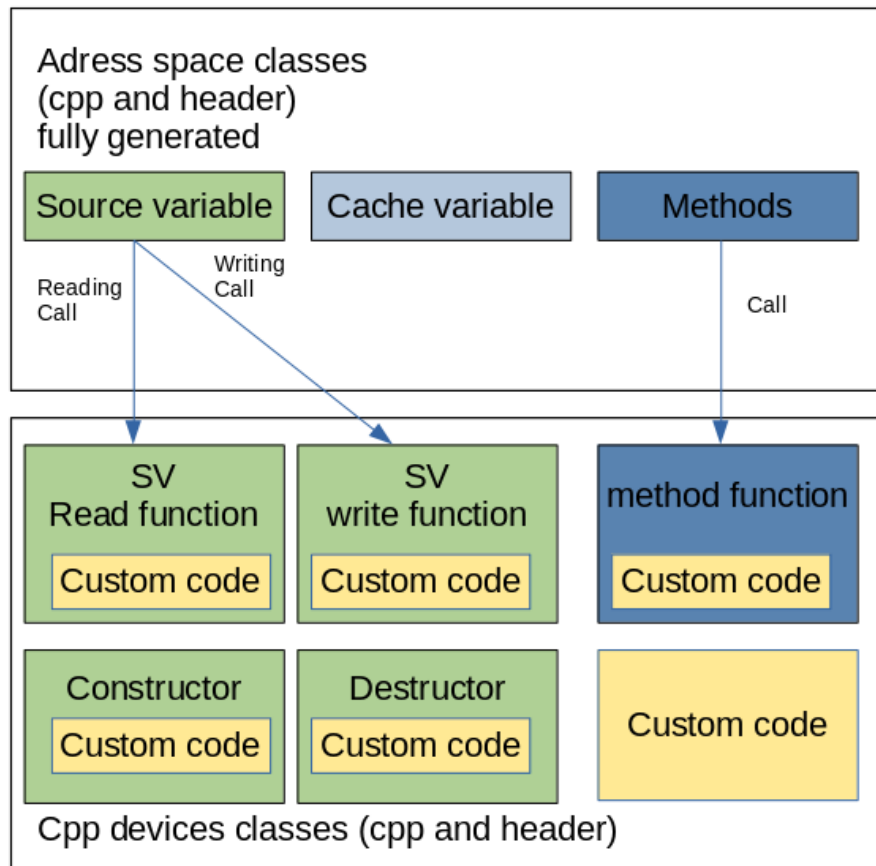


Figure 2.24.: Schematic of the quasar generated C++ classes for the Device Logic classes with the placeholders for custom code. The address space class is fully generated and the device C++ classes contain placeholder for custom code.

Part of the C++ code of the server is generated by quasar from a Design XML file. In this design file the Device logic classes are defined with their hierarchy and all their address space variables and methods. Quasar generates two C++ classes (represented in figure 2.24) for each of the Device classes defined in the design XML file:

- The Address space C++ class: This class handles the address space of the Device logic class. It provides methods to get/set the address space cache variables from the rest of the C++ code. This C++ class contains only generated code.
- The device C++ class: This class is partially generated by quasar. The generated parts are:
 - Member variables representing the configuration entries: These variables are initialised with the value of the Quasar XML configuration file when the class is instantiated.
 - Address space access: A pointer to the corresponding address space class.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

- Constructor and destructor of the class: These methods contain placeholder for custom code.
- Methods for source variable calls and methods calls: The code to handle the OPC-UA methods and source variables access in the address space is partially generated. The only modified code by the developer are the callback C++ methods. A callback function is generated per method. Two callback methods are generated for the source variables, one called when the variable is read, and the other is called when the variable is written. This callback methods contain the code accessing the LTDB hardware.
- Placeholder for the fully custom class members and methods.

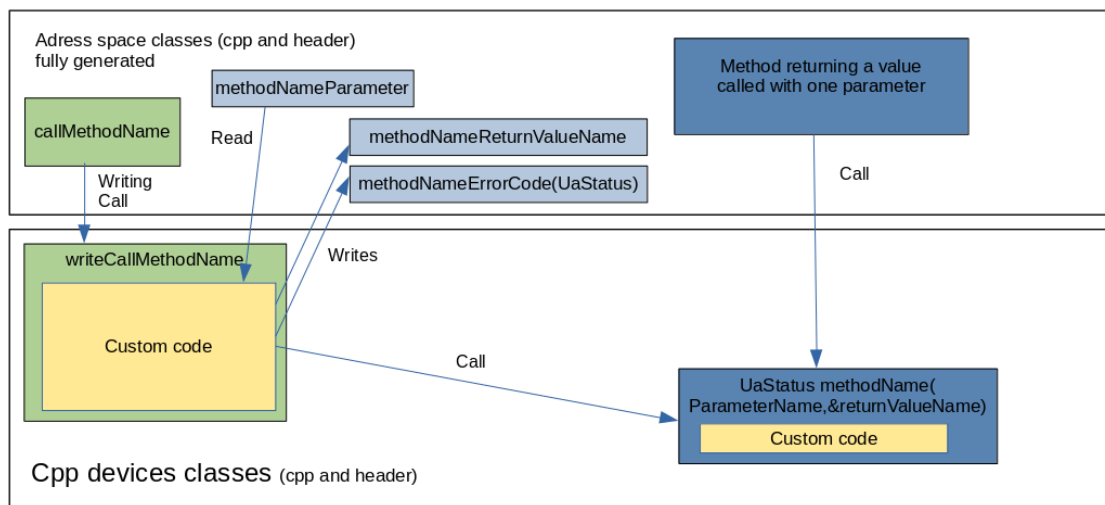


Figure 2.25.: Schematic of the implementation of the call of a method using address space variables.

The DCS client software (Wincc OA) does not fully implement the OPC-UA protocol in its version 3.16. This version will be used for run 3. The main feature not implemented yet is the usage of methods. Wincc OA clients are only able to access cache and source variables in the address space of the OPC-UA LAr LTDB server. A work-around solution has been developed to mitigate this issue as described in figure 2.25. For each of the methods to be used by the Wincc OA client several address space variables are defined:

- **Parameters cache variables :** A cache variable is created for each of the parameters of the method.
- **Error code cache variable :** A cache variable is created to contain the OPC-UA error code of the method.
- **Return value cache variables:** A cache variable is created for each of the returned values of the method.

- Method call source variable : A write in this source variable calls the method.

To call the method, the Wincc Oa client writes its parameters in the parameters cache variables. After, the client writes a value in the method call source variable. The call-back function associated to the source variable reads the parameters, calls the method and writes its return values and error code in the corresponding address spaces cache variables. The DCS client finally reads these address space variables.

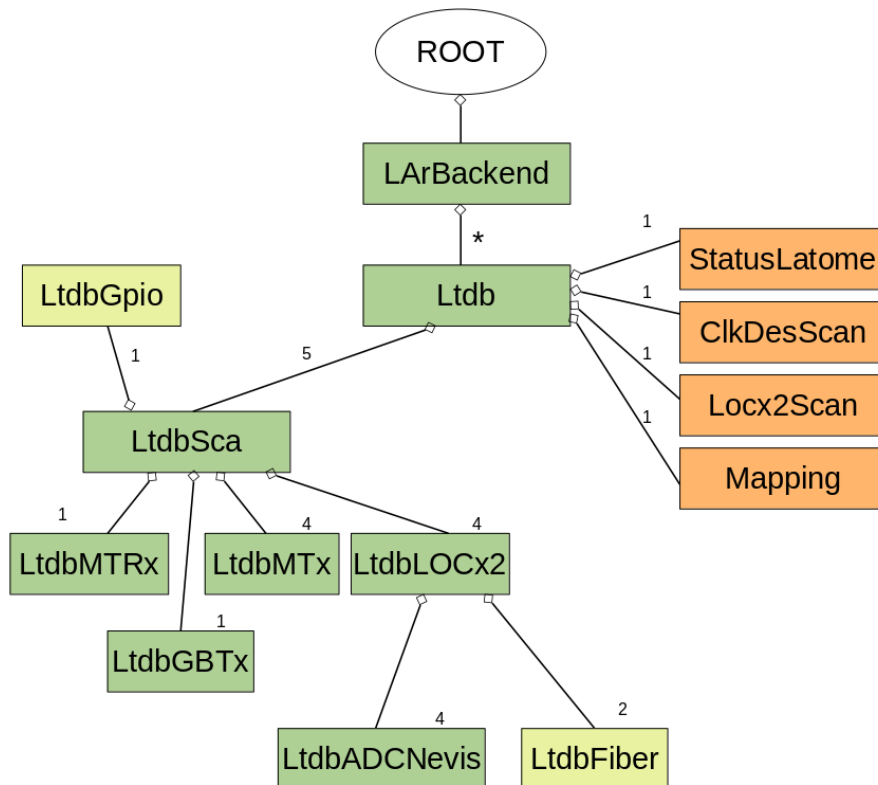


Figure 2.26.: Diagram representing the Device Logic Hierarchy of the OpcUaLarLtdb-Server and its Address-Space. Dark-green classes represent the hardware (board or components), pale green classes represent a part of the parent component and orange classes represent a complex functionality which call several components.

Figure 2.26 represents the device classes hierarchy. Fourteen classes represent the LTDB, some of them represent hardware components and some of them complex functionalities which need to access several hardware components.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

class name	hardware component	Instances	methods in address space	monitoring values in address space	Short class description
LArBackend	FELIX Server	1	YES	YES	Configure the FELIX boards
Ltdb	LTDB board	1/LTDB	YES	YES	Control and monitor the LTDB.
LtdbSca	SCA component	5/LTDB	YES	YES	Control and monitor the SCA.
LtdbGpio	SCA's GPIOs	1/SCA	NO	NO	Control the GPIO of the SCA.
LtdbMTx	MTx component	4/SCA	NO	NO	Control the MTx.
LtdbMTRx	MTRx component	1/SCA	NO	NO	Control the MTRx of the SCA.
LtdbGBTx	GBTx component	1/SCA	YES	YES	Control and monitor the GBTx.
LtdbLOCx2	LOCx2 component	4/SCA	YES(deprecated)	NO	Control the LOCx2.
LtdbFiber	LOCx2 component	2/LOCx2	YES(deprecated)	NO	Set the LOCx2 clock delay
LtdbADC Nevis	ADC component	4/LOCx2	YES(deprecated)	NO	Control the ADC.
Status LATOME	LATOMEs board	1/ LTDB	YES	YES	Communicate with LATOMEs.
ClkDesScan	None	1/LTDB	YES	YES	Calibrate the GBTx clock phase.
LOCx2Scan	None	1/LTDB	YES	YES	Calibrate the LOCx2 clock phase.
Mapping	None	1/LTDB	YES	YES	Generate the LTDB/LATOME mapping.

Table 2.5.: The OPC-UA Device classes implemented in the OPC-UA LAr LTDB server.

The LArBackend class represents the Felix Server installed in the the back-end USA15 cavern of the experiment. The OpcUaLarLtdb server runs on this host. This class is made to control and monitor the clock of the FELIX boards. This class will evolve in future releases of the FELIX software to include more monitoring functionalities of the FELIX boards.

The goal of the LTDB class is to provide: fast address space methods acting on the full board, monitoring of the board, a state indicating at which configuration step the board is. The methods of this class call the functions included in the different children classes. The implementation is multi-threaded to allow parallel access to the five SCAs on each LTDB. Each thread implements an error detection and correction procedure for the corresponding SCA. The monitoring of the full LTDB is implemented in a specific thread. This thread updates the monitoring values included in its children classes address space. Two monitoring groups are implemented and handled by the same monitoring thread. One group is updated every second, the other is updated every minute. The state is handled by the LTDB class and available in its address space. This state is updated following the top level methods calls and if the monitoring thread detects an error that leads to a state change.

The LtdbSca class represents the SCA component. This class is made to access the monitoring values handled by the SCA component, to configure the LTDB components through the SCA output while handling errors, and to check the SCA identifier. This SCA identifier is unique and allows to validate that the expected SCA is configured. The monitoring is done directly by the custom code of this class. It provides several functions to read the monitoring values and save them in the address space.

The LtdbMTRx, LtdbLOCx2 and LtdbMTx classes do not have any address space and contain only custom functions to configure the components. These functions access the different I2C registers of the related component. Some of the registers are simple and contain only one field but some are complex with several fields in

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

the same register. Some configuration values are also split into different registers at different addresses. The component class provides read and write functions for each field abstracting the register complexity. The components do not have any address space and are made to be used by top level classes and functions.

The `LtdbGpio` class represents the gpio part of the SCA and provides functions to write/read each GPIO. The class provides methods to power on/off the board and methods to reset the components. This component class does not have any address space and is made to be used by top level classes and functions.

The `LtdbAdcNevis` class contains all the custom functions to configure, calibrate and operates the Nevis ADCs. Its address space contains a cache variable for each of the MDAC parameters written in the ADC during configuration.

The `LtdbFiber` class represents a fiber and its LOCic frame. Its code contains functions to change the delay of the LOCx2 input clock phases. These delays are written in several registers of the LOCx2 components. This class provides the same methods independently of the fiber position and corresponding registers of the LOCx2. Its address space gives access to the mapping of the fibers.

The `LtdbGBTx` class contains all the functions to access the GBTx registers in particular the phase delay and monitoring values. These monitoring values are written in the address space of this class by the LTDB monitoring thread. The GBTx access is done on the board by one I2C bus with several master. The SCA does not implement multi I2C master protocol so multiple GBTx access can collide. To protect from collision only one SCA is used as master. The different GBTxs are controlled by different threads during configuration and collisions can occur between them. To protect the I2C access between them a mutex is used. The mutex is a computing lock that allows only one user (SCA) to access the part of the code using the I2C bus.

The `StatusLatome` class handles access to the LATOMEs. It provides an abstraction over the mapping between the LTDBs and the LATOMEs and over the ipBus-protocol for other classes that need information from LATOME registers. For example the `ClkdesScan` class needs the status of the LOCic encoding for all fibers. The `StatusLatome` provides it by reading the different LATOME registers and selecting the data of the fibers connected to the corresponding LTDB and returning a unique vector that is ordered following LTDB fiber organisation.

The `ClkdesScan` and `Locx2Scan` classes have similar purposes and handle the calibrations of the phases of the LOCx2. The `Locx2Scan` class handles the LOCx2 internal delay for each LOCic encoder. The `ClkdesScan` class handles the delay of the GBTx CLKDES going to the LOCx2. The address space of these classes contains all the methods to execute such calibration and store the results.

The Mapping class is used to generate the mapping between the LTDB and the LATOME. This mapping is used by the StatusLatome class. The mapping generation is needed to validate the cabling of the boards; once it is validated it is fixed during normal operation.

The errors and warning that are generated during code execution are managed by LogIt. LogIt is the standard logging library of quasar servers. The log files are saved on the localdisk of the Felix server. Several levels are defined: trace, info, warning, and error. By default the server is started with the info level.

The transitions are methods changing the state of the LTDB. They are used during the standard configuration of the detector. The DCS and TDAQ clients need to receive more details than the standard OPC-UA error code in case of failure. For that, a log cache variable is defined in the address space of the LTDB. The log cache variable contains all the errors and warning reported during the last transition.

To facilitate the development of the C++ based clients , in particular the TDAQ client, a library is generated by quasar to access the address space of the OPC-UA LAR LTDB server. This library is named UaoClientForOpcUaLarLtdbServer. This library defines a client class per device class of the server. This client class contains methods to read/write address space variables of the server and call its methods.

2.3.4. The DCS ATLLARLTDB Wincc OA project

The DCS System is build using the Wincc OA software from Siemens. For the LTDB control sytem, a Wincc OA project named ATLLARLTDB was created. I was in charge of the design, implementation, test, and integration of the ATLLARLTDB project. This software represents all elements in monitored and controlled systems as data points. A data-point (DP) is of a defined data point type (dpType) and contain several data-point elements(DPE). Each DPE is composed of values, timing, configuration and status data.

For the LTDB project eight data point types are defined each representing a device class of the OPC-UA LAR LTDB: LarBackend, Ltdb, LtdbSca, LtdbGbtX, LtdbMapping, ClkDesScan, Locx2scan, StatusLatome. A data points type is defined for each of the corresponding classes. The names of the data point types are the names of the corresponding classes preceded by the prefix Peripheral. A data point is created for each of the instantiated classes in the OPC-UA SCA server. For each data point, there is one DPE for each address space variable of the class. Table 2.6 lists all the data points types with the number of data points and data point elements for the LTDB system.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

Data point types	Number of Data points	data point elements per data point	Total number of data point elements
PeripheralLarBackend	1/FELIX SERVER: 8	4	32
PeripheralLtdb	1/LTDB : 124	24	2976
PeripheralLtdbSca	5/LTDB : 620	43	26660
PeripheralLtdbGbtX	5/ LTDB : 620	8	4816
PeripheralLtdbMapping	1/LTDB : 124	18	2232
PeripheralClkDesScan	1/LTDB : 124	19	2356
PeripheralLocx2scan	1/LTDB : 124 WIN	28	3472
PeripheralStatusLatome	1/LTDB : 124	8	992
TOTAL	1868		43536

Table 2.6.: List of all dpTypes of the ATLLARLTDB project and corresponding number of DPs and DPEs.

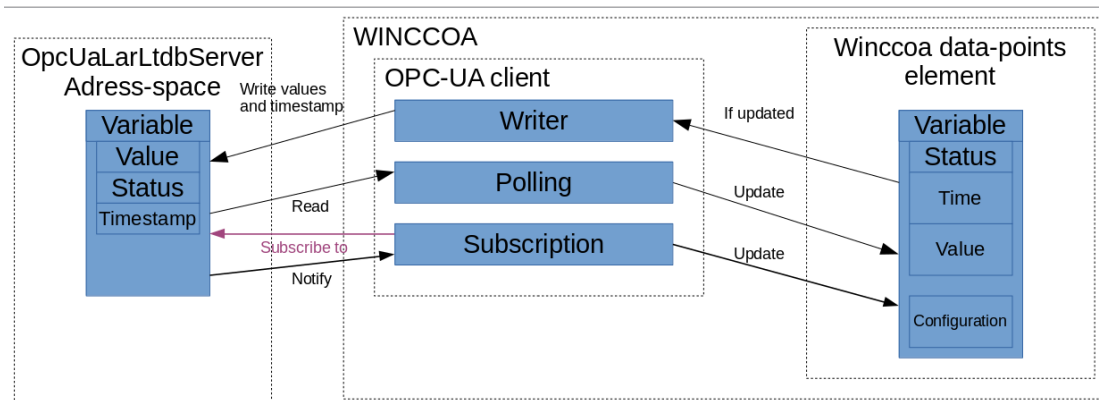


Figure 2.27.: Schematic of the connection between the Wincc OA data-points and the OPC-UA LAR LTDB server address-space.

The connection between a data-point element and the corresponding OPC-UA LAR LTDB server address space variable is shown in figure 2.27. The connection to the server is managed by an OPC-UA client declared in the Wincc OA project. On the LTDB system, eight clients are defined to connect to each of the OpcUaLarLtdbServers. Three managers can connect to the server address space variables and the DPE:

- The writer manager : this manager transmits the values from the DPE to the OPC-UA server.
- The polling manager: this manager copies periodically the values from the OPC-UA server to the DPE.
- The subscription manager: this manager subscribes to a variable of the server. When the value is modified the server notifies the manager and the manager copies the updated value to the DPE.

The writer is used in the ATLLARLTDB system for transmitting methods parameters and calls. The polling manager is not used and the subscription manager is used for all the output cache variables of the server. Managers do not only transmit values but also a timestamp and a status. The timestamp indicates the time of generation of the value and the status indicates its validity.

On the Wincc OA LAr LTDB project several components access the data-points:

- **Scripts:** In the scripts several watchdogs can be defined for data point elements. These watchdogs are called upon modification of the data point elements. They compute values of data points internal to the project depending of the value of several data points.
- **Alarms:** Alarms detect data point values reaching threshold and report them with a severity level.
- **Archives:** The archiving manager saves the values of the data point elements in a database.
- **Graphical User Interface:** This interface allows the user to see the monitored values and act on the system by modifying data point values.

The Graphical User Interface are composed of widgets. The root widget is the panel and it contains several other widgets. These children widgets can be for example buttons, text displays, text inputs. Each widget has a graphical view and configuration but is also composed of several scripts. These scripts are developed in a script language named CONTROL. Per widget, two types of scripts exist: the widget library named ScopeLib and the events scripts. The library of a widget contains functions accessible from the event scripts. Furthermore all child widgets of a panel have access to the panel's ScopeLib. The event scripts are call-back scripts triggered on specific events on the graphical interface. The list of events triggering a script for the panel widget is available in table 2.7.

Script	Description of the event that trigger the script
Initialize	Panel started
Clicked	Left-Click
Close	Panels closed
DoubleClicked	Left-double-click.
DragDrop	Data are dropped into the pannels
DragEnter	Data dropped are checked and accepted
GestureTriggered	One or two finger swipe on a control pad
LangChanged	Language Changed (The language is a global parameter of Wincc OA project)
RightMousePressed	Right-Click
Terminate	When parent panels is closed (if included in an other panel)
Wheel	Mouse wheel event
Zoom	Zoom in or out of the panel

Table 2.7.: List of all events that can start a scripts on panels.

The CONTROL language follows the same syntax as the C++ programming language. Furthermore it provides functions to interact with the data-points and the panels. There is a synchronisation issue occurring when a write operation to the OPC-UA server is followed by a read operation from the OPC-UA server in the CONTROL script.

A write operation followed by a read operation occurs during calls to methods where a write to a source variable calls the method followed by reading the error code in the corresponding cache variable. Two different managers running independently handle these transfers with the OPC-UA servers. The transfer from the data-point to the server of the call value can still be ongoing when the CONTROL script reads the value of the error code data point. In such case the error code is outdated and corresponds to the one of the precedent call of the method. To solve this issue a procedure has been developed using the data point timestamp to respect the write and read order:

1. Write to the data point representing the call source variable and save the corresponding timestamp.
2. Read the timestamp of data point containing the error code in a loop until the value is bigger than the timestamp of the write function.
3. Read the error code data-point.

To facilitate the development and integration of the Wincc OA projects, CERN developed a framework named Joint Controls Project (JCOP)[5]. This framework contains several CONTROL libraries and widgets grouped in components. Each component is independent and installed on demand in Wincc OA projects. This framework was extended by the ATLAS team for their needs and contains specific ATLAS components.

On the ATLLARLTDB project several components are needed:

- fwCore: Base components of the JCOP framework.
- fwAccessControl : Allows users to log with standard CERN accounts to access the protected parts of the project depending on their access rights .
- fwAtlas, fwAtlasInstallation : Base components of the ATLAS project.
- fwDeviceEditorNavigator, fwFSM, fwFsmAtlas: Components used for the creation,management and integration of the FSM of the project.
- fwTrending: Handles trend plots of the data point values.

On the ATLLARLTDB project, hundreds of data points are created containing thousands of data-point elements configured to be connected to the OPC-UA LAR LTDB Server. To create and configure these data points, I designed and implemented an automatic tool. The ATLAS DCS team provides a generic framework to create the dpTypes, the dpElement and to configure them. However a specific tool was created to allow more flexibility for LAR during the commissioning period. Figure 2.28 is the graphical view of this tool implemented as a dedicated panel in the ATLLARLTDB project. The different buttons allow to create, delete, and configure all the data points of the project.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.3. Software

EMF

BARREL

A SIDE FLX-00 FLX-01 FLX-02 FLX-03

C SIDE FLX-04 FLX-05 FLX-06 FLX-07

END-CAP

FLX-00 FLX-01 FLX-02 FLX-03

FLX-04 FLX-05 FLX-06 FLX-07

LTDB List E01R/E01L

Driver Number 8

Server name PERIPH-OPCUA

Subscription name PERIPH_SUBSCRIPTION

DELETE_DP

CREATE_DP

DELETE_DP_DP_TYPES

CREATE_DP_DP_TYPES

CREATE_DP CLKDesScan

CREATE_DP GBTx

CREATE_DP LARBACKEND

Figure 2.28.: ATLLARLTDB panel used to create, modify, configure, and delete the data points of the ATLLARLTDB project.

The ATLLARLTDB project is composed of 22 different panels. These are separated in three categories depending of their purpose:

- FSM panels: These panels are made to be integrated with the global ATLAS DCS project. They are made to monitor the boards and do basic configuration of the system.
- Experts panels: These panels show more specific monitoring and actions. They are used for the recovery of the boards with issues. These panels are visible for everyone but only a user with the expert rights can perform actions.
- Management panels: Panels to create configure and manage the data points of the system. These panels are not accessible from the global project and are used only by managers of the project.

2.3.5. Trigger and Data acquisition system

The Trigger and Data Acquisition system (TDAQ) system is a software and hardware system responsible for configuring the full detector at the beginning of a data-taking run, for the calibrations procedures, and for the monitoring of the status of the data taking. Several TDAQ applications are needed to configure the system, collect the data, and monitor the quality of the data. The interdependence between the different applications and the corresponding hardware is managed by the so-called partition system. A partition contains several segments managing several parts of the detectors allowing them to work together. Several partitions can be created for parts of the detector that can operate independently of each other. Partitions can also be grouped to allow larger parts of the detector or for the full ATLAS detector to operate all together. The segments follow a tree hierarchy and each sub segment can be individually activated or deactivated. Figure 2.29 shows the hierarchy of the LTDB segment. A FELIX server controls all the LTDBs of a quadrant of the detector and is named FLX-LTDB-HUB. In the figure, the FELIX server 09 (FLX-LTDB-09-HUB) controls the LTDBs of the first quadrant of the barrel C.

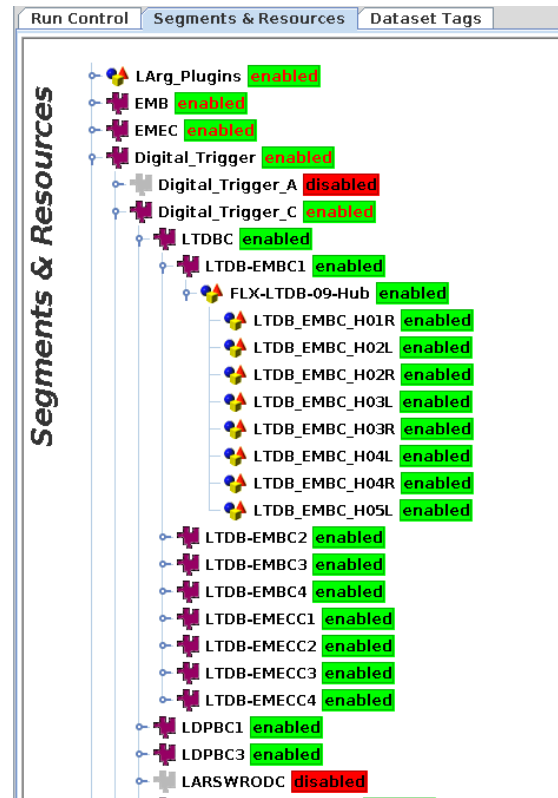


Figure 2.29.: Tree view of the LAr side C partition in the TDAQ system. The LTDB of the first quadrant of Electro-Magnetic Barrel side C are expanded in the tree.

A run control application controlling the LTDBs is included in the partition tree. This

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.4. DCS Monitoring of the LTDB boards

application is a client of the OPC-UA LAR LTDB server. It implements an application library named `UaoClientForOpcUaLarLtdbServer` (seen in section 2.3.3) to do the connection with the OPC-UA server. This application library is generated by quasar. The run application sends commands to configure the boards through the address space methods of the server.

2.4. DCS Monitoring of the LTDB boards

The ATLLARLTDB project is included in the ATLAS DCS global system. On this system the full ATLAS detector is represented by nodes with a tree hierarchy. ATLAS is the top node containing all sub-detector nodes. The LTDB boards are all included in a node named LTDB. The LTDB node is included in the infrastructure node which is included in the LAr node.

The ATLAS DCS project contains three main systems:

- The Alarms: These are alarms on each sub system data-point. They are reported in a global alarm viewer.
- The Database: Several databases contain the archived values of the data points.
- The Finite State Machine (FSM): This system displays the state and status of the nodes and implements commands to control them.

The FSM system is mainly organised around two values per nodes. These values are the state and the status. The state represents the state of the configuration of the node and the status indicates if there are errors on it with several levels. These state and status are propagated to the upper hierarchy. The state of an upper node is the state of the sub nodes composing it. The status of an upper node is the status most critical of its sub nodes. The FSM also implements commands. These commands can be executed on a top node and are propagated to all sub nodes. The following status are defined for the LTDB:

- OK: the LTDB slow control is fully reachable.
- WARNING: One or several SCAs are not online.
- ERROR: The five SCAs are not online.

The states for the LTDB are described in section 2.5. A global state of one quadrant or of the full detector is computed by combining the state of the corresponding LTDBs. These global states can be:

- SHUTDOWN: when all LTDB are in OFF state.
- NOT READY: when one LTDB is in ON or INIT state.
- READY: when all LTDBs are configured .

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.4. DCS Monitoring of the LTDB boards

The LTDB System implements two commands:

- Power ON: power on the LTDB boards.
- Power OFF: power off the LTDB boards.

Figure 2.30 is a graphical view of the LTDB FSM. The top node grouping all the LTDBs of the END-CAP A is displayed. The top node is in the not ready state since one of the LTDBs is in INIT state and all others in ON state. All the LTDBs have the OK status so the top node status is also OK.



LAR			
INFRASTRUCTURE			
LTDB			
LAR_LTDB_EMECA	NOT_READY	OK	
ltldb_A01L	ON	OK	✓
ltldb_A01R	ON	OK	✓
ltldb_A02H	ON	OK	✓
ltldb_A02L	ON	OK	✓
ltldb_A02R	ON	OK	✓
ltldb_A03L	ON	OK	✓
ltldb_A03R	ON	OK	✓
ltldb_A04L	ON	OK	✓
ltldb_A04R	ON	OK	✓
ltldb_A05L	ON	OK	✓
ltldb_A05R	ON	OK	✓
ltldb_A06H	ON	OK	✓
ltldb_A06L	ON	OK	✓
ltldb_A06R	ON	OK	✓
ltldb_A07L	ON	OK	✓
ltldb_A07R	ON	OK	✓
ltldb_A08L	ON	OK	✓
ltldb_A08R	ON	OK	✓
ltldb_A09H	INIT	OK	✓
ltldb_A09L	ON	OK	✓
ltldb_A09R	ON	OK	✓

Figure 2.30.: FSM of the ENDCAP A LTDBs. All the status are OK. One LTDB is in initialised state so the global status on top is NOT_READY.

For each node of the FSM, a detailed panel exists. Figure 2.31 shows the panel of monitoring of the ENDCAP partition included in the FSM panel. The different LTDBs are distributed on four circles depending on their type and the detector region that they cover. Going from the center we can find the FCAL LTDBs followed by the HEC LTDBs, the Special LTDBs, and finally the standard EMEC LTDBs. This detailed panel is required to be simple and robust allowing a non-expert to quickly identify the components with problems. In practice, non-experts that are on regular shifts to monitor the detector need to detect problems efficiently and notify the experts. The same values as the FSM are shown in the panel. The colors represent the state and

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.4. DCS Monitoring of the LTDB boards

status of the LTDBs and the display representing the detector geometrical positions allows to quickly identify the source of the issue.

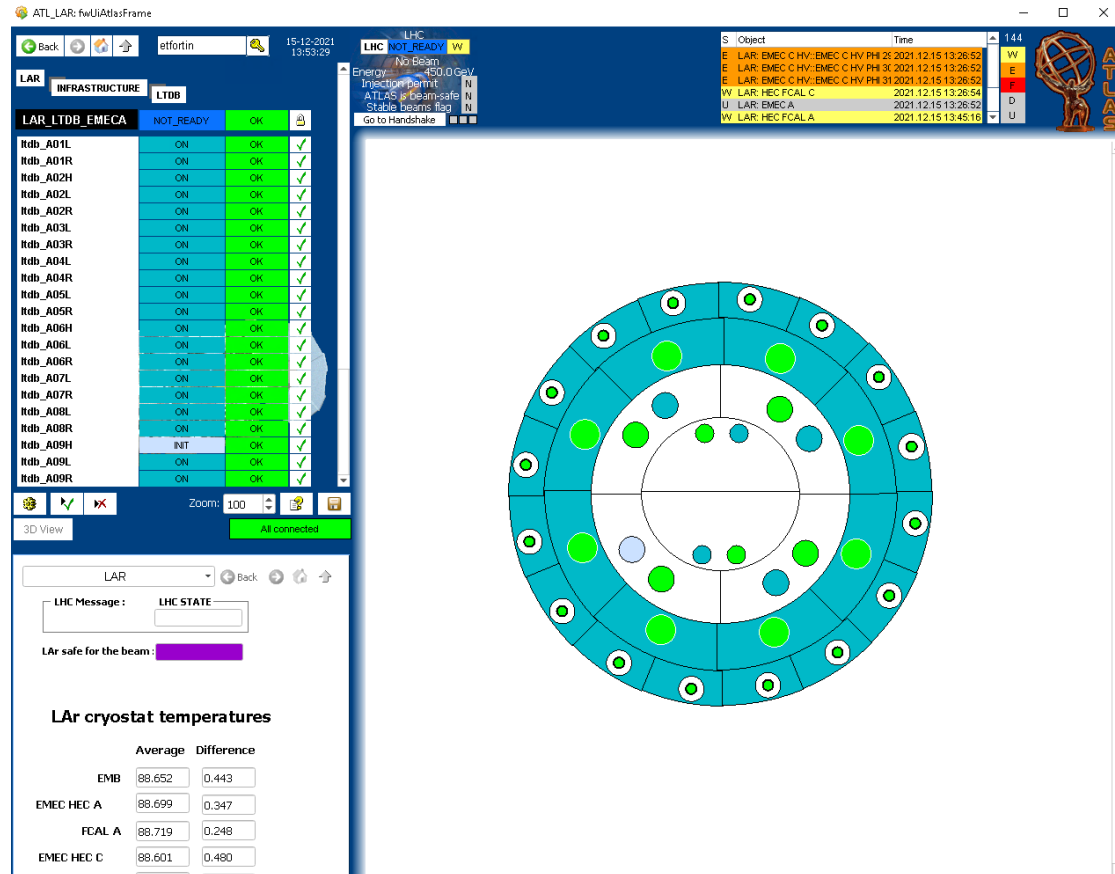


Figure 2.31.: Panel of the status of the ENDCAP side A LTDBs. On the top Left one the FSM for the ENDCAP A, on the top middle the FSM of the LHC, and in the top right all the alarms received by the top partition opened, here LAR. On the right is the main panel of the LTDB LAR partition and the secondary panels is shown in the bottom left, there is no secondary panels for LTDB project so the LAR one is displayed.

The LTDB panel shown in figure 2.32 is associated with the LTDB node. Values from the PDB are shown on the left of the panel. These correspond to voltages being read by the SCA #1 internal ADC and the currents computed by the OPC-UA LAR LTDB server. The currents are computed in the server from the measured voltages. Two voltage values are read on both sides of a resistor to compute the current using Ohm law. Three power status are provided by the regulators on the PDB. For each SCA of the LTDB several temperatures values are displayed: two are measured in the analog part of the board and two in the digital part. On the GBTx more than 400 registers exists, the most important ones are displayed in the panel of the LTDB node. On the Wincc OA project the values of the SCA and of the PDB are archived. The values can

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.5. Control of the board

be displayed as a function of time by a simple right click. This display is shown on the bottom left of figure 2.32. When connected as expert, this panel allows to reset the GBTx and SCA components.

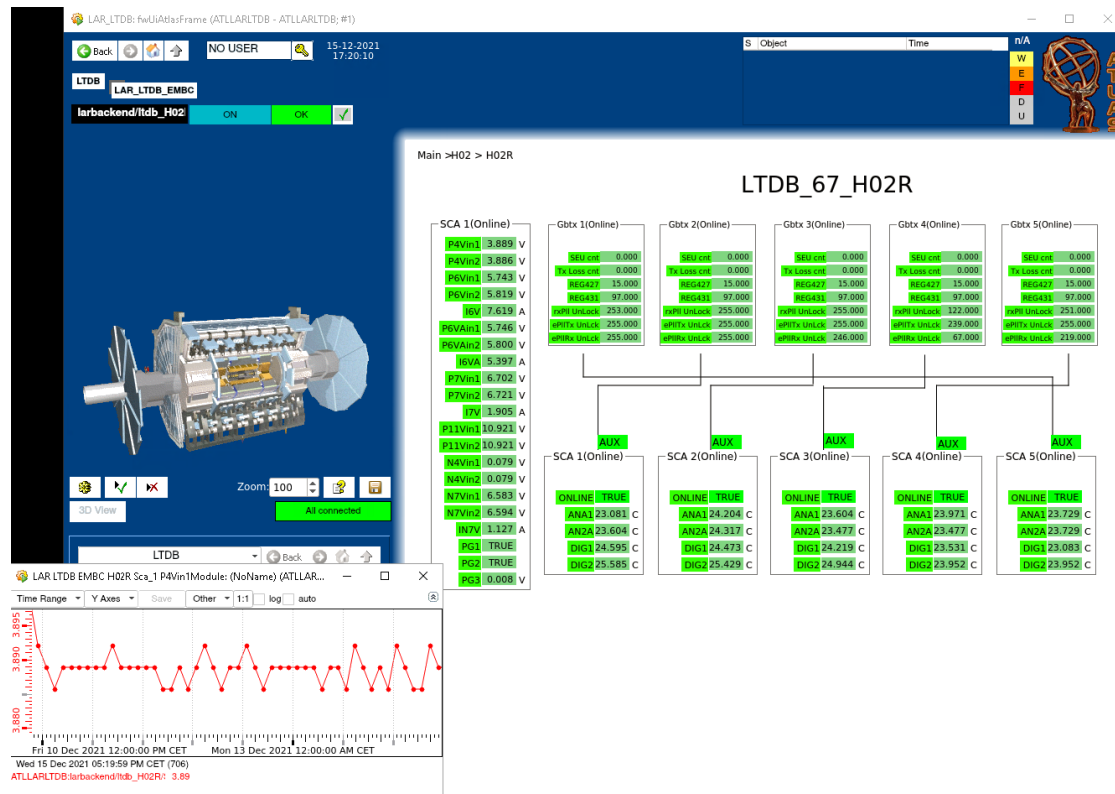


Figure 2.32.: Monitoring panel of the LTDB. Values from the PDB, GBTxs, and SCAs are available. These values can be displayed as a function of time by a simple right click.

2.5. Control of the board

The configuration of the LTDB is organised into different steps. Some of the steps are used for default configuration and some for recovery of boards with issues. The default configuration steps are named transitions. The transitions change the state of the LTDB. Figure 2.33 shows the five different states of the LTDB and the transitions between them.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.5. Control of the board

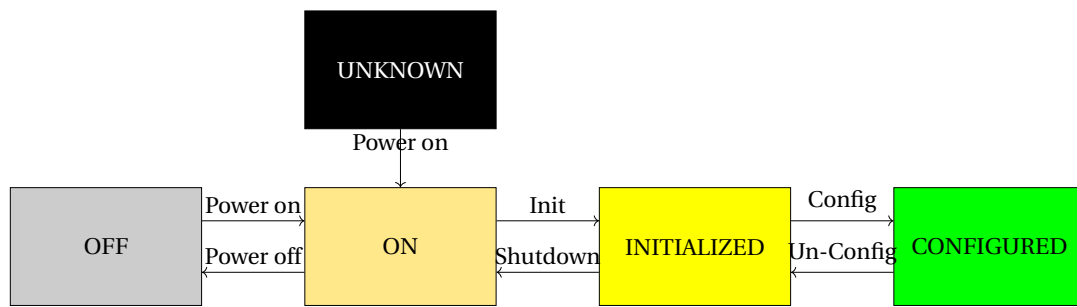


Figure 2.33.: LTDB transitions and states.

Six transitions control the board, some transitions are restricted to the DCS client. The transitions are detailed in the following list:

- Power on: During this transition the following actions are performed:
 - Switch the PDB regulators ON.
 - Switch the state to ON.
- Init: During this transition the following actions are performed:
 - Check that the IDs of the SCAs correspond to the targeted board.
 - Configure the MTRx control links transceiver.
 - Load the XML files describing the configuration of the board in the memory of the server.
 - Load the mapping file in the memory of the server.
 - Switch the state to Initialised.
- Configure. During this transition the following actions are performed:
 - Configure the GBTx.
 - Reset and configure the MTxs.
 - Reset and configure the LOCx2s.
 - Reset and configure the ADCs.
 - Load the delay inside the LOCx2s.
 - Load the MDAQ correction coefficients inside the ADCs.
 - Put the ADCs in data mode.
 - Switch the state to configured.
- UnConfigure: During this transition the following actions are performed:
 - Switch the state to Initialised.
- Shutdown: During this transition the following actions are performed:

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.5. Control of the board

- Switch the state to ON.
- Power OFF: During this transition the following actions are performed: (DCS client only)
 - Switch the regulators OFF.
 - Switch the state to OFF.

Before each transition, The OPC-UA LAR LTDB server checks that the state of the board is compatible with the called transition. Five different states are implemented for the control of the LTDB:

- Unknown (UNK): The OPC-UA LAR LTDB server is not able to know the state of the board. This can be due to several reasons: the OPC-UA LAR LTDB server is not started, an SCA is reset, or the connection to the OpcUaSca is lost.
- Off (OFF): All power regulators on the PDB have been switched off .
- On (ON): All power regulators on the PDB have been switched on .
- Initialised(INIT): Board is reachable, the server is ready to configure the board.
- Configured (CONFIG): Board is reachable and is configured for data taking

When the power regulators are switched on the temperature of the board increases. Frequent temperature changes may damage the boards so it has been decided to have them always powered on. In this state the board is idle. To be ready for data taking the board needs to be first initialized . In this state the configuration of the control path is done and the server has loaded in memory all the configurations files. When configured the board data paths are fully configured.

The LTDB boards are installed in the detector cavern with relatively high radiation. This can cause some Single Event Upset(SEU) on configuration registers that leads to data corruption. A reconfiguration of the LTDB is needed before each data taking run to reduce the effect of these SEU. For that the LTDB is put back in idle state (ON) after each run. The current state of each LTDB is computed by the the OPC-UA LAR LTDB server and provided in it address space.

2.5.1. Control by the TDAQ Partition

At the beginning of each data taking run the LTDBs are configured by the TDAQ run control application. Figure 2.34 represents the control application tree for the LAR partition in a running state. On the top left, a command to configure the system is available. This command is propagated to each of the run control applications that configure the corresponding components. For the LTDB a run control application (rcApp) controls each of the Felix Hubs (Felix boards). On the bottom part the alarm

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.5. Control of the board

window is displayed. If the OPC-UA LAR LTDB server raises warning or error messages during the configuration of the LTDBs, it will be displayed in this alarm window.

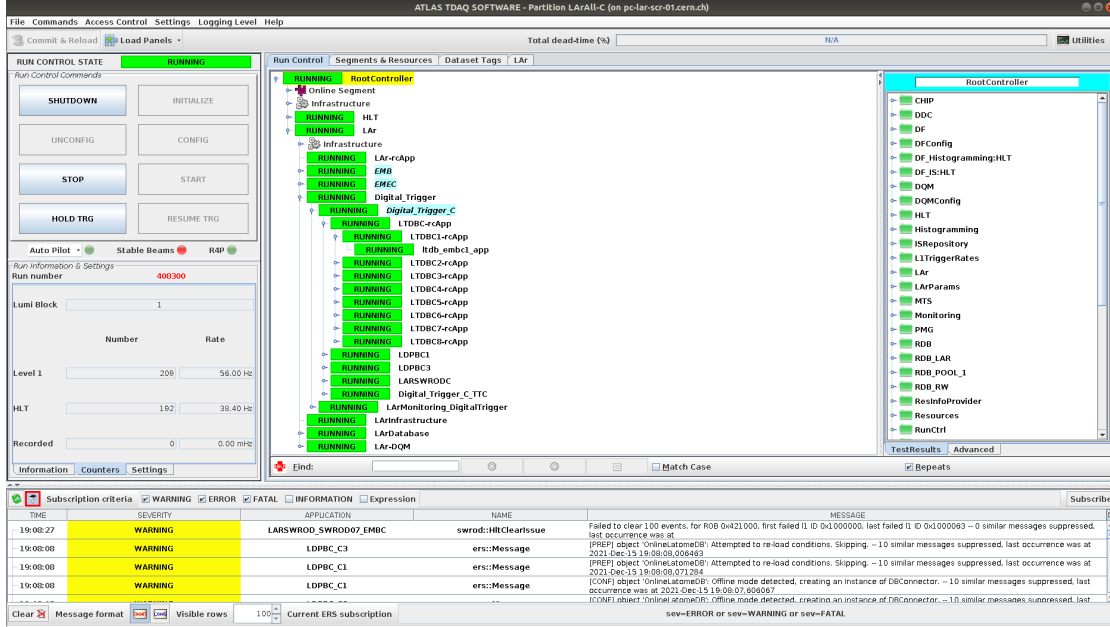


Figure 2.34.: GUI of the TDAQ system representing a partition with its different run control applications in running state. The central view focus on the LTDB System.

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.5. Control of the board

2.5.2. Control by DCS panels

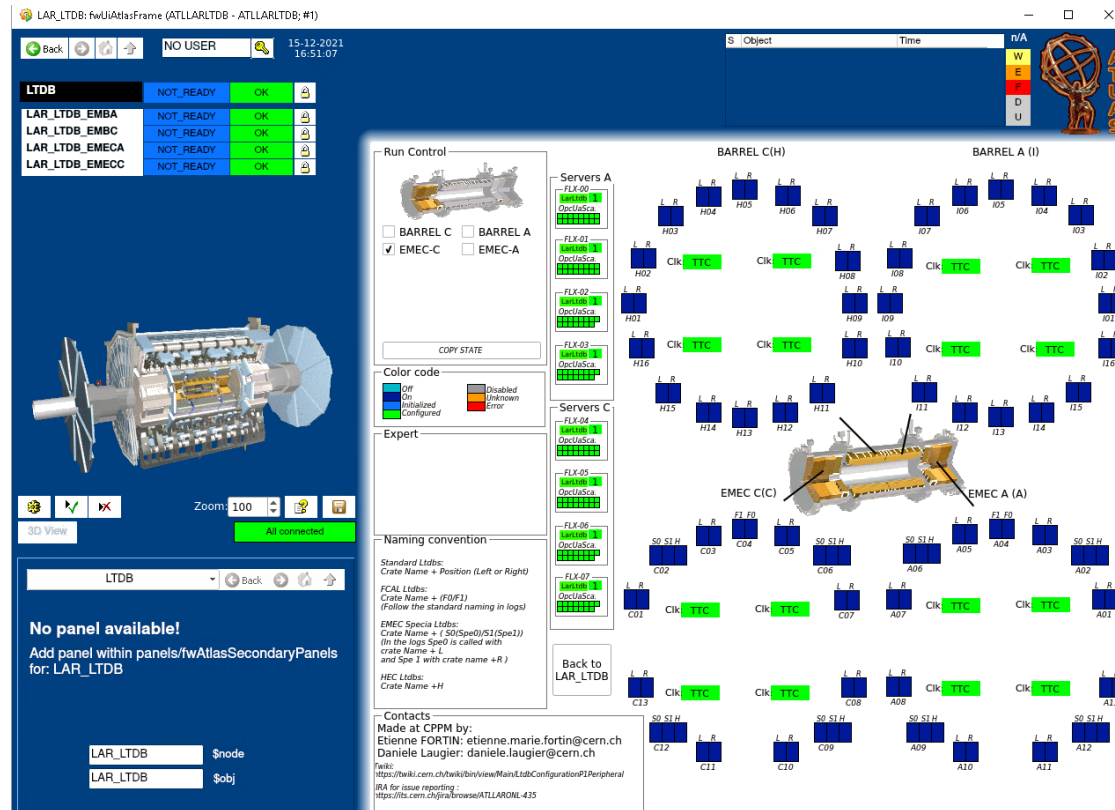


Figure 2.35.: Expert main panel seen with a non expert account.

The DCS system allows the full control of the LTDBs and in particular some critical operations such as the power on/off of the board and the reset of GBTx and SCA components. The resets are not done in the standard LTDB configuration but are used for recovery of the board. The other transitions are possible with DCS but done by TDAQ during the standard operation of the LTDB.

The panels controlling the LTDBs are visible for every user but the action buttons are masked for non-experts user. The configuration of the full LTDB system can be done on the panel shown in figure 2.35 but can also be done per LTDB on the panel shown in figure 2.36. This panel is separated in two or three parts depending on the crate. Each part contains data and action buttons for one LTDB. The displayed data are:

- State of the LTDB with its timestamp.
- Status of the links with the timestamp of the last check.
- Detailed logging of the last transition.

For each LTDB several buttons allow to perform various board transitions and other buttons allow to open expert panels for specific actions:

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.6. Calibration of the LTDBS boards

- Scans panels: Allow to do the clock delay calibrations of the links using the LATOME.
- Mapping panel: Allows to generate and check the mapping with the LATOME.
- ADC panel: Allows to configure the ADC in pattern or data mode.
- Debug panel: This panel allows to read back each register of the board and compare it to the expected value.

I developed all the panels of the LTDB systems. I modified them to be compatible with the ATLAS DCS global project. They are actually fully integrated in the DCS project of ATLAS.

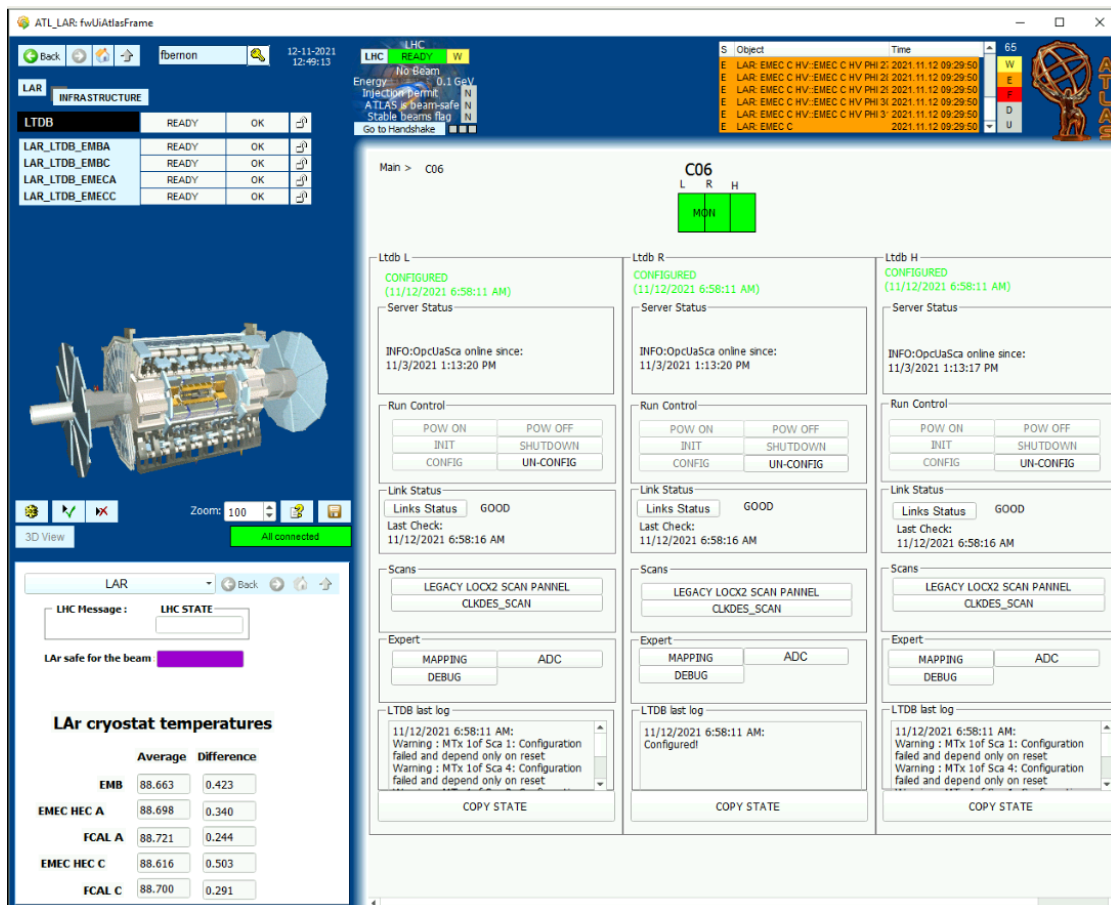


Figure 2.36.: Expert panel for a special crate containing 3 LTDBs

2.6. Calibration of the LTDBS boards

Several calibrations and scans are needed to properly operate the LTDBs:

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.6. Calibration of the LTDBS boards

- ADC calibration: This calibration aims to find the best parameters for the digital corrections inside the ADCs.
- Link calibration: The phase of the clocks of the LOCx2 needs to be adjusted to correctly encode the LOCic frame.
- Mappings scans: These scans are used to validate the connections between the LTDBs and LATOMEs.

2.6.1. ADC calibration

As detailed in section 2.2.1, to correctly operate the LTDBs, the MDAC stages of the ADC must be calibrated. An initial calibration was performed during the production of the boards. However this calibration needs to be updated regularly to take into account various changes of the boards including aging or temperature effects especially since these boards will operate for more than 15 years. The update frequency for this calibration and its stability is not yet known and will be studied during the lifetime of the boards. This calibration [22] consists of measuring the jumps caused by each MDAC level and compute coefficients to correct it. To do so, four outputs need to be known for each MDAC with different reference voltages. In total, sixteen calibration modes exist per ADC, methods to switch all ADC to each of the mode are implemented in the OPC-UA LAr LTDB server. The calibration is managed by the TDAQ system. Data are collected with the ADCs configured in each of the 16 modes. The TDAQ system configures the LTDBs and all the boards needed to collect the data, mainly LDPB and FELIX boards. An analysis is then performed to compute the MDAC coefficients from the collected data. The computation consists of comparing the actual gain of the MDAQ stage to the optimal one as described in the ADC measurement datasheet [9]. The computed coefficients are saved and can be used in the next configurations of the LTDBs.

2.6.2. Link calibration

As describe in section 2.2.1.2, the phase between the edges of the ADCs clocks and the LOCic encoder internal clock must be adjusted for a correct encoding of the LOCic frame. Two ADC clocks at 320 MHz are received by one LOCic encoder block. A delay can be added for these two clocks before the LOCic. This delay is common to both ADCs. The LOCic encoder internal clock frequency is 320 MHz. This clock is derived from the 40 MHz clock coming from the GBTx and is used by two LOCic encoders. Inside the GBTx this clock can be delayed. Two methods of adjusting the phase between the clocks are possible: the first one by adding a delay on the clock coming from the ADCs, the other by adding a delay on the input clock coming from the GBTx. Two specific calibrations were developed, one for each method.

2.6.2.1. LOCx2 phase shift delay calibration

The first calibration method changes the phase of the input clock coming from the ADCs. For each LOCic encoder a delay is chosen and applied to the input clock of the two ADCs. All possible delay values are scanned and the corresponding error rate is checked on the corresponding fibers on the LATOMEs. The possible values range is between 9 and 25 with each step corresponding to a delay of 220 ps. The error rate is checked during 5 seconds for each fiber. This value is chosen as a compromise between the scan duration (2 min in this configuration) and the possibility to detect low error rates (> 0.2 Hz in this case). The presence or absence of errors is then checked for each delay as shown in figure 2.37. In this figure "1" in green represents delays with no errors and "0" in red the delays with errors. The best delay is chosen to be the center of the largest continuous range with no errors.

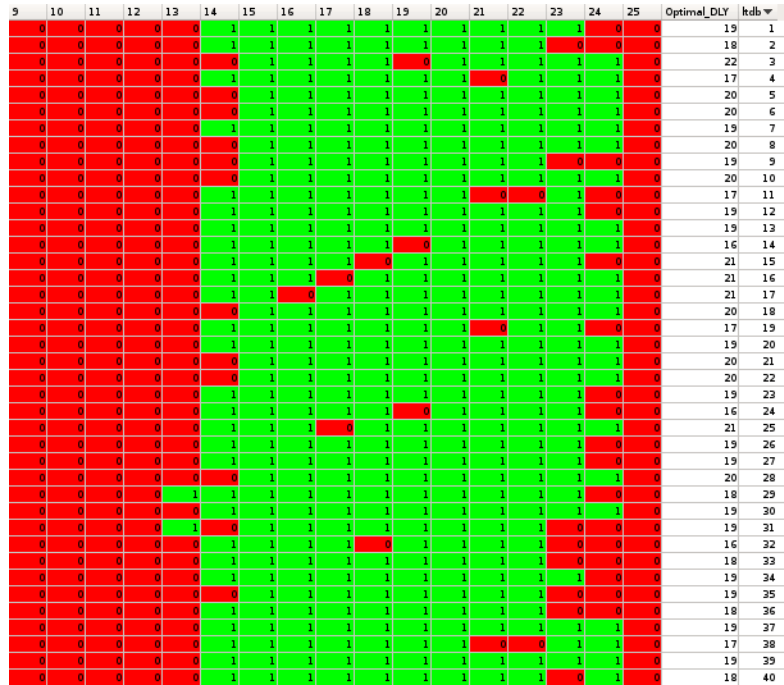


Figure 2.37.: View of the results of the LOCx2 phase shifter scan inside the ATL-LARLTDB dedicated panel. Values at 0 are shown as red and the ones at "1" in green for better clarity. The values are shown for the 40 fibers present on one LTDB(vertical axes) and for each possible delays(horizontal axes).

This algorithm has the disadvantage of not detecting low rate errors without increasing significantly the duration of the scan. A better method is to find the worst case delay and choose a delay which is at the maximum distance from this worst case delay. The worst delay corresponds to the delay with the peak of errors. While moving from this peak the error rate decrease. This delay cannot be found by the LOCx2 scan as it cannot scan the full period of the 3125 ps period of the LOCic internal clock.

2.6.2.2. LOCx2 input phase calibration

The origin of the 40 MHz input clock of the LOCx2 is the GBTx. On this component the clock can be delayed by eighty steps of 48.8 ps. The period of the 320 MHz clock being 3125 ps, 64 delays cover the full range. As in the first calibration method, all possible delay values are scanned and the corresponding error rate is checked on the corresponding fibers on the LATOMEs. In figure 2.38 the scan results are presented with a "1" for delays with no errors and "0" for delays with errors. Since the full range of the clock is scanned, the scan always finds the worst delay where the LOCic frame is badly encoded with high error rate. One block of consecutive delays with errors are seen for each fiber. The best delay is at the middle of the interval between the worst delays of the two fibers using the same GBTx clock.

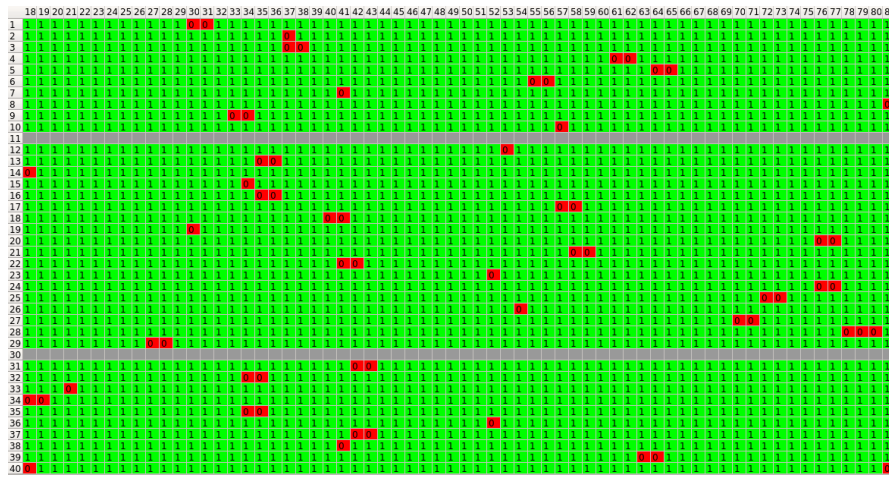


Figure 2.38.: View of the results of the CLKDES clock phase shifter scan inside a ATL-LARLTDB dedicated panel. Values at 0 are shown as red and the ones at "1" in green for better clarity. Fibers in grey are the ones unused (disabled by the mapping). The values are shown for the 40 fibers present on one LTDB (vertical axes) and for each possible delay (horizontal axes).

This method and calibration have shown better stability over time on the detector. Furthermore the duration of the scan is reduced compared to the LOCx2 phase shift one.

2.6.3. Mapping

The LTDB system contains a total of 1240 optical fibers for the control of the LTDBs and 4960 fibers for sending data to the LDPB system. The connections between the boards are complex and an automatic validation is mandatory. For that, 3 scans are developed to validate 3 different mapping aspects of the boards: the mapping of the control fibers between the FELIX and the LTDB, the mapping of the data fibers between the LTDB and the LDPB, and the mapping of the ADC channels between the LTDB ADCs channels and the LDPB channels.

2.6.3.1. Control fiber mapping between the FELIX and the LTDB

As seen in figure 2.39, each FELIX server contains 2 FELIX boards, one is used for barrel LTDBs the other for ENDCAP LTDBs. Each FELIX server handles one quadrant of the detector including barrel and end-cap. Each FELIX board has two MTP-48 connectors connected to the same wendy box. This box remaps 90 fibers of the two 48-fibers cables into 9 MTP-12 connectors. On these connector 12-fibers cables are connected to the LTDBs. Two fibers per cable are not used. The barrel and endcap wendy boxes are different since they handle different mapping (presented in tabular 2.1). Each server handles at maximum eight LTDBs, so only eight of the nine connectors are used. The remaining connector is kept as spare.

To validate the mapping a scan based on the SCA IDs is developed. The IDs of the SCAs for each LTDB are known and collected in the LTDB production database. The scan reads the SCA ID connected to a specific link and compares it to the expected value from the production database. This allows to automatically verify all the cabled links after the installation. This is now in operation routinely.

2.6.3.2. Data fiber mapping between the LTDB and the LATOME

As detailed in table 2.4 and table 2.2 LATOMES and LTDBs exist in different hardware variations with different mapping. The interconnections between several LTDBs and several LATOMES depend on the geometrical position on the detector and is relatively complex. A LATOME can receive fibers from up to 3 LTDBs and an LTDB can have fibers going to up to 4 LATOMES. More details about the LTDB/LATOME fiber connections can be found in table 2.8.

LATOME type	Input From LTDB Type	LTDB type	Output to LATOME Type
EMB	2xEMB	EMB	EMB and EMB/EMEC
EMB/EMEC	2xEMB 1xEMECSpe0	EMEC	EMEC
EMEC	1xEMEC	EMECSpec0	EMB/EMEC
EMEC/HEC	1xEMECSpe1 1xHEC	EMECSpec1	EMEC/HEC
FCAL1	FCAL1	HEC	EMEC/HEC
FCAL2	FCAL2	FCAL1	FCAL1
		FCAL2	FCAL2

(a) LATOME side

(b) LTDB side

Table 2.8.: LTDB and LATOME connections. For each LATOME types the type of connected LTDB is shown. For each LTDB types the type of connected LATOME and its amount is shown.

To be able to validate this mapping a method to generate and check the mapping has been created on the OpcUaLArLtdbServer. To generate the LTDB/LATOME fiber

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.6. Calibration of the LTDBS boards

mapping for a specific LTDB the following algorithm is applied:

- Power off all LTDBs of the quadrant. This leads to a bad encoding on all the corresponding fibers and therefore errors seen on the LATOME boards for these fibers.
- Power on and configure the wanted LTDB.
- Switch all LOCx2 of the LTDB delay to 9, a delay where the LOCic frame encoding is always in error.
- For each fiber on this LTDB:
 1. Select a valid LOCx2 delay
 2. Check on all LATOMEs belonging to the quadrant if a specific fiber has a valid LOCic frame.
 - If one is found save the LATOME and the fiber ID.
 - Else go back to step 1 with a different delay, if all delays are tried without success this fiber is not found.
 3. Put back this fiber delay to 9.

This algorithm was used to generate the mapping of the full detector. Several mistakes were detected during the cabling, in particular inverted connections on different MTP12 connectors of the LATOME. This is now an official tool to generate and validate the mapping in case of cabling changes.

2.6.3.3. Channel mapping between ADC channels and LATOME channel

To validate the mapping between an ADC on the LTDB and channels of the LATOMEs, the ADC pattern mode is used. When the ADC is in such mode the value of the channel in the LATOME is expected at a known value. Only one channel in the LTDB is configured in pattern mode and the LATOMEs channels are scanned to find where the value is equal to the pattern. The value is read through registers implemented in the LATOME firmware using the IPBus protocol. This mapping generation was used to validate and correct issues in the files defining the mapping of each LTDB. It is an official tool to validate a board.

2.6.4. Error mitigation in the software

During the configuration of the LTDB or the communication between the LTDB and the control system, several errors can occur. A strong and robust mechanism of error detection is needed to operate the LTDBs without loss of data taking efficiency. Most of the errors come from the I2C communication. To mitigate these errors a mechanism of write, read and compare was implemented for all the I2C write operations. Each configuration value is written, readback and compared to the expected value. If one of

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.7. Board Installation and validation

these steps fails the full write, read and compare operation is retried. Several retries are performed before reporting a critical error causing the abortion of the configuration of the I2C register. For most of the components when such a failure is reported the LTDB configuration is aborted and the error is propagated to the OPC-UA clients. For the ADC components the configuration is not aborted, a full reconfiguration including a reset of the ADC is attempted before aborting the configuration of the LTDB. Without this retry mechanism, the configuration of the board is impossible. At each configuration of the LTDB between one and five errors occur and are recovered by the server. This retry mechanism is now deployed in production and allows to reliably configure the boards.

2.7. Board Installation and validation

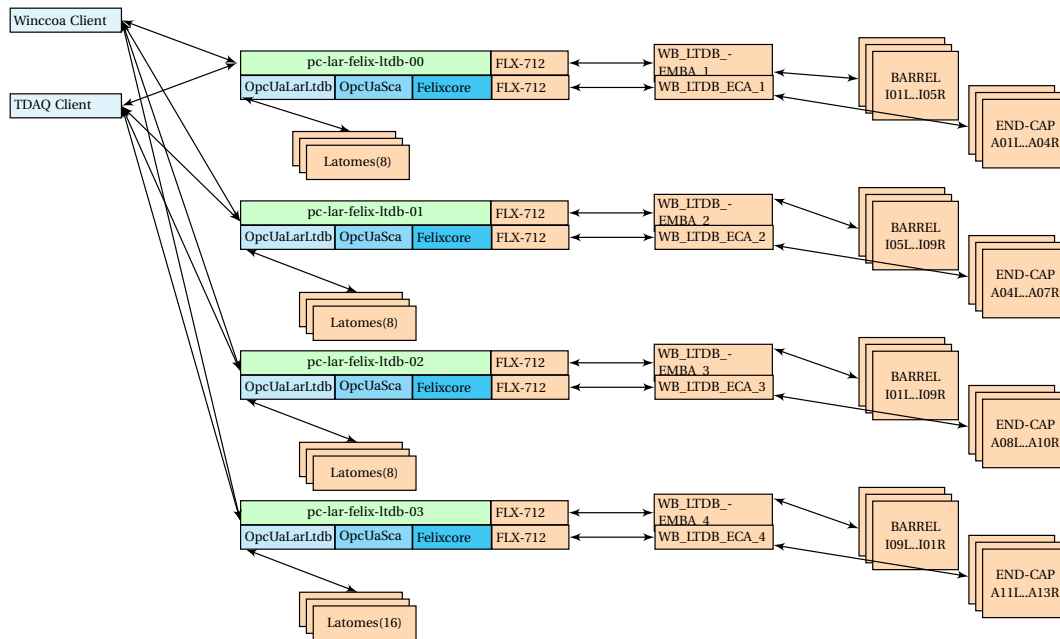


Figure 2.39.: Schematic of the connections between the different hardware and software of the LTDB system. The hardware components are in orange. The software components are in shades of blue. Each FELIX server contains two FELIX boards and hosts the OPC-UA servers and the felixcore software. Each FELIX board is connected to a wendy box and controls up to 9 LTDBs.

Figure 2.39 shows the hardware and software components used in the control system of the 64 LTDBs of the A side of the detector. The C side follows exactly the same organisation. On each FELIX server, instances of several software applications are running. The FELIX servers are dedicated to the LTDB system and no other applications are running on them. Each FELIX server controls the LTDBs of one quadrant of the

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.7. Board Installation and validation

detector. The end-cap and barrel LTDBs are separated on the two FELIX boards for each server.

The installation of the LTDBs in their crates took several months. To access the crates, dedicated scaffolding were installed. It was mandatory to validate the LTDB boards as soon as they were installed to be able to remove the scaffolding. The following procedure describes what was needed after the board insertion in its crate and its cabling.

- Validation of the control links cabling: Validate that all links are running and target the expected SCAs.
- Validation of the slow control path and I2C components : Validate that all components are configurable and correctly answer the OPC-UA requests.
- Validation of the data links: Validate that all the data links with the LATOME are working.

The first step is to validate the control links. Without these links no control can be done on the LTDB. During the cabling of the LTDB to the FELIX, a measurement of the optical power of the links is done with an external tool. It validates that all the MTRx are emitting with the correct optical power. After the cabling this measurement is also checked on the FELIX board with its internal optical power measurement. The optical power measure is followed by a scan of the mapping between the FELIX and the LTDB as presented in section 2.6.3.1. This scan validates that all SCAs are responding and that the mapping is correct.

The second validation step is to validate the I2C communications on the board. During this step the configuration for the LTDBs for the OPC-UA servers is generated and the LTDB added to the DCS Wincc OA project. Several configuration attempts are tried with different parameters. These attempts are controlled by the DCS system that checks if all components respond correctly to the commands.

The final step is to validate the data links. During the cabling between the LTDB and the LATOME, an optical power measurement is done. To validate the mapping between the LTDB and the LATOME the two mapping scans described in section 2.6.3 are performed. After this validation the delay scans are realised. After the verification of the configuration, data are collected by the TDAQ system to check that all channels are sending reasonable values.

2.7.1. Results

The installation and validation period ended in October 2021. All 124 LTDBs of the detector are validated. During the validation 19 LTDBs listed in table 2.9 were found to be problematic and were exchanged. Eleven of them had issues on the data links with the LATOMEs, detected during the mapping scans. Six LTDBs were extracted repaired and reinstalled on the detector because of an issue with the GBTx I2C bus being always grounded. This issue was observed during configuration of the board

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.8. Conclusion

Number of LTDB	Issue
6	Fiber with LATOME in error: no LOCic frame decoded
5	Fiber with LATOME in error: no light received
1	SCA unstable
1	MTx not responding to I2C and not working
6	I2C GBTx bus grounded

Table 2.9.: List of LTDBs replaced during the validation period

Component	Number of LTDB affected	Issue
Ctrl link optical power	1	High error rate link
Ctrl link	4	No light
ADC	2	I2C read not working
ADC	1	I2C read always at 0
LOCx2	2	I2C write not acknowledged
LOCx2	2	Bit stuck in register
MTRx	1	I2C not working

Table 2.10.: List of errors on the LTDBs installed in the detector. These errors are mitigated by a dedicated configuration on the control software.

and confirmed with low-level tools. This I2C bus is connected to test pins used during board production. These pins must be cut at board production before the cooling plates installation. On the faulty buses the pins were not cut properly and found to be touching the cooling plates. The cooling plates are grounded causing an issue on the I2C bus. Several other issues were observed on some components. The replacement of all the faulty LTDBs was impossible because of cost, time and access issues. Table 2.10 lists all these issues. For LTDBs with issues on one control link, the redundant one is used to control the board from the OPC-UA server. If the component is in the data path, tests were developed to determine if it is possible to operate the board without corruption. If found to be impossible, the board was replaced. For each component with non-critical errors a dedicated configuration procedure has been developed and the board was not replaced. I was in charge of finding, developing and testing all these procedures.

2.8. Conclusion

In total 124 LTDBs were installed on the detector. Their control system is based on several servers mainly based on the OPC-UA protocol. The most complex software in this chain is the OPC-UA LAR LTDB server. The development of this server was challenging as it must be robust and must mitigate several hardware issues. The OPC-UA LAR LTDB server possesses several interfaces with various components such as the LTDB and LATOME hardware, the FELIX, the DCS, and the TDAQ systems. This server was optimized to reduce the resource consumption and the configuration time

2. Control and Monitoring of the front-end boards of the phase-I upgrade – 2.8. Conclusion

of the LTDBs. It also includes several recovery procedures to handle hardware errors. All these functionalities were successfully tested during the LHC pilot run in October 2021 and the server is being used without notable problems since then. This server is fully finalized and integrated with the ATLAS system. It is deployed and ready for the operation of the new phase-I components during the run 3 of the LHC, seen to start in June 2022.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade

Summary

3.1	Hardware components of the LAr phase 2 upgrade	116
3.2	Reconstruction of the energy deposited in the LAr calorimeter	119
3.3	Neural network theory	121
3.4	Neural Network for LAr energy reconstruction	124
3.4.1	Recurrent Neural Networks	124
3.5	Neural network firmware considerations	129
3.5.1	Internal neural network operations	129
3.5.2	Multiplexing and pipeline	130
3.5.3	Latency	130
3.5.4	Multiplications	131
3.6	Firmware Implementation	132
3.6.1	Intel FPGA blocks	133
3.6.2	The Intel High Level Synthesis	135
3.6.3	Implementation of Matrix Multiplication	136
3.6.3.1	Design with power of two weights	140
3.6.4	Activation functions	143
3.6.4.1	Implementation of the computation of a RELU	143
3.6.4.2	Implementation of the sigmoid function	144
3.6.5	Fixed point representation	146
3.7	Firmware performance	153
3.7.1	HLS resource estimate	154
3.7.2	Implementation and testing on hardware	154
3.8	Conclusion	158

3.1. Hardware components of the LAr phase 2 upgrade

During the phase 2 upgrade of the LAr calorimeter of ATLAS the full readout electronics will be replaced. The FEB and the ROD will be removed. The new readout data path

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.1. Hardware components of the LAr phase 2 upgrade

will be composed of two new boards. The front-end Board version two (FEB2) will be installed in the Front-End crates replacing the FEB. The ROD will be replaced by a new board named LAr Signal Processing board (LASP). The FEB2 will be in charge of amplifying, shaping and digitizing the analog signals of the detector cells. The FEB2 will transmit the digitized data to the LASP board installed in the USA15 counting cavern. The LASP board will compute the energies from the values of the analog to digital converters(ADC) embedded in the FEB2 board. The ADC values will be transmitted to the trigger system. The computed energies will be transmitted to the Trigger System and to the ATLAS data acquisition system.

The LASP board is an ATCA board containing FPGAs. Two Intel Stratix 10 FPGAs are responsible of the processing of the data received by high speed electro-optical transceivers. These transceivers will be used for the input and output connections of the board. The input fibers will come from the FEB2 with up to four FEB2 connected to one LASP FPGA. 22 input optical links will be needed per FEB2. In addition fast output links will be needed to send the output data to the trigger system and several bi-directional links will be needed for data readout and control. A demonstrator board for the LASP is currently available. This demonstrator board seen in figure 3.1 is composed of two different Stratix 10 FPGAs. One FPGA is a Stratix-10 MX and the second one is a Stratix-10 SX. Both FPGAs provide enough inputs/outputs to allow the processing of data from 4 FEB2 per LASP FPGA. The Stratix-10 MX contains a large amount of memory blocks and the Stratix-10 SX contains instead a System-On-Chip processor. This demonstrator board is a proof of concept that a board with 2 Stratix-10 FPGAs and high component density is possible. It will allow also to characterise the performance of the SX and MX FPGAs from INTEL.

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.1. Hardware components of the LAr phase 2 upgrade

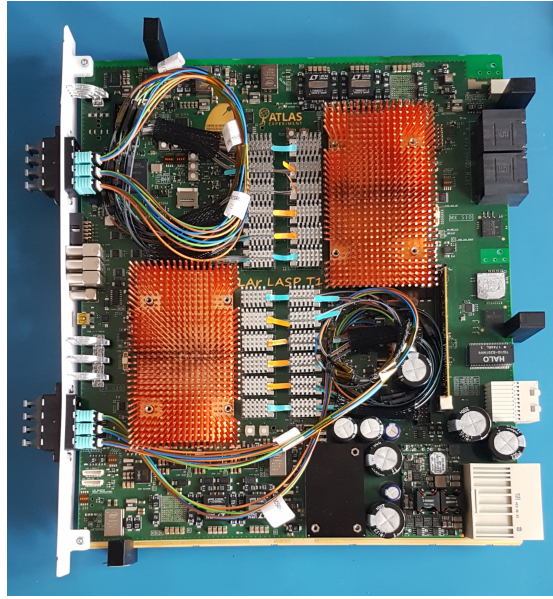


Figure 3.1.: Picture of the LASP demonstrator board. The two FPGAs are under the two copper heat sinks. The optical emitters/receivers are under the aluminium heat sinks.

Figure 3.2 show a schematic of the LASP board firmware. This firmware is organised in different blocks:

- The input alignment block. This block synchronises the data that correspond to the same LHC bunch crossing from the different input links.
- The remapping block : This block receives the data from the input alignment block. The remapping block reorganises the data in region of interest to simplify the logic for the trigger computations.
- The data processing core block : This block is located after the remapping block. It computes the deposited energies in the calorimeter.
- The trigger packer block: It receives the energies computed in the data processing core block. Packets are formed containing these data and are send to the trigger system.
- The data monitoring block: This block receives the raw ADC data, the energy data and store them in buffers while waiting for the trigger decision.
- TTC block: It receives the trigger and timing data and transmits them to all the blocks that need them.
- Slow control block: This block will receive/transmit all the slow control data and propagates it to all other blocks.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.2. Reconstruction of the energy deposited in the LAr calorimeter

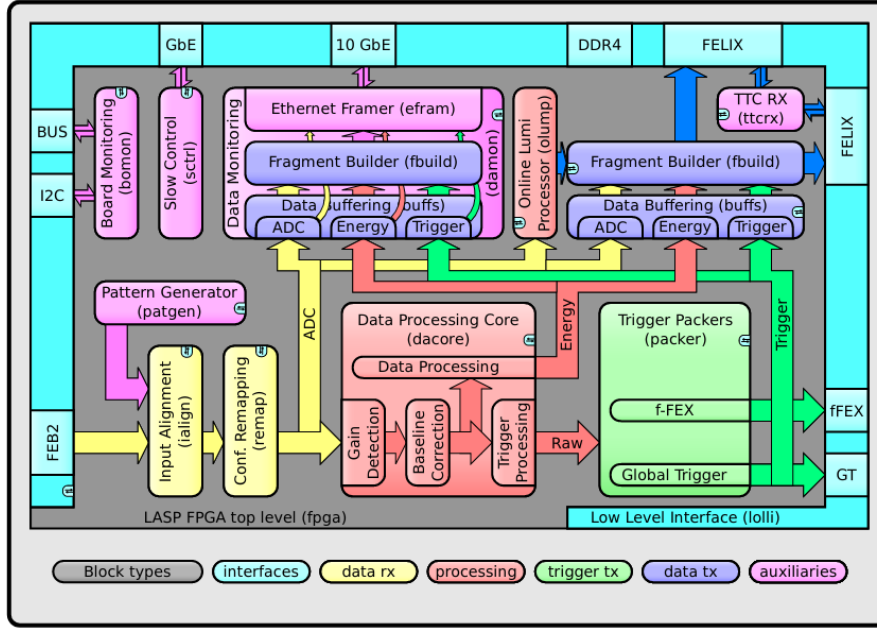


Figure 3.2.: Schematic of the block organisation of the LASP firmware and the data flow between these blocks.

3.2. Reconstruction of the energy deposited in the LAr calorimeter

The energy reconstruction for the LAr calorimeter is performed using the digital data coming from the front-end boards. Digital filtering algorithms are needed to reduce the noise by combining several ADC samples. The digital filtering is also in charge of finding the peak of the pulse since the sampling may not be at the peak. During the run 3 the Optimal Filtering (OF) will be used. This algorithm computes the pulse amplitudes and its time(phase) from four samples. The max finder algorithm can be used to determine the BCID of the pulse [6]. The OF algorithm is sensitive to the presence of pileup noise. The pileup noise is the noise caused by past energies deposits in the calorimeter cell. A pulse signal caused by the energy deposited in the detector is longer than the period between two collisions. If an energy is deposited in a LAr cell before the end of the pulse of a past deposit, the two pulses will overlap. The optimal filtering algorithm shows good performance to correct the electronics noise only in low pileup conditions.

With the HL-LHC increased luminosity, the pileup noise will increase. The average time between two pulses in a given cell will decrease. Figure 3.3 shows the resolution of the OF algorithm as function of the time gap between two pulses. For a gap lower than 20 BCs, corresponding approximately to the length of a pulse, the computed energy is smaller than the real deposited one.

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.2. Reconstruction of the energy deposited in the LAr calorimeter

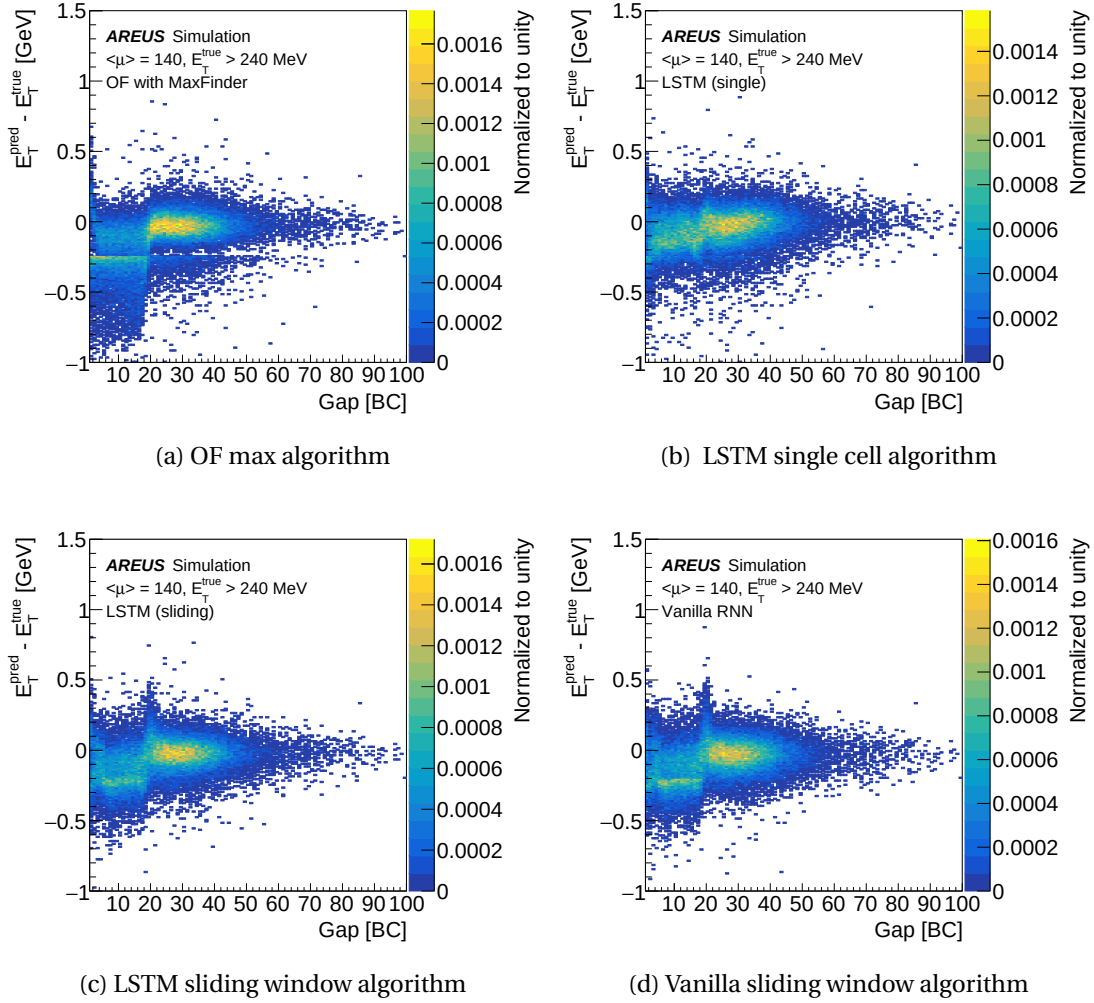


Figure 3.3.: Resolution of the transverse energy reconstruction as a function of the gap, i.e. the distance in units of BC, between two consecutive energy deposits for four different energy reconstruction algorithms. This plot is using AREUS[23] simulation in HL-LHC conditions[3].

New algorithms are under development to improve the energy reconstruction at the HL-LHC. Several extensions of the OF algorithm or the Wiener filter have been tested [10]. Furthermore new approaches using neural networks are under investigation. The algorithm providing the best compromise between performance and resource utilisation on the LASP FPGAs will be chosen. The LASP Stratix 10 FPGA provides limited computing blocks and is required to simultaneously compute the energies for several hundreds of LAr channels.

3.3. Neural network theory

The first neural networks were describe and modelled by scientist working on understanding the brain. In 1943 Warren S. McCulloch and Walter Pitts described a first model of a neuron in "A logical calculus of the ideas immanent in nervous activity" [26] . Their neuron model will be named MCP neuron. A biological neuron is composed of several inputs (dendrites) receiving signals from the connected neurons through a synapse. Each neuron is connected to one or several neurons through axons. In their article from 1943, McCulloch and Pitts describe the biological neuron as following 5 assumptions:

1. The activity of a neuron is an "all-or-none" process. "All or none" is the old name for a binary system where signal can be active or not active. The neuron is described as a binary output system.
2. A certain fixed number of synapses must be activated in order to activate a neuron at any time, and this number is independent of previous activity and the position on the neuron.
3. The only significant delay within the nervous system is synaptic delay.
4. The activity of any inhibitory synapse prevents activation of the neuron at that time.
5. The structure of the network does not change with time.

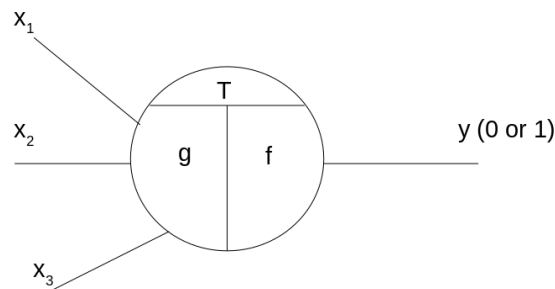


Figure 3.4.: A neuron as described by McCulloch and Pitts.

From these assumptions they created a model of a neuron that is represented in figure 3.4. The g function aggregates all the binary inputs (x_i) and the f function compares the result of the aggregation to a threshold T . If the result is superior or equal to the threshold, f outputs 1, otherwise it outputs 0. This simple model is sufficient to implement all logical functions existing in modern computers. For example a logical AND function between two inputs can be describe by the MCP neuron shown in figure 3.5. The truth table of this neuron can be found in table 3.1. The neuron is configured with:

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.3. Neural network theory

- The aggregation g is the sum of the two inputs.
- The threshold T is set to two.

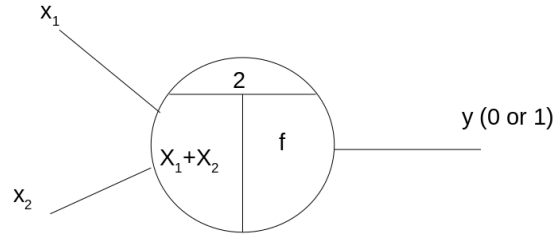


Figure 3.5.: MCP-Neuron implementing the logical AND function.

X_1	X_2	g	f
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

Table 3.1.: Truth table of the logical AND function describe as an MCP neuron with $T = 2$ and $g = X_1 + X_2$.

In 1949 Hebb improves the understanding of the biological neuron in "The Organization of Behavior" (Original publication unavailable but republished in 2002[17]). His postulate is that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that adds to the neuron stability. When an axon of a cell A is close enough to activate a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased. This postulate explains how a biological neuron can learn. Cellular changes occurs when neurons are activated.

In 1958 F. Rosenblatt improved the model of S. McCulloch and Walter Pitts by adding a learning possibility following Hebb work[35]. He created the perceptron shown in figure 3.6. Like the MCP neuron the perceptron inputs and outputs are binary. The inputs are always summed but a weight is applied to each input. An activation function outputs 1 if the sum is positive and 0 if not. F. Rosenblatt proposes a method to determine the weights. The perceptron as described by F. Rosenblatt is able to solve linear problems.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.3. Neural network theory

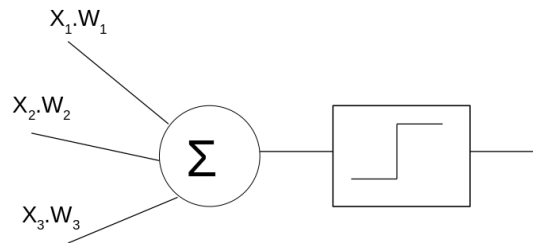


Figure 3.6.: Perceptron as described by F. Rosenblatt.

In 1986 David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams published a method that allows the evolution of the perceptron in a network structure. The network is built of several layers of neurons shown in figure 3.7. Each layer consists of several neurons, each of them connected to the output of all the neurons of the previous layer. The output of the neurons are not binary and the activation functions are not linear. This allows the network to solve non-linear problems. Figure 3.8 shows the architecture of such network. The first layer is named input layer, the number of cells in this layer corresponds to the number of inputs handled by the network. The internal layers of the network are named hidden layers. The last layer is the output layer, one cell in this layer is needed per network output. This architecture is named Feed-Forward Network as it doesn't contain any loops, the inputs of each layer correspond to the outputs of the cells in the previous layer.

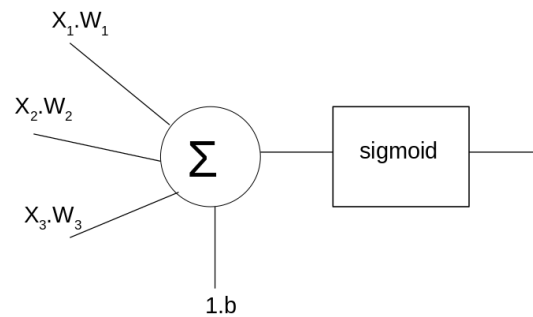


Figure 3.7.: The neuron used by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams to demonstrate the training of a network by back-propagation of errors. The 1.b input is the bias.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.4. Neural Network for LAr energy reconstruction

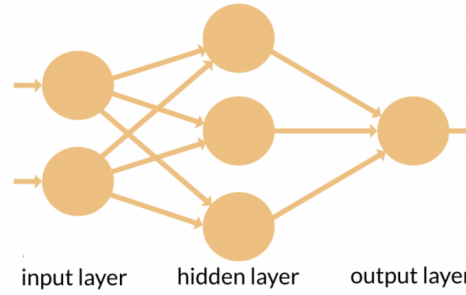


Figure 3.8.: Neural network architecture with several layers. The first layer is named input layer. The internal layers are named hidden layers. The last layer is named output layer.

Since the creation of the training algorithm, several different types of networks have been created. The increasing capacity of computing of modern computers allows to train complex networks composed of thousands of neurons.

3.4. Neural Network for LAr energy reconstruction

Several network architectures are studied to reconstruct the energies deposited in the LAr calorimeter. The following sections will focus on the recurrent neural network architecture.

3.4.1. Recurrent Neural Networks

The recurrent neural network (RNN) differs from Feed-Forward Neural Networks. The RNN cells output at a given time (named state) is taken as input of the cell in the next iteration. Figure 3.9 shows this difference in the network architecture. The size of the state vector of a cell is named state dimension. The network output at a given time depends on its past iterations. This type of networks is able to process variable length sequences of inputs. The RNNs are mainly used for tasks where the inputs are pseudo-continuous over time. At each iteration the network combines the input data with the internal state to produce a new output. Speech recognition is an example of tasks implemented in recurrent networks.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.4. Neural Network for LAr energy reconstruction

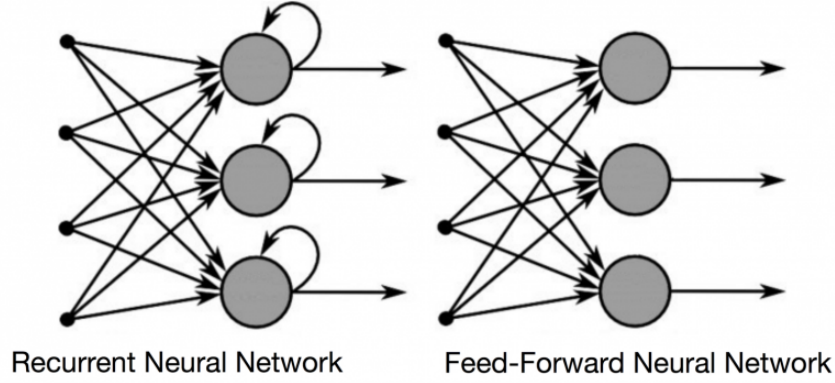


Figure 3.9.: Difference in architecture between a RNN and a Feed-Forward NN. The RNN cell output is taken as input of the same cell at the next iteration.

Different architectures of recurrent networks exist. The different architectures are shown in figure 3.10. The choice of an architecture for a network depends on the number of inputs/outputs of the network. The one to one and many to one architectures are considered in the rest of this chapter.

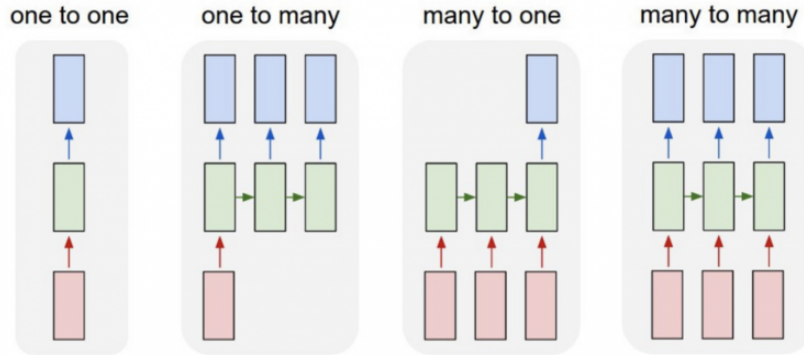


Figure 3.10.: Schematic of the different architectures possible for an RNN. The red blocks represent the input layer cells. The green blocks represent the hidden layer cells. The blue blocks represent the output layer cells. The one to one architecture maps one network input to one network output. The one to many architecture maps one network input to several outputs. The many to many architecture maps several network input to several network outputs. The many to one architecture maps several inputs to one output.

For the LAr energy reconstruction, the recurrent networks are a natural candidate. The ADC values are digitized every clock cycle from a continuous signal. The network

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.4. Neural Network for LAr energy reconstruction

computes the energy deposited in a calorimeter cell from the ADC values corresponding to this cell. A one to one architecture is thus a natural choice.

The network is trained by an error back propagation algorithm. This training uses data from the simulation of the detector. The detector simulation software used is named AREUS. The simulated response of the detector for a given energy deposit is used for training. The reconstructed output by the network is compared to the expected deposited energy. The deviation between the deposited and computed energies is back propagated to the network and the network weights are modified accordingly.

Two types of cells are studied for the LAr energy reconstruction: the vanilla cell[40] and the Long Short-Term Memory(LSTM)[18] cell. Figure 3.11 shows the architecture of these two types of cells. The cells are composed of gates. Each gates is composed of a weight multiplication, a bias addition and an activation function. The vanilla cell is simple and is composed of only one gate, one input and one state vector. The LSTM is composed of five gates, one of them is composed only of an activation function. This gate is named the forgot gate. Furthermore the LSTM cell have an additional state vector named hidden state.

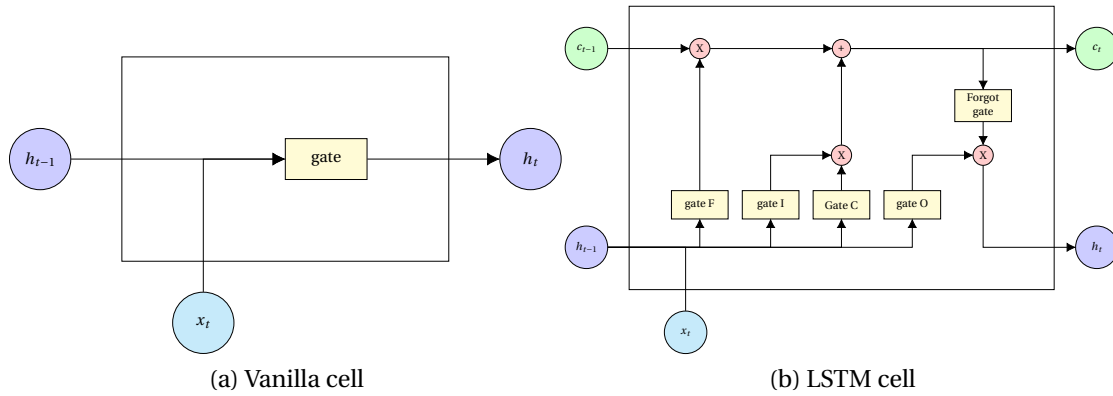


Figure 3.11.: Architectures of recursive cells. The cell computes the output state h_t (and c_t for the LSTM) from the previous iteration (h_{t-1} and c_{t-1}). Each gate is composed of a weight multiplications, a bias addition, and an activation function except the forgot gate of the LSTM cell which is composed of only an activation function.

The FPGA resources being limited the network design must reduce as much as possible the number and complexity of the required mathematical operations. The smallest network in a one to one architecture containing only one cell. This architecture is named single cell network. The output layer is composed of a perceptron cell which aggregates the components of the state vector. This output layer is named dense layer. Figure 3.12 shows the schematic of this architecture. In such design the ADC data are

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.4. Neural Network for LAr energy reconstruction

handled as a continuous set of data. The network retains data from the beginning of the dataset. A training was done for such architecture for both the vanilla and the LSTM networks. For the vanilla cell the training did not converge properly. The vanilla network doesn't possess the required computation complexity to be stable over long sequences. For the LSTM cell the training was successful. Figure 3.3b shows the energy resolution of such network. The trained LSTM network gives better reconstruction performance compared to the OF algorithm particularly in the low gap regions.

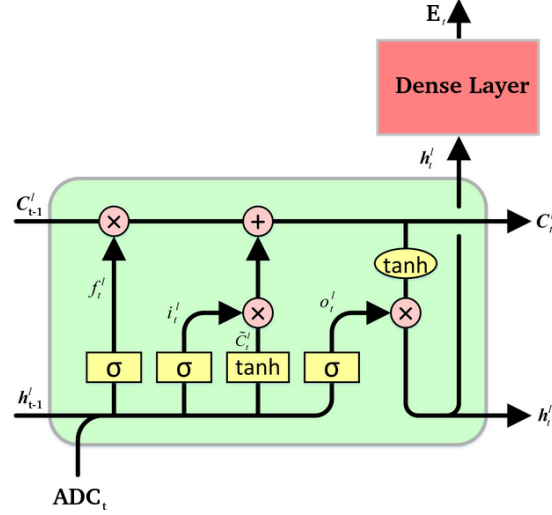


Figure 3.12.: Schematic of a single cell network for an LSTM cell. At each iteration the energy E at the time t is computed from the cell state(C), hidden state(h) and the ADC input. These two states are the output of the previous iterations. The input is the ADC value corresponding to the timestamp of the current iteration (t).

The single LSTM cell architecture firmware implementation cannot fulfill the latency requirement as explained in section 3.5.1. A different architecture called sliding windows was developed to mitigate the latency issue. Figure 3.13 shows the schematic of this new network architecture. It is trained in the many-to-one RNN architecture in keras[21] targeting an energy prediction with five consecutive ADC samples as input. The training of such network is possible for both the Vanilla and LSTM network due to the short sequence length of 5. Figure 3.3c and 3.3d shows the resolution of such sliding window architecture. The energy resolutions of the two sliding windows are similar in the low gap region. Both of them outperform the OFmax algorithm in this region. However one can notice that the single cell architecture slightly outperforms the sliding window architecture for the LSTM cells. This is due to the longer sequence which profits the network from additional information about past events.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.4.
Neural Network for LAr energy reconstruction

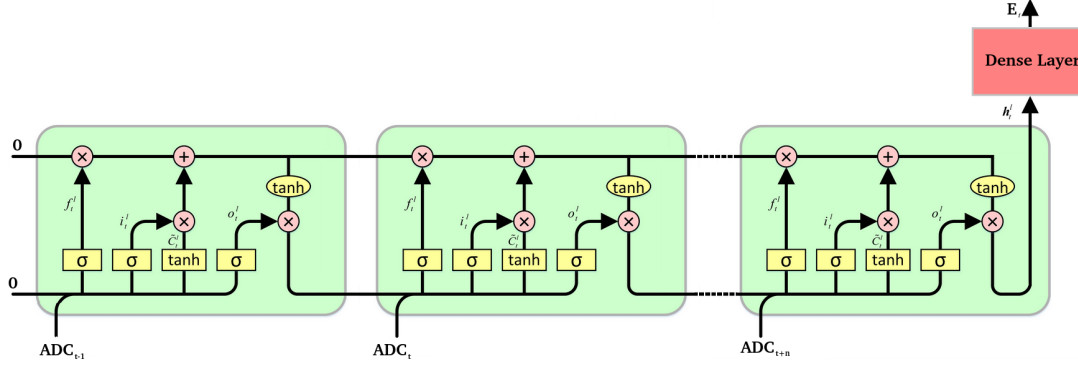


Figure 3.13.: Schematic of a sliding windows network for an LSTM cell. Each cell is taking as input one timestamp of the data and the output state of the cell computing the previous timestamp.

Figure 3.14 compares the performance of the three types of RNN algorithms described before with the performance of the OF with max finder algorithm. All the RNN algorithms outperform the OF algorithm. The best performance is given by the LSTM single cell algorithm. The performance of the two sliding window algorithms are similar.

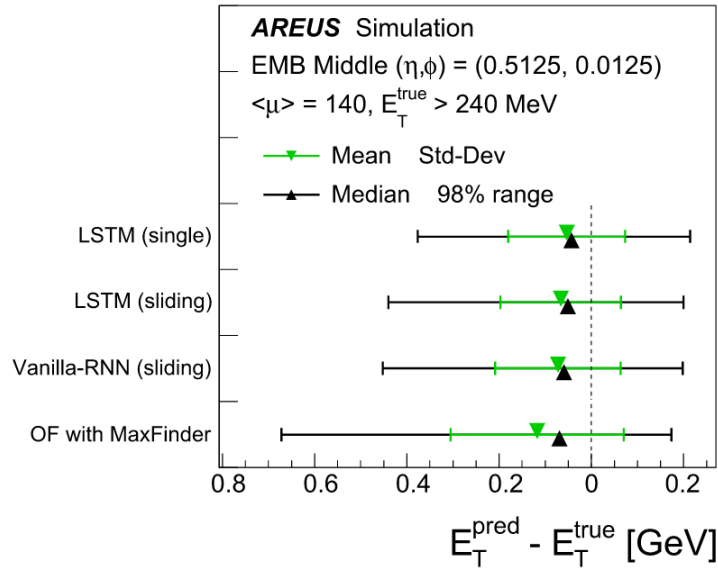


Figure 3.14.: Transverse energy resolution for the optimal filtering and the various RNN algorithms. The performance is assessed by comparing the true transverse energy deposited in an EMB middle LAr cell ($\eta=0.5125$ and $\Theta=0.0125$) to the RNN prediction after simulating the sampled pulse with AREUS assuming $\langle \mu \rangle=140$. Only energies 3σ above the noise threshold are considered. The mean, the median, the standard deviation, and the smallest range that contains 98% of the events are shown.

3.5. Neural network firmware considerations

The mathematical operations of the NNs and the corresponding implementation in hardware logic are described in this section.

3.5.1. Internal neural network operations

Figure 3.15 shows the schematic of the internal structure of a gate. The gates are composed of four computation blocks:

- Input weight multiplication: This block multiplies the input value with the input weights. The output of this block is a vector of the size of the state of the cell.
- Bias addition: this block does the addition between the output of the input weight multiplication block and the bias vector.
- Recurrent weight multiplication: This block does a matrix multiplication between the state of the cell and the weight matrix.
- Summing block : this block adds the output of the bias addition and the output of the recurrent weight matrix multiplication
- Activation function : Apply the activation function on each of the vector members of the output of the summing block.

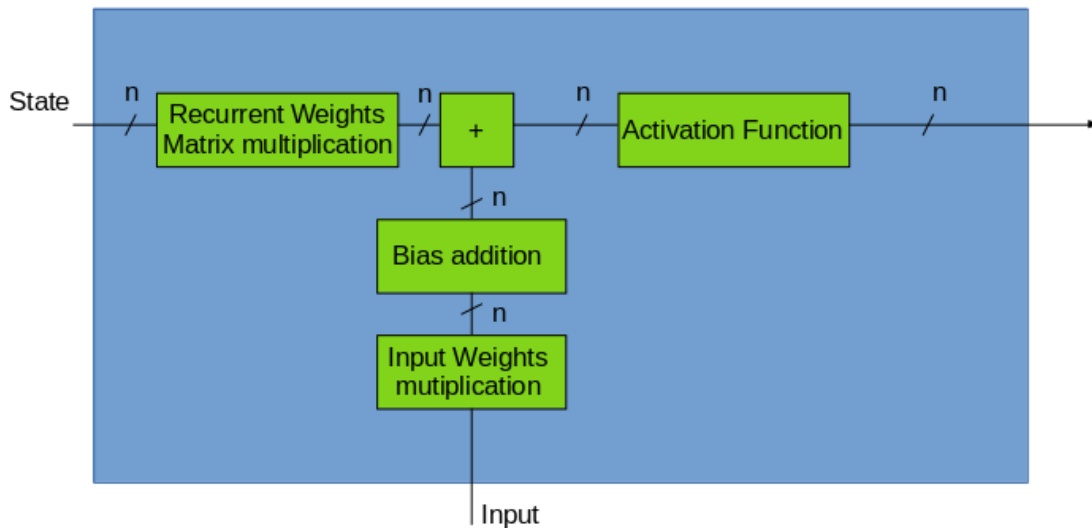


Figure 3.15.: Schematic of the computation blocks inside an RNN gate. n is the size of the state vector.

3.5.2. Multiplexing and pipeline

The computing resources of the FPGA are limited in particular the multiplication blocks contained inside digital signal processing (DSP) blocks. The key principle to reduce the number of used DSP blocks is to use each one of them at its maximum capacity. The DSPs are composed of a pipeline with an initiation interval of one. A pipeline is a succession of data processing elements connected in series. The output of one element is the input of the next one. The initiation interval is the number of clock cycles needed by the pipeline between two inputs. With an initiation interval of 1, at each clock cycle the multiplication block accepts a new input and produces an output. To fully use the pipeline capabilities of the DSP the network design must be fully pipelined with an initiation interval of 1.

The frequency of the input to the neural network is 40 MHz. It corresponds to the frequency of the digitization of the signal. This frequency is much lower than the maximum frequency of the DSP block which is around 600 MHz in Stratix 10 FPGAs. To use the DSP at their maximum frequency different channels are multiplexed into the same network implementation. For example a network implementation multiplexed 4 times runs at 160 MHz and handles the computations for 4 channels.

3.5.3. Latency

The latency is the time for the result of a computation to be available. It can be expressed in number of clock cycles. The latency of a sequence of blocks is equal to the sum of the latency of its blocks. If there is two independent inputs the latency is equal to the largest latency between one input and the output. In an RNN gate the block with the largest latency is the recurrent weight multiplication. Its computation is longer than the input weight multiplication and bias addition. The full path constraining the latency is composed of this matrix multiplication, an addition and an activation function. Table 3.2 shows the latency of the computation blocks of the RNN. From this table we can estimate the minimum latency of the computation of a gate. In the firmware implementation the latency may be larger, several register are added by the compilation software to achieves high frequency throughput. The minimum gate latency varies between 12 or 15 clock cycles depending of the activation function. The different activation functions are detailed in section 3.6.4. The minimum latency of a vanilla cell is between 12 and 15 clock cycles. For the LSTM cell the slowest path is composed of a gate, two multiplications, one addition and an activation function outside the gate. The LSTM minimum latency is estimated to 32 clock cycles.

To be used in the single cell mode, the LSTM cell latency must be lower than the time between two inputs. The state at a given timestamp must be computed before the next timestamp. The maximum latency of the LSTM cell must thus be smaller than 25 ns. However the minimum latency of 32 clock cycle corresponds to 53 ns for a 600 MHz clock. The minimum latency of the LSTM is larger than the maximum

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.5. Neural network firmware considerations

Computation	Latency in clock cycles
Multiplication	6
Matrix Multiplication	10
Addition	1
Relu activation function	1
LUT based activation function	4

Table 3.2.: Minimum latency of the basic computation block used in the RNNs. This latency can be increased by the HLS compiler to achieve high frequency.

latency of the single cell implementation. Therefore a single LSTM cell architecture is impossible for the LAr energy computation at 40 MHz with Stratix 10 FPGA.

3.5.4. Multiplications

The Stratix 10 FPGA has a maximum of 11520 multiplier blocks. This is one of the main limiting resources. The only operations implemented in this type of block are the multiplications so its usage can be easily evaluate by counting the number of needed multiplications. In the case of the recurrent network in sliding windows architecture the main design parameters are the number of NN cells and the hidden dimensions. The networks considered here for the LAr energy reconstruction is a 5 timestamp sliding windows network and thus possess 5 cells. The resources usage will only depend on the hidden dimension.

Figure 3.16 and 3.17 show the estimated number of LAr channels that can be handled by one FPGA for both the LSTM and vanilla networks. The number of channels needed depends on the number of FEB2 that the FPGA will handle. This number is still under discussion and 3 hypotheses exist with 2, 3, or 4 FEBs per LASP FPGA. The implementation of the vanilla network with eight hidden dimensions fits the LASP requirement with 4 FEBs or less while the LSTM implementation with 10 hidden dimension does not fit the requirements even for 2 FEBs. To have enough resources in all FEB/LASP configuration a multiplexing of fourteen is needed at minimum for the vanilla network. This network targeting a fifteen multiplexing and 600 MHz clock will be used as reference for the following sections.

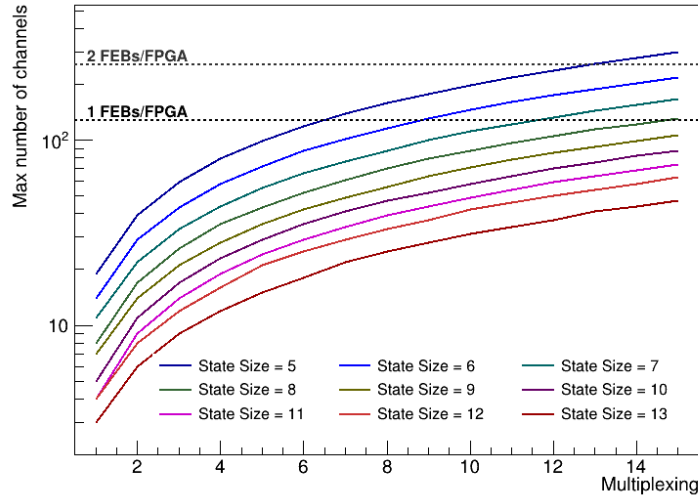


Figure 3.16.: Number of channels fitting into Stratix 10 SX multiplying blocks for a 5 cell LSTM network in sliding windows algorithm as a function of the multiplexing.

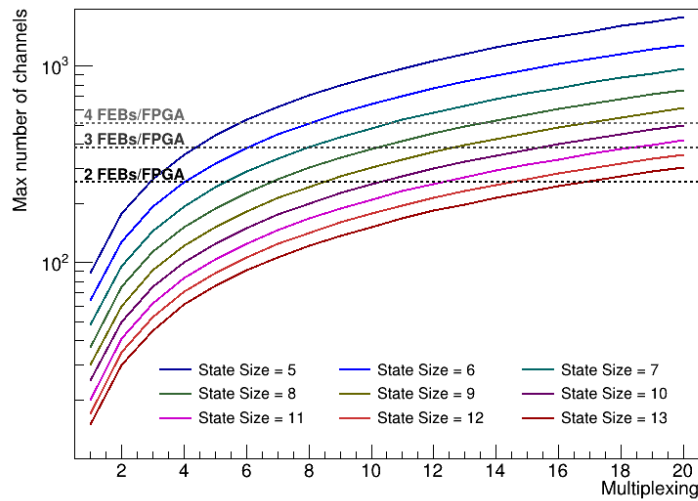


Figure 3.17.: Number of channels fitting into Stratix 10 SX multiplying blocks for a 5 cell Vanilla-RNN network in sliding windows algorithm as function of the multiplexing.

3.6. Firmware Implementation

The RNN for LAr energy reconstruction has been implemented in Intel High level synthesis. This implementation allows more flexibility than a classic Hardware De-

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.6. Firmware Implementation

scription Language (HDL) implementation. Each part of the code has been optimised to fit within the resources of the FPGA.

3.6.1. Intel FPGA blocks

An FPGA component is composed of an array of interconnected blocks. An FPGA contains configurable blocks to perform complex digital operations and blocks dedicated to a given functionality. The main digital operation blocks are arithmetic and logic units, flip flops (FF), memory, and digital signal processing (DSP) blocks. Dedicated blocks can be high speed transceivers, PCIe interface or ARM processors. Figure 3.18 shows the different blocks composing the Stratix 10 FPGA.

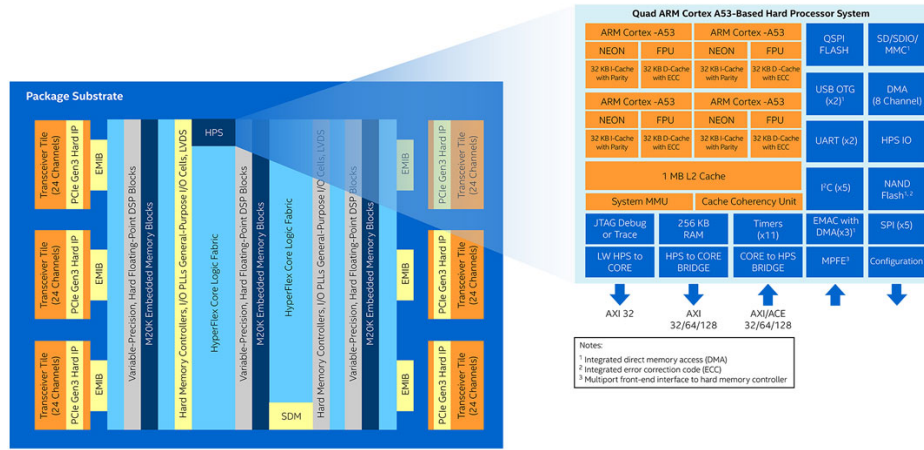


Figure 3.18.: Block design of the Stratix 10 SX FPGA[37].

The DSP block is optimized for mathematical operations during signal processing, in particular multiplications. Each DSP of the Stratix 10 contains two multiplier of size 18x19 bits. The DSP can be configured to do fixed point or floating point operations. Configuration with a 27x27 bits multiplier is also possible. Figure 3.19 shows the schematic of the DSP in two 18x19 multipliers for fixed point operations. DSPs contain the multipliers but also several addition blocks. One pre-adder is attached to one input of each multiplier block. This pre-adder adds two inputs of the DSP. The outputs of the two multiplier blocks can be summed. A special adder named accumulator can also be used to sum the outputs of several DSPs. In such mode the DSPs are chained with their chain in/out interfaces. The 18x19 multiplier mode corresponds to the one used in this work. The FEB2 ADC output is an integer with less than 19 bits. The neural network weights are chosen to be fixed points with less than 18 bits. This choice is

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

made to fit into this operation mode of the DSP, allowing two multiplier blocks per DSP.

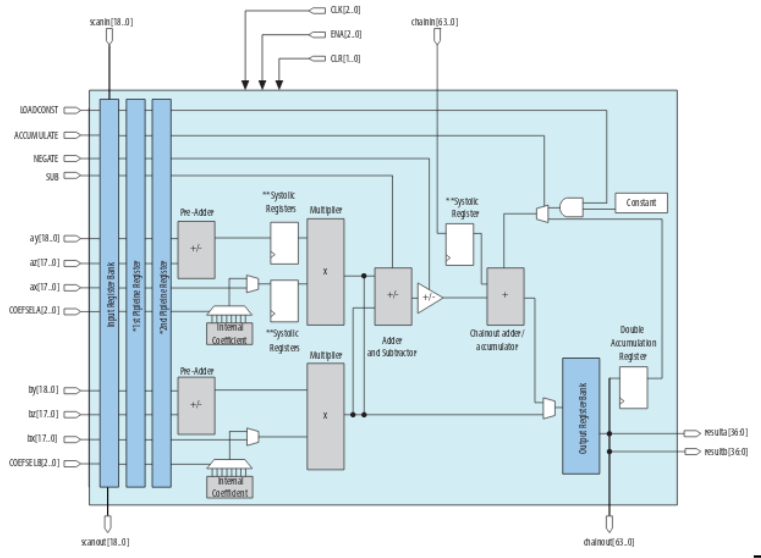


Figure 3.19.: Schematic of a Stratix 10 DSP block. The block is configured in the two 18x19 multiplier mode.

The Adaptive Logic Module (ALM) is composed of two Adaptive Look-Up Tables (ALUT). Figure 3.20 shows the ALM schematic. Each of the lookup tables is connected to a full adder block and one register. With its lookup tables and adder, this block is able to implement every logic gate. The LUT can also be used to implement a memory.

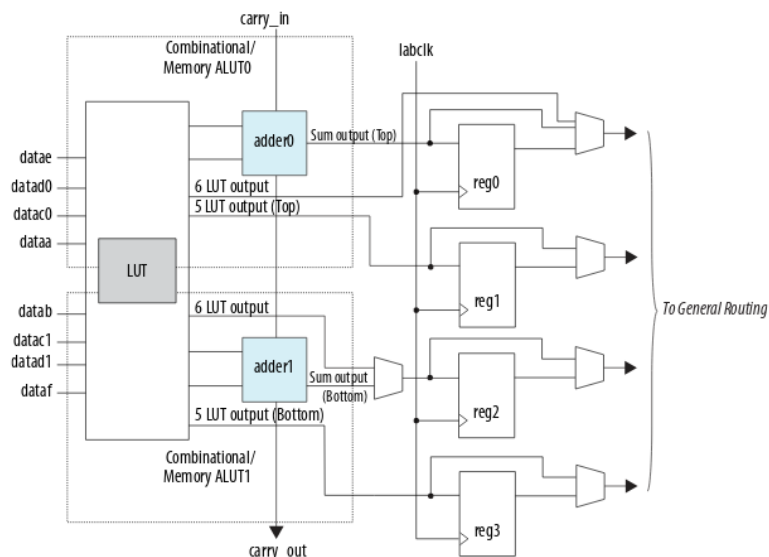


Figure 3.20.: Schematic of an ALM of the Stratix 10 FPGA.

3.6.2. The Intel High Level Synthesis

The Intel High Level Synthesis (HLS) compiler is a tool that generates an RTL code optimised for FPGA from a C++ code. It raises the abstraction level for FPGA hardware design. The HLS is chosen for the RNN implementation since it allows additional flexibility compared to the low level Hardware Description Language (HDL) usually used in firmware design. In the HLS C++ code several macros are defined to modify the way the RTL code is generated.

Figure 3.21 shows the design flow for components implemented in HLS or HDL. The HLS allows to verify and refine the algorithm without the RTL simulation software by performing a standard C++ compilation. During the simulation of an HLS design the Intel HLS compiler estimates the resources used by the design. It allows to optimize the FPGA performance and resources without a full compilation as in the HDL flow.

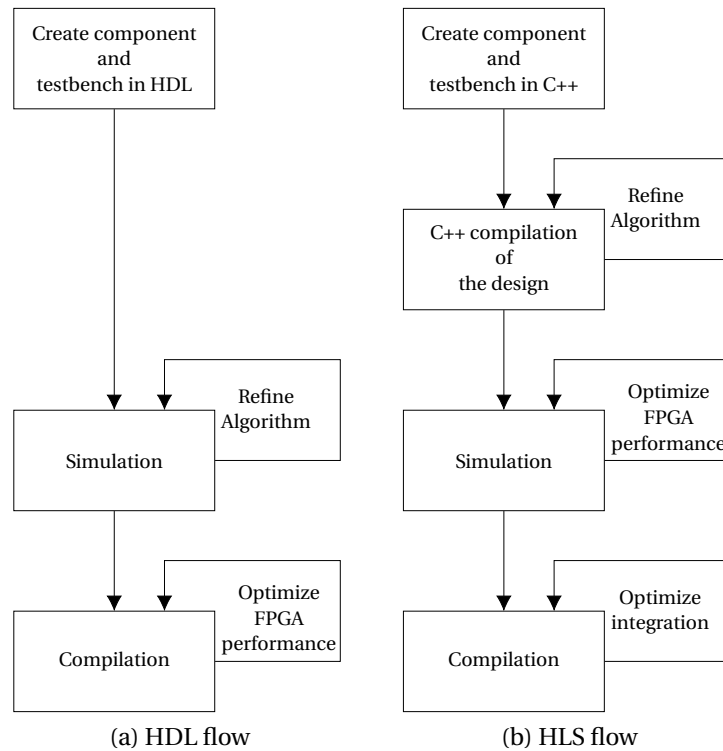


Figure 3.21.: Design flow for an HLS component and an HDL component.

The components produced by the HLS can have different types of interfaces. Only the streaming interface will be presented since it is the only one compatible with the high throughput needed for the LAr energy reconstruction. Figure 3.22 shows the schematic of the data interfaces of such components. The different signals are:

- Data inputs: a and b signals in the schematic.

- Start signal: When this signal is set the component starts the computation from the input data if it is not busy.
- Busy signal: Indicates that the component is already performing a computation and can't accept a new input.
- Done signal: Indicates that a computation is finished.
- Return data: Value computed by the component.
- Stall signals: indicates that the subsequent components to which the output value is being sent is busy and can't accept new inputs. The output value of the HLS component is stored in one FIFO waiting the subsequent component availability. In the LAr system this functionality is not used as it creates variable latency computation. The stall signal and the FIFO are removed by a macro.



Figure 3.22.: Schematic of the data interfaces of an HLS streaming component.

3.6.3. Implementation of Matrix Multiplication

Equation 3.1 depicts a matrix-vector multiplication. The operation is a dot product between the corresponding line in the matrix and the vector. This operation is composed of several multiplications followed by the accumulation of their results. The computation of each the components of the results vector is independent. Optimizing the matrix-vector multiplication corresponds to optimizing the dot product between two vectors.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n \end{bmatrix} \quad (3.1)$$

Three different implementations of the dot product are compared. The dot product will be written dot-X with X the size of the vectors. In this section the studied operation is a dot-8. An HLS component is defined including only the operation describe in

equation 3.2.

$$A.B = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_8 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_8 \end{bmatrix} = \sum_{i=0}^7 a_i.b_i \quad (3.2)$$

The A and B vectors are represented in the code as arrays. The elements of the A array are 18 bits fixed point and the elements of the B array are 19 bits fixed point. The number of bits is chosen to correspond to the one of the Stratix 10 multiplier. The three different implementations are:

- Consider the standard C++ style multiplication accumulation; the accumulator(acc) is a 37 bit fixed point variable :

```
for (int i=0; i < 8; i++){
    acc += a[i] * b[i];
}
```

- Consider the HLS style following the logic of the chained DSP. This implementation corresponds to the DSP internal logic. The goal of this implementation is to use the DSP with the adder between the two multiplications and also the accumulator with chained DSP. Several intermediate 37 bit variables named tmp are created to sum the output of two multiplications. This tmp variable has the size of the output of the DSP adder. The accumulator sums all these tmp variables. The accumulator is also of size 37 bits. This design will be called ACC37.

```
for (int i=0; i < 4; i++){
    tmp[i] = a[i]*b[i] + a[7-i]*b[7-i];
}
for (int i=0; i < 4; i++){
    acc += tmp[i];
}
```

- Consider the HLS style following the logic of the DSP without chain design. This implementation follows the same logic as the ACC37 design except that a level of register is added at the output of the adder between the two multiplication results. There is no register in the DSP so the HLS compiler is forced to output the multiplication results into logic blocks to implement the register. With this implementation the accumulation with chain DSP can't be used and the accumulator is implemented in ALM logic blocks. Furthermore in this design the accumulator size is reduced to 19 bits. This design will be called ACC19.

Implementation	ALUTs	FF	DSP
C++ style	709	222	8
ACC37	116	79	4
ACC19	137	78	4

Table 3.3.: Resource usage for three different implementations of a vector multiplication and accumulation. The implementation is compiled with the quartus 21.2 HLS compiler at a target frequency of 100 MHz.

```

for (int i=0; i < 4; i++){
    tmp[i] = hls_fpga_reg(a[i]*b[i] + a[7-i]*b[7-i]);
}
for (int i=0; i < 4; i++){
    acc += tmp[i];
}

```

The resource usage for the 3 implementations are summarized in table 3.3. The implementation without 37 bit intermediate variables uses much more resources including DSPs. In the C++ style design the HLS compiler implements a 37x18 bit multiplier. The multiplication outputs are 18 and 19 bits so the 37x19 bit multiplier is a waste of DSP. The C++ style design is so discarded.

Figure 3.23 shows the schematic of the ACC19 and ACC37 implementations. In the ACC37 a dot-8 operation is implemented. It corresponds, as expected, to a chained DSP implementation. In the ACC19 four dot-2 operations are implemented and their output summed. The output sum is done in ALUT explaining the larger ALUT consumption.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

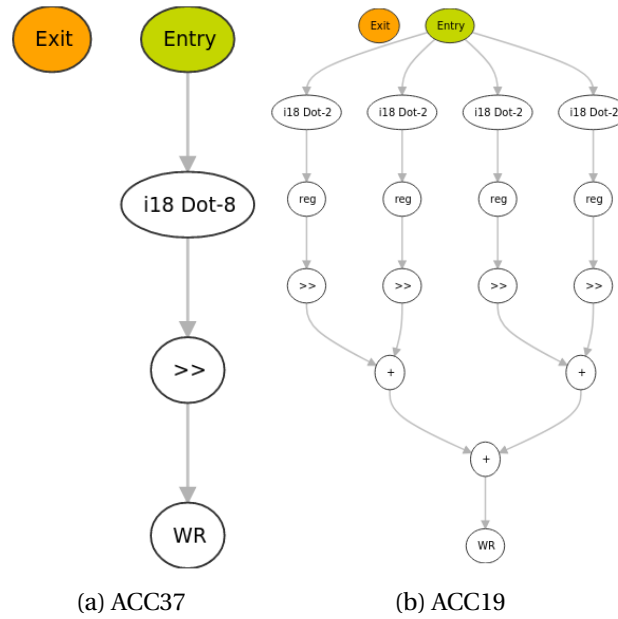


Figure 3.23.: Operations implemented by the HLS compiler for two designs of a dot-8 operation. (a) is the operations for the ACC37. (b) is the operations for the ACC19. >> represent a bit-shift reducing the size of the value representation.

To run at high frequency the HLS compiler uses additional FFs and ALUTs. More resources are added for the ACC37 design than for the ACC19 as the target frequency of the design increases. As detailed in table 3.4 the maximum frequency of the DSP is higher if the chain in/out is not used. Furthermore this maximum frequency depends of the speed grade of the FPGA. Figure 3.24 shows the resource usage of the dot-8 product depending on the frequency. At low target frequency the ACC37 implementation uses less resources than the ACC19 implementation. At target frequency higher than 400 MHz the ACC19 implementation uses less resources than the ACC37 implementation. The choice between the two design will be done depending of the target frequency of the component.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

Mode	Speed grade 1 frequency (MHz)	Speed grade 2 frequency (MHz)
Fixed-point 18 × 18 multiplier adder summed with 36-bit input mode	1000	771
Fixed-point 18 × 18 multiplier adder summed with 36-bit input mode and chain in/out	750	578

Table 3.4.: Core maximum frequency of the Stratix 10 DSP with and without the chain in/out for Stratix 10 FPGAs with two different speed grades.

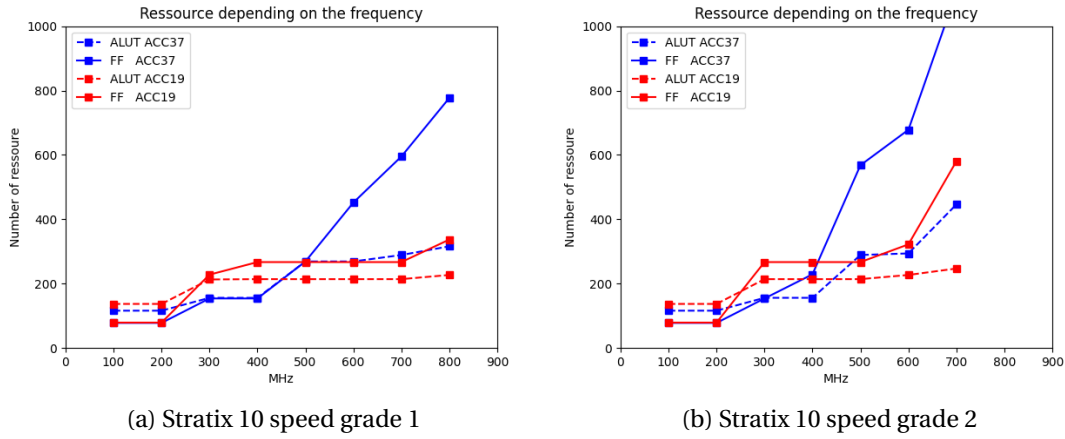


Figure 3.24.: Resource usage of the ACC19 and the ACC37 designs depending on the target frequency. The DSP usage does not change depending on the target frequency.

3.6.3.1. Design with power of two weights

To reduce the usage of DSP, the NN is trained with its weights constrained to be a power of two. With such weights the multiplication can be simplified to a bit shift. In this section the implementation and resource usage is studied for an HLS component performing only one multiplication.

In Intel HLS the bit-shift C++ definition is different from the standard C++. In standard C++ a bit-shift can be a right shift or a left shift with always a positive number of bits shifted. In Intel HLS the shift operand can be positive or negative, the shift direction depending on the sign.

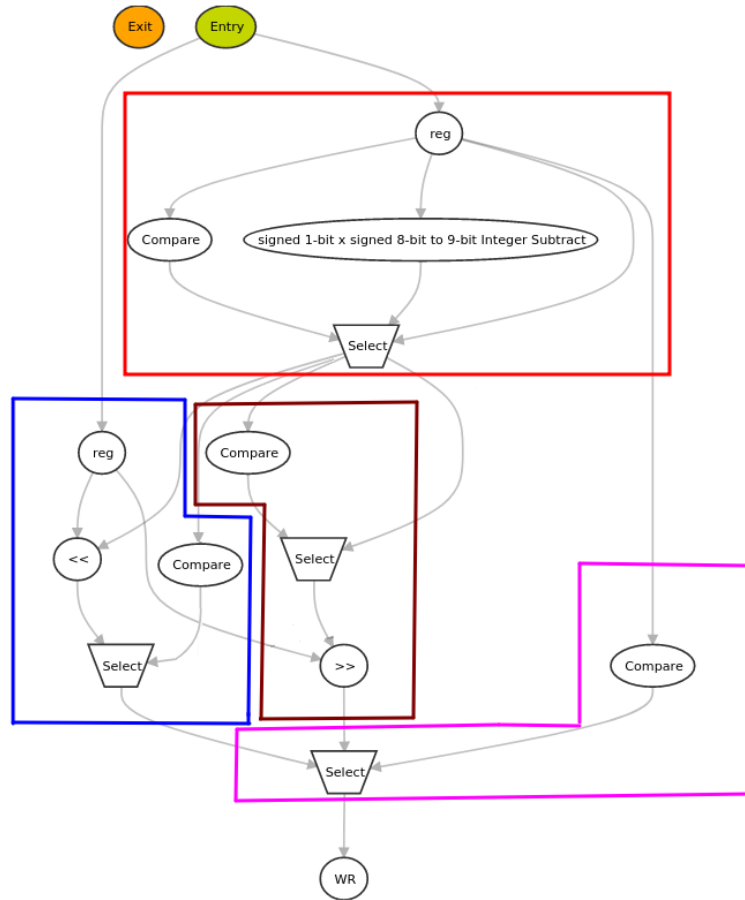


Figure 3.25.: Schematic block of a bit shift implementation. The brown block is a right shift. The blue block is a left shift. The pink part selects the shift direction. The red part inverts the sign of its inputs.

Figure 3.25 shows the block schematic of the implementation of a bit shift in HLS. The shift direction being unknown, both left shift (brown part) and left shift (blue part) are implemented. The result of the desired shift is selected depending of the sign of the operand (pink part). The bit shift register logic, right or left, is always performed with a positive value for the number of shifted registers. An absolute value computation is thus done in the red part.

If the direction of the shift is fixed, the implementation can be simplified with only one direction implemented. In HLS it is indicated to the compiler with an unsigned type for the operand. Figure 3.26 shows the block schematic of this implementation. On this design one shift direction is implemented. The compare and select blocs handle the case where the shift value is equal to 0. As shown in table 3.5 such implementation uses less resources than the one with an integer value.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

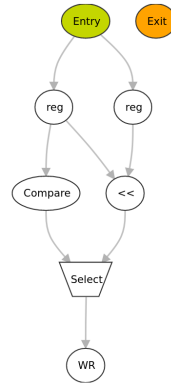


Figure 3.26.: Schematic block of a bit shift implementation with its direction fixed.

If the operand value is fixed the implementation doesn't have any branching as shown in figure 3.27. Only a shift is implemented. This shift is just a reordering of the bits in the data bus so no logic is needed. The ALUT usage for this operation that is shown in table 3.5 handles the input and output logic of the component.

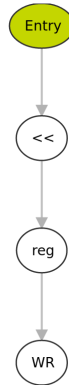


Figure 3.27.: Schematic block of a bit shift implementation with its direction fixed and a constant operand value known at compilation time.

Operation	bit-shift operand type	ALUT	FIFO	DSP
Bit shift	integer value	172	66	0
	unsigned integer value	100	84	0
	Fixed integer value	5	0	0
Multiplication		19	0	1

Table 3.5.: Resources usage of the implementation of a bit shift depending on the operand types. The resources usage of the multiplication is given for comparison.

Using power of two weights allows to implement the multiplication using only ALUT and no DSP. However the LAr network needs to have configurable weights to

be multiplexed. Therefore the value of the weights are unknown and can be positive or negative. The only possible implementation for the multiplexing of these weights will be the one with the integer bit-shift operand type. In this implementation the ALUT usage will increase significantly and thus NN with power of two weights are not considered further.

3.6.4. Activation functions

Several activation functions are used in neural networks. The activation functions must not be linear to be able to solve non-linear problems. For the LAr energy reconstruction three activation functions were studied:

- Hyperbolic tangent : For two gates in the LSTM.
- Sigmoid : For three gates in the LSTM.
- Rectified Linear Unit(Relu): For the gate in the Vanilla-RNN.

3.6.4.1. Implementation of the computation of a RELU

The Relu function is defined by equation 3.3. Its implementation on FPGA is relatively simple. The fixed point representation used in the LAr network possesses a sign bit. A test on this sign bit is enough to implement the relu function. Figure 3.28 shows the RTL view of this implementation.

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{else} \end{cases} \quad (3.3)$$

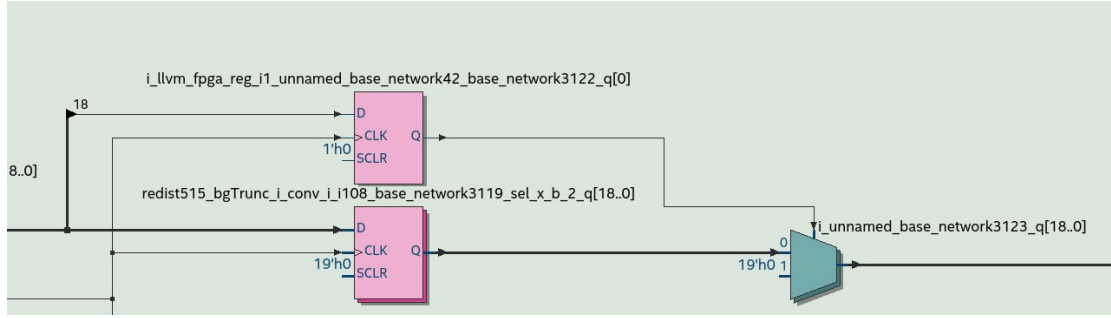


Figure 3.28.: RTL view of the Relu activation function extracted from the RTL viewer of quartus. Pink rectangles are registers. The blue trapezoid is a multiplexer. The top register extracts the bit at position 0 of the data. This bit is the sign bit equal to 0 if positive and 1 if negative. Depending of this value the multiplexer selects the input 0 or input 1. The multiplexer propagates the selected input value to its output. The input 0 of the multiplexer is connected to the register containing the data. The input 0 is selected in case of sign bit at 0 (positive value). The input 1 is connected to a null value and selected when the sign bit is 1(negative value).

3.6.4.2. Implementation of the sigmoid function

In this section the implementation of the sigmoid function will be presented. The same implementation principle applies to others complex activation functions such as the tanh function.

The sigmoid function is defined by equation 3.4. It contains a division and an exponential. These two complex operations cannot be implemented in FPGAs without a large resource usage.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

A look up table is used to approximate the sigmoid function. As represented in figure 3.29 the value of the sigmoid is calculated for a fixed number of evenly distributed value. The interval between two computed values is named a bin. The calculated values for each bin are stored in a table in the FPGA. The sigmoid computation is thus replaced by an index computation and a look up in the table.

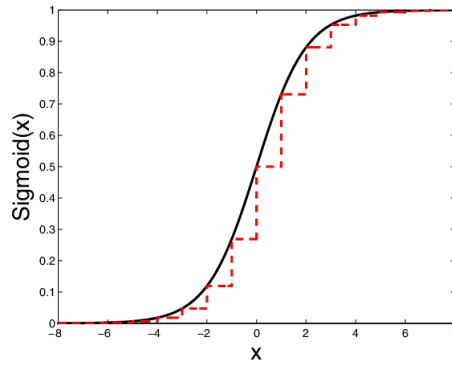


Figure 3.29.: Approximation of a sigmoid function using a look up table. The solid line is the sigmoid function. The dashed line is the LUT approximation.

The resolution of the implementation of the LUT is shown in figure 3.30. The sigmoid function is symmetric around $(0,0.5)$: $f(-x) = 1 - f(x)$. By using this symmetry it's possible to reduce the size of the LUT for the same precision. Only the value of the positive part of the function is stored in the LUT. The value of the negative part is computed from it. By keeping the same LUT size the resolution is improved by a factor of two. The resolution of the function using a symmetric LUT is shown in figure 3.31. On this figure two different computation of the bin value are shown: the first one is using the value of the lower limit of the bin, the second at the value in the middle of the bin. Table 3.6 shows the comparison between the implementation of a 5 cell LSTM sliding windows for two different implementations of the LUTs. One can see that the resolution is comparable between the one implementing 4096 bins LUTs without any optimisation and the one implementing 2048 bins LUTs with value computed in the central value of the bin. The 2048 bins LUT reduce the resource usage, in particular the usage of RAMs and ALUTs.

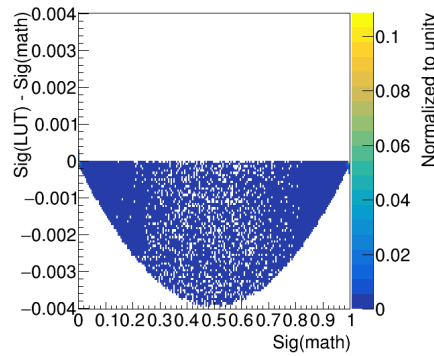
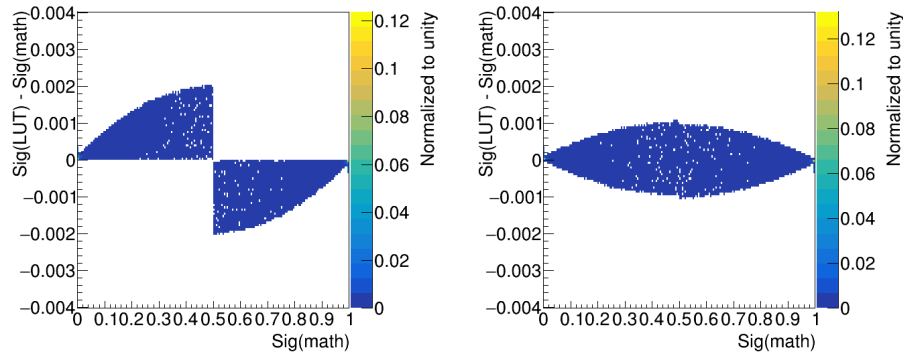


Figure 3.30.: Resolution of the Look up table of the sigmoid function. The lookup table contain 1024 entries evenly distributed between -8 and 8. The value of a bin is computed from the value of the lower limit of the bin.

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.6. Firmware Implementation

	4096 bins lower value	2048 bins central value
Absolute Resolution Mean	-0.0059	-0.0012
RMS	+/- 0.0092	+/- 0.0112
Relative Resolution Mean	-0.0064	-0.0040
RMS	+/- 0.016	+/- 0.015
ALUTs	160000(9%)	152809 (8.1%)
FFs	258576(7%)	249993 (6.69%)
RAMs	966(8%)	486 (4%)
DSPs	845(15%)	844 (15%)

Table 3.6.: Resolution and resource usage for two different LUT implementations of a 5 cell sliding window LSTM network. The resolution is the difference between the output of the LUT based implemented network with the floating point network implementing real function. The first implementation uses LUTs with 4096 bins where the value of the bin is computed from the lower value of the bin. The second implementation uses 2048 bins LUTs with their values computed from the central value of the bin.



(a) Value of the bin computed from the lower limit
(b) Value of the bin computed from the value in the center of the bin

Figure 3.31.: Resolution of the Look up table of the sigmoid function. The lookup table contain 1024 entries evenly distributed between 0 and 8. The negative values are computed using the symmetry of the function.

3.6.5. Fixed point representation

Inside the FPGA numerical values can be represented by a fixed point or a floating point implementation. Each DSP of the Stratix 10 supports 2 fixed point multiplications and only one floating point multiplication. Thus fixed point representation allows to reduce the DSP resource usage.

The fixed point representation implemented in quartus HLS follows the Algorithm-

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6.
Firmware Implementation

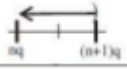
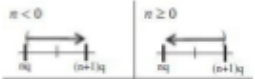
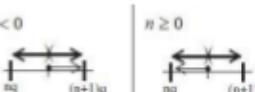
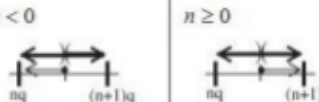
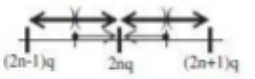
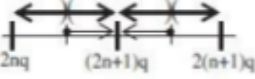
Modes	Behavior n is integer, q is 2^{I-W}	Simulation/Synthesis cost
AC_TRN (default) (trunc towards $-\infty$)		Delete bits, no cost except for /= (div assign) signed
AC_TRN_ZERO (trunc towards 0)		No cost for /=, or unsigned src For signed: incrementer, OR for deleted bits, AND with sign bit
AC_RND (round towards $+\infty$)		Various forms differ only on the direction of the rounding for values that are exactly at half point. All require an incrementer, some require to OR deleted bits, some only require to look at the MSB of the deleted bits.
AC_RND_ZERO (round towards 0)		
AC_RND_INF (rounds towards $\pm\infty$)		
AC_RND_MIN_INF (round towards $-\infty$)		
AC_RND_CONV (round towards even q multiples)		
AC_RND_CONV_ODD (round towards odd q multiples)		

Table 3.7.: Quantization modes for a fixed point[7]. W is the total width and I the numbers of bits of the integer part.

mic C (AC) Data-types defined by Mentor Graphics [7]. Four parameters define this representation: the total number of bits, the number of bits of the integer part, the quantization, and the overflow.

The quantization defines how the bits to the right of the Less Significant Bit (LSB) are lost. The loss of bits occurs during the conversion of the floating points representing inputs and weights. The inputs and weights are given by a simulation of the detector and by the training software. These two software use 32 bit signed floating point representations. The bit loss also occurs in the neural network cells after the multiplication. The 37 bits output of the DSP is reduced to the number of bits internally used. Two types of quantization are implemented: truncation and rounding. Each of these types have a different implementation defined in table 3.7. By default the AC_TRN mode is used.

The overflow defines how the bits to the left to the Most Significant Bit(MSB) are lost. This loss of bits occurs when the results of the multiplication or addition is outside the range of the represented numbers. Four types are defined: a drop of bits and three different saturation. The saturation implementation uses logic. As it is needed after all operations it represents a large amount of signals. Furthermore a lot of logic is needed to propagate these signals and handle them. As the value of the weights and the maximum value of the ADC input are known, the maximum value after operations can be determined for each cell. The number of bits and the position of the radix point is chosen to be able to represent this maximum value. No saturation detection is thus needed. The drop of bits is thus used in the LAr neural network.

In this section the performance of a 5 timestamp sliding windows vanilla network will be studied. Figure 3.32 shows the schematic of the vanilla cell used. Three data types are defined. Each of these types can have a different width and a different mode of quantization. The three types are:

- Weight type: All the NN weights are defined in this type. This maximum width of weights is set to 18 bits since the weight must fit into 18x19 bits multiplier. 18 bits are enough to have a good resolution of the network output.
- IO type: The input data and the output data are defined in this type. The maximum width of IO is set to 19 bits to fit into the 18x19 multiplier.
- Internal type: The internal type is for all internal operation of the neural network cell. The output of the multiplication blocks is set to this type.

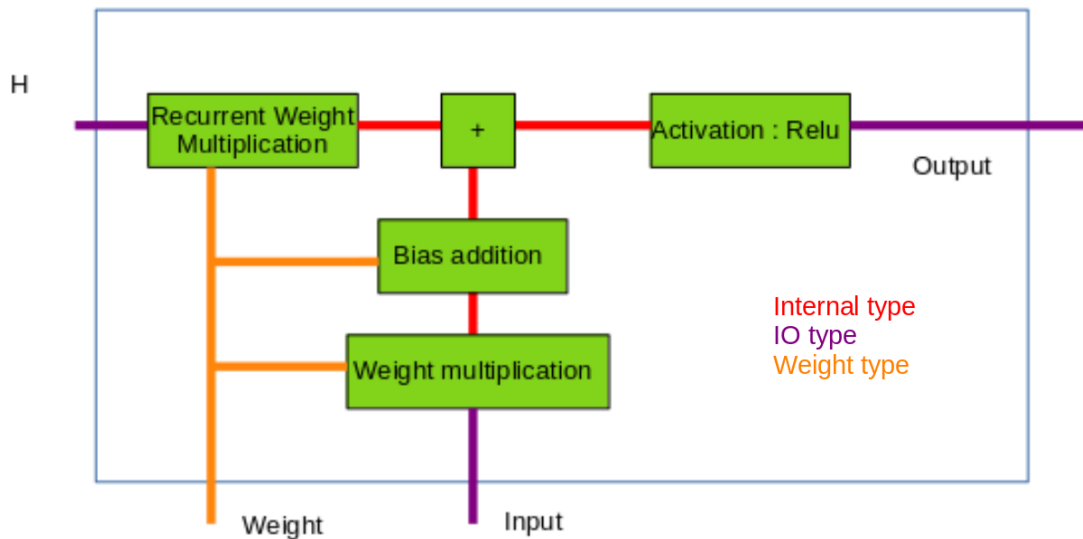


Figure 3.32.: Schematic of a vanilla cell with three different data types.

3. Embedded Recursive Neural Network in FGPA for the phase-II upgrade – 3.6. Firmware Implementation

Figure 3.33 shows the energy resolution of the vanilla network output compared to the full floating point computation for each of the quantization modes. For this plot 16 bits are used for the weight type width, and 19 bits for the internal and data type width. Each quantization mode is simultaneously applied for the three types. The default truncation mode resolution is not symmetric: the implemented network tends to under-predict the energies. For all the other modes the resolution is relatively symmetric and shows comparable results. As detailed the table 3.7 all quantization modes have a cost in term of logic compared to the default TRN mode. This added logic increases the resource usage and the latency of the network. Figure 3.34 compares the latency and resource usage of the different modes. As expected, the TRN mode has lower resource usage and lower latency. The cost of the RND mode is lower than all other modes in term of logic elements and latency. For the rest of the section, only the RND rounding mode and TRN truncation mode will be studied.

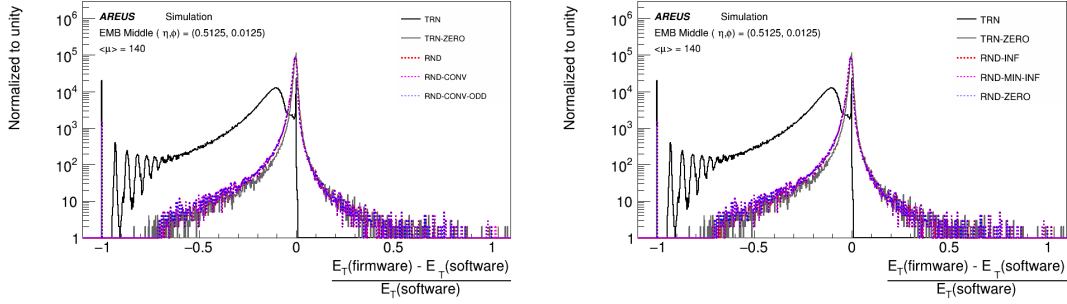


Figure 3.33.: Energy Resolution plot of the output of a vanilla sliding windows RNN with 5 timestamps. The weight have a width of 16 bits and the data have a width of 19. The different quantization modes are compared. The quantization mode is the same for the data, the internal, and the weights types.

3. Embedded Recursive Neural Network in FPGAs for the phase-II upgrade – 3.6. Firmware Implementation

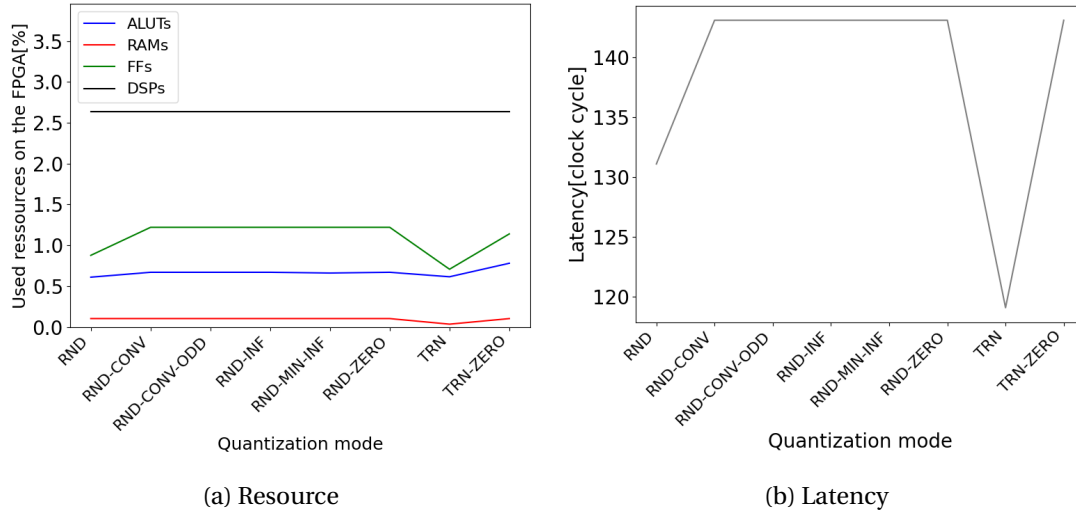


Figure 3.34.: Latency and resource usage of a vanilla RNN sliding windows with 5 timestamps depending of the quantization mode. The weight are sixteen bits width , the data are nineteen bits width.

The quantization has a different effect on the resolution and a different resource cost depending on the type of data being rounded. Figure 3.35 shows the resolution for several combinations of rounding for different data types. On subfigure (a) only one type is rounded. One can see that the rounding of the weights type is the only one giving significant improvement on the resolution. On subfigure (b) several combinations of type rounding is shown, weights type rounding is also kept for reference. The simultaneous rounding of the internal and IO types gives the worst resolution. The rounding of the weights and IO types and the rounding of weights and internal types gives similar resolutions. The best resolution is achieved when all types are rounded. Figure 3.36 shows the resource cost of these implementation. The rounding of the weights doesn't increase the resource usage. This rounding is done by the HLS compiler and not on hardware, thus no resources are needed. The small increase of resources and latency of the IO type is due to the quantization inside the dense layer. The type of all this layer is the same and is the IO type. A small amount of logic is implemented for this layer. As the cost is small the weights and data types will be rounded for the rest of this section. The rounding of the internal type increases the latency and the logic resource usage. The next part of this section will discuss how to mitigate this effect.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

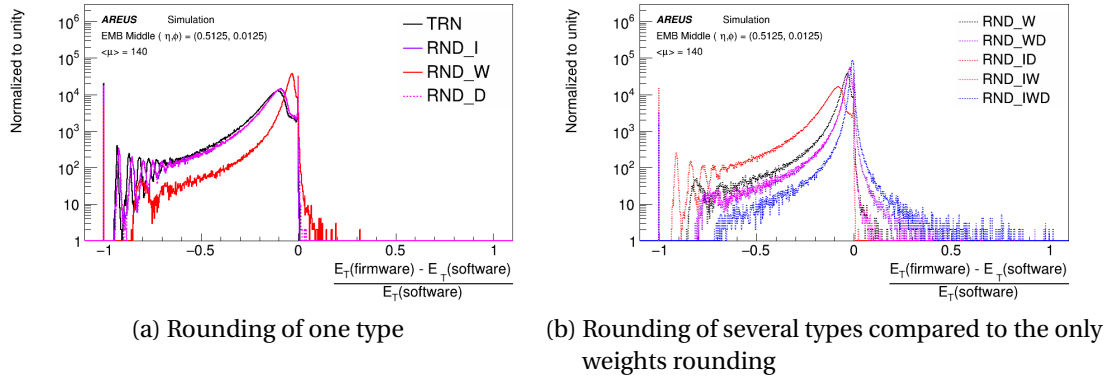


Figure 3.35.: Resolution plot of the output of a vanilla RNN sliding windows with 5 timestamps. The weight are sixteen bits width, the data are nineteen bits width. The plots compare the rounding of the types applied to the different data types defined in the vanilla cell. The rounded types are defined with RND followed by one or several letter indicating which types are rounded, the others being kept in TRN truncation mode: I for internal type, W for the weight type, and D for IO data type.

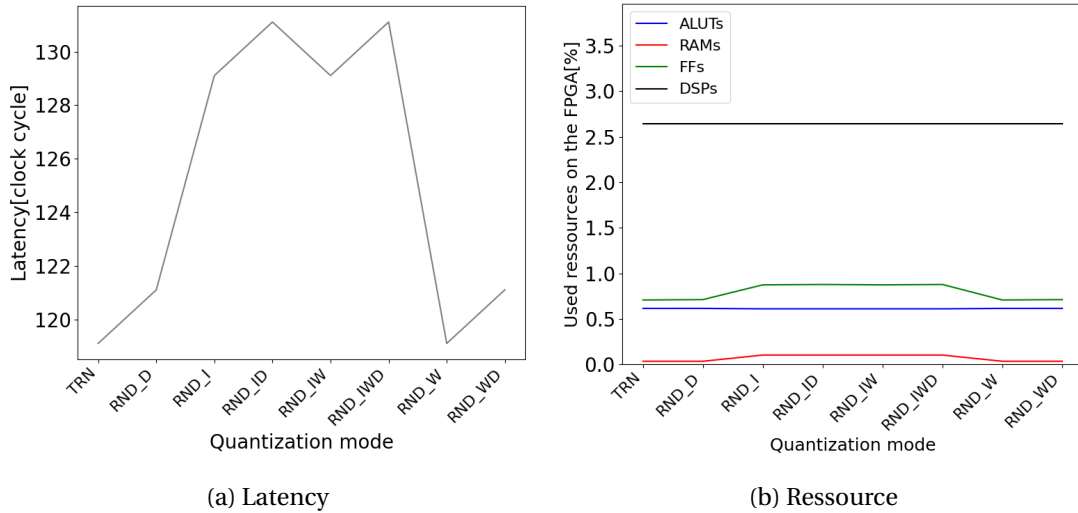


Figure 3.36.: Resources usage and latency of a 5 cell vanilla network depending on the quantization mode of its data types. Several combinations of the quantization are represented. TRN indicates that all the data types are in truncation mode. The rounded types are defined with RND followed by one or several letters indicating which types are rounded, the others being kept in TRN truncation mode: I for internal type, W for the weight type, and D for IO data type.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.6. Firmware Implementation

Figure 3.37 shows the energy resolution of the network depending on the number of bits of the internal types. The mean and the RMS of the energy resolution of the TRN quantized network with 23 bits internal width are similar to the mean and the RMS of the energy resolution of the RND quantized network with 20 bits internal width. Figure 3.38 shows the resources and latency of these implementations. The ALUTs used are similar for more implementations and the rounding uses more Flip Flops. The latency is larger if rounding is applied.

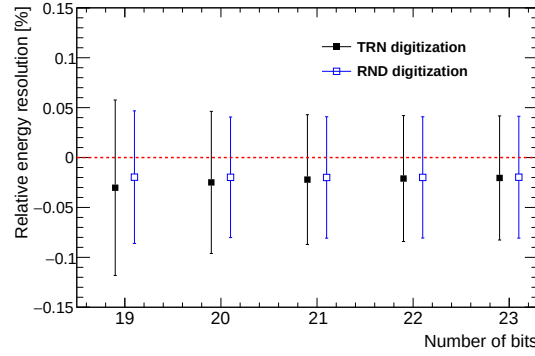


Figure 3.37.: Resolution of the network depending of the number of bits and of the quantization of the internal fixed point representation.

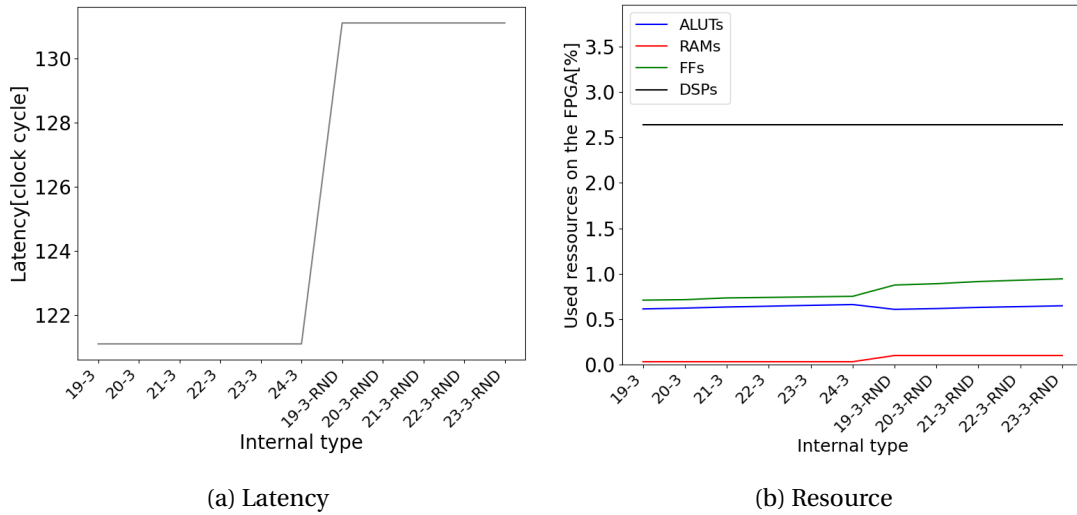


Figure 3.38.: Resources usage and latency of a 5 cell vanilla network depending on the size and rounding of its internal data type. The data types are written X-3 with X the total number of bits and 3 the number of bits on the integer part. RND is added at the end if the type uses RND quantization mode.

For the LAr network the fixed point representation with rounding is chosen for the IO type and the weights type since it provides the best performance without a

significant increase of latency or resources. For the internal types a width of 23 bits with the truncation mode provides similar performance to rounding with 19 bits without a significant increase of latency. This 23 bits width will be used for the LAr network.

3.7. Firmware performance

The chosen network to be implemented and tested on hardware is the vanilla network with 5 cells in sliding windows design. The key parameters of the implementation are summarised in the table 3.8. The network implements all the optimisation presented in section 3.6. Figure 3.39 compares the resolution of this vanilla network as implemented in software, the resolution of its implementation in firmware and the resolution of the OF algorithm. The firmware implementation slightly decreases the performance of the network but still outperforms the optimal filtering algorithm.

Parameter	Value
Multiplier accumulation	ACC19
Activation function	Relu
IO type	Fixed point 19 bits with 3 bits of integer, RND quantization
Weight type	Fixed point 16 bits with 3 bits of integer, RND quantization
Internal type	Fixed point 23 bits with 3 bits of integer, TRN quantization
Multiplexing	10
Targeted frequency	500 MHz (400 MHz + 100 MHz to facilitates the routing)

Table 3.8.: Parameters of the Vanilla RNN network implementation chosen for hardware testing.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.7. Firmware performance

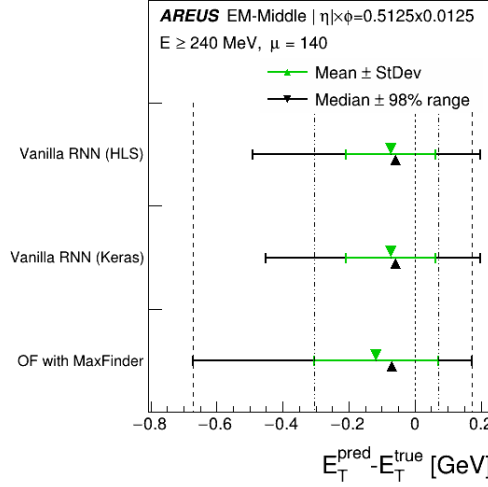


Figure 3.39.: Transverse energy reconstruction performance for the optimal filtering and the vanilla RNN implemented both in software (keras) and firmware (HLS). The performance is assessed by comparing the true transverse energy deposited in an EMB middle LAr cell ($\eta=0.5125$ and $\Theta=0.0125$) to the RNN prediction after simulating the sampled pulse with AREUS. Only energies 3σ above the noise threshold are considered. The mean, the median, the standard deviation, and the smallest range that contains 98% of the events are shown.

3.7.1. HLS resource estimate

The HLS estimated resource usage of this design is detailed in table 3.9. The DSP is the most critical resource. This design allows to simultaneously fit thirty seven networks in one FPGA. Thirty seven networks are able to compute the energies from 370 detector cells due to multiplexing. This allows one LASP FPGA to process the data for 2 FEB2 boards.

ALUTs	FFs	RAMs	MLABs	DSPs
11602 (1%)	26288 (1%)	4 (0%)	382 (0%)	152(3%)

Table 3.9.: HLS estimated resource usage of the vanilla RNN implementation.

3.7.2. Implementation and testing on hardware

A test setup has been developed to test the implemented firmware on the hardware. This setup uses a Stratix 10 intel devkit containing a Stratix 10 GX speed grade 2 FPGA. The Stratix 10 GX contains exactly the same resources as the SX ones except the absence of an ARM core. The SX FPGA is the one present on the LASP test board and foreseen for the prototype. A mezzanine board is connected to the devkit through the FMC connector. This mezzanine board was developed at CPPM to add optical

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.7. Firmware performance

transceivers to the devkit. With the Intermezzo board the devkit is able to emulate most of the LASP board functionalities with a reduced number of optical channels. Figure 3.40 shows the setup including the devkit and the Intermezzo.

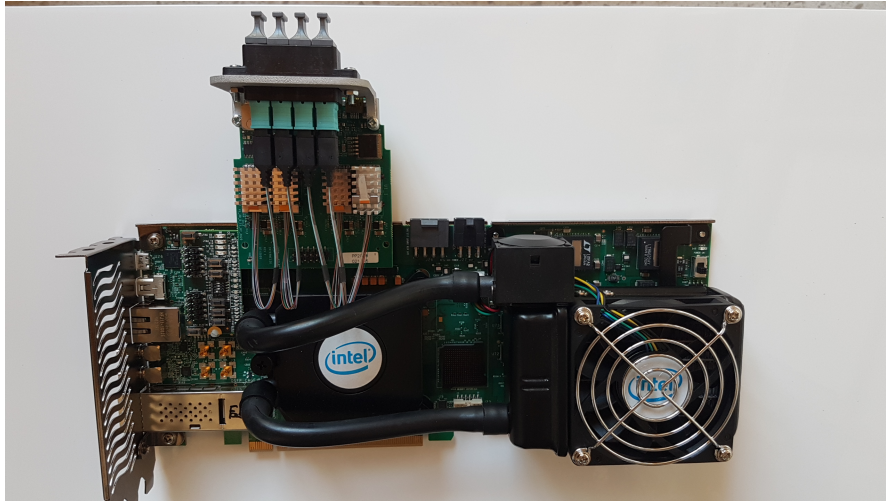


Figure 3.40.: The Stratix 10 GX devkit with the connected Intermezzo board which is used to test the firmware containing the neural networks.

Figure 3.41 shows the schematic of the firmware implemented in the FPGA for the test setup. It is composed of three main blocks that are nested one in the others:

- Base_Network: Component produced by the HLS compiler.
- Neural_Network: VHDL entity containing the HLS produced network in addition to its weight memory and the corresponding logic. Weight_Ram is a memory that contains at each address the value of the weights of the network for one detector cell. The cpt_addr block handles the address selection of the Weight_Ram. The delay blocks function is to synchronise the different signals.
- BaseNetwork_Top: VHDL entity made to be able to test the neural network. The FSM block handles the logic to init, reset, and start the Neural_Network block. Data_Ram contains a set of data provided by the AREUS simulation software. Cpt_addr handles the address selection of the data_ram. Cpt_wait is a counter needed by the FSM logic to handle ddelay between states. The FIFO at the output contains the energy processed by the network.

This firmware is duplicated several times on the FPGA to handle more channels.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.7. Firmware performance

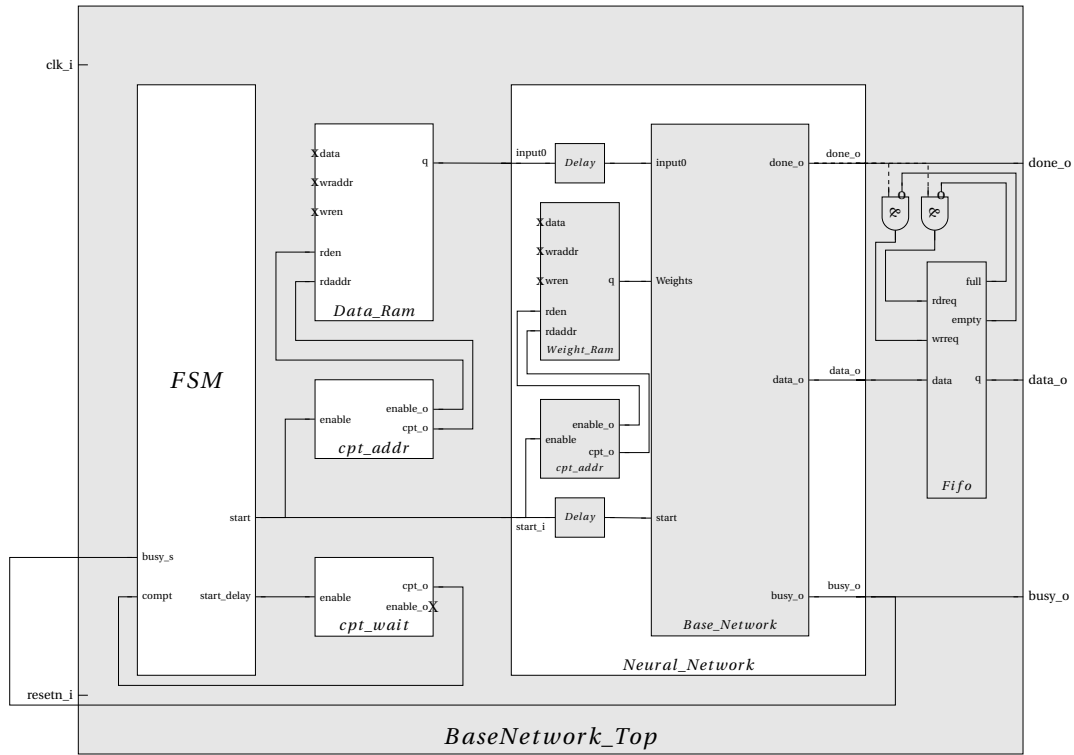


Figure 3.41.: Schematic of the firmware used to test the network produced by the HLS compiler (Base_Network). The Neural_Network block contains the HLS produced network in addition to its weight memory and the corresponding logic. The BaseNetwork_Top block contains all the logic to generate the signals and the data needed by the Neural_Network block.

Figure 3.42 shows the resource usage of the firmware as function of the number of implemented networks. The maximum number of networks in one FPGA according to the resource usage given by HLS compiler is 37. The number of needed DSP given by Quartus software corresponds exactly to the number predicted by the HLS compiler. The number of DSP implemented in the resulting firmware is larger than the needed number. For multiplications without output summing only half a DSP is needed. During the placement quartus implements these multiplications in DSP blocks. Sometimes the corresponding second half of the DSP is not used to facilitate routing. At higher FPGA occupancy the Quartus compiler is forced to fully use the two halves of each DSP to be able to fit all the required multiplications within the available DSPs. The same effect is also seen in the ALM usage, where only one ALUT is implemented in some ALM to facilitate routing. However the number of needed ALMs as computed by Quartus is 2 times larger than the one expected by the HLS compiler. Table 3.8 shows this difference. The exact reason for this large difference is currently under investigation.

3. Embedded Recursive Neural Network in FGPAs for the phase-II upgrade – 3.7. Firmware performance

Figure 3.42 shows the maximum clock frequency of the neural network firmware depending on the number of implementations. As the network is designed targeting a multiplexing of ten, a maximum frequency larger than 400 MHz is needed. At high occupancy this is not fulfilled by the speed grade 2 FPGA. However speed grade 1 FPGAs are capable of achieving frequencies higher than 400 MHz for this network implementation.

Type	Number of network	ALM	LABs(ALM + MLAB)	DSP	Fmax
HLS	1	5801(0.62%)	1620(1.7%)	152(3%)	NA
Quartus	1	24665(2.6%)	3036 (3.2%)	172 (2.6%)	456MHz
Quartus	37	833238(89.3%)	90688(97.2%)	5760(100%)	414 MHz

Table 3.10.: Comparison between the HLS and Quartus resource usage.

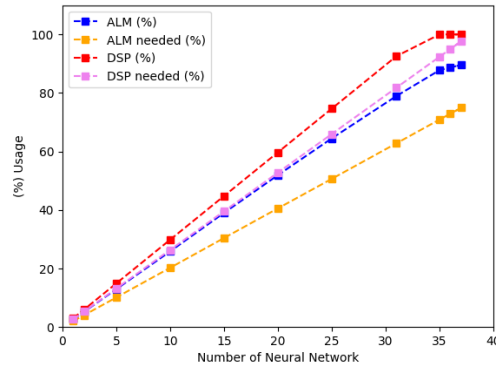


Figure 3.42.: Percentage of available ALM and DSP blocks used in the network implementation and corresponding needed percentage as a function of the number of implemented networks in the FPGA.

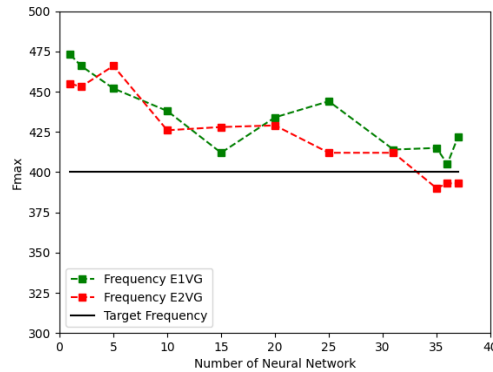


Figure 3.43.: Maximum frequency as a function of the number of implemented networks in one FPGA depending on the choice of the speed grade of the FPGA.

3.8. Conclusion

With the increased pile-up at the HL-LHC new algorithms are needed to reconstruct the energies deposited in the LAr calorimeter. With the increased computing capabilities of the new LASP readout boards, artificial intelligence algorithms are studied. Several RNN algorithms network have been developed at CPPM for this end. The network internal structure is studied in details to build RNN architectures with reduced computational resource needs while maintaining the performance in terms of energy resolution. The activation function block, the matrix multiplication block, and the quantization of the full network have been optimized. Additional studies are still needed to optimize the bias addition block and the dense layer block. With the current optimisations the vanilla network is capable of computing the energy for up to 370 channels and can fit into one Stratix 10 FPGA. The LSTM networks are found to be too large to fit the LAr calorimeter phase 2 requirements in terms of latency and number of channels per FPGA. However the LSTM networks can be used at a subsequent step to refine the energy computation at lower frequency e.g. for events selected by the first level of the trigger system. Additional optimizations for the vanilla network are needed to further reduce its size to be able to fit in the FPGA in real operation conditions. The FPGA is required to perform additional tasks and not all resources are available for the energy computation. The exact resources available for energy computation will depend on the final number of channels treated by one FPGA, the exact choice of the FPGA, and the resources needed by the other tasks. These ingredients are not fully known at the time of writing this document. However the vanilla RNN implementation presented in this chapter presents a promising candidate for the use of artificial intelligence for computing the energy deposited in the LAr calorimeter at the HL-LHC.

Conclusion

This document presents work on two subjects: the configuration of the new front-end boards of the phase 1 upgrade of the liquid argon calorimeter of ATLAS, and the design and implementation of an embedded firmware in the new backend board of the phase II upgrade that is responsible for computing the energy deposited in the calorimeter.

The LTDB board is the new frontend board installed on the detector. The LTDB is composed of several ASIC components. The board digitizes the electronic pulses of each channel of the trigger readout system called super cells. The digitized data are transmitted to the LATOME back-end board by the LOCx2 and MTx components. The slow control of the board is done through an SCA component providing several I2C buses and GPIO for the slow control. The FELIX board installed in the USA15 cavern communicates with these SCAs. Several layers of software handle this slow control. The main software of this chain is the OPC-UA LAR LTDB server. This server provides the interface between the LTDB and the DCS and TDAQ systems in charge of the control of the ATLAS detector. This server is responsible for the configuration of the board, the calibration of its components through different calibration scans, and its monitoring. Furthermore it includes several methods to recover various errors that can occur on the LTDBs. The development of this server is now completed and it is in use for operating the detector at the start of the run 3 of the LHC. It has been fully validated during the pilot run of the LHC the occurred in October 2021.

The development and the implementation of a DCS GUI for the LTDB system is also presented in this document. This interface provides a clear, fast, and reliable access to the status and to the configuration of the LTDB system. All the monitoring values are archived and displayed in several panels. This system have been included in the general ATLAS DCS system and is used regularly by the shifters operating ATLAS. The full LTDB system has been validated and will be used for the run 3 of the LHC and beyond.

RNNs are an interesting candidate to efficiently and reliably compute the energy deposited in the LAr calorimeter at the high pileup conditions expected at the LHC. The ATLAS group at CPPM developed RNN algorithms that are found to outperform the current algorithms that are based on optimal filtering techniques. The energy reconstruction algorithms must run on the new phase II LASP boards composed of two Stratix 10 FGPA's. The RNNs have been implemented in Quartus HLS code which

was highly optimised to reduce FPGA resource usage and latency while keeping good performances.

Several designs have been tried at CPPM for the weight matrix multiplication blocks. The tested designs include matrix multiplication done fully in DSP block, matrix multiplication done in DSP and ALUT, and a matrix multiplication fully in ALUTs with weights constrained to powers of two. The matrix multiplication done in DSP and ALUT design was found to be more adapted to the different constraints of the LASP firmware requirements.

The fixed point representation and the internal quantization of the numerical values have been extensively studied. The choice between a rounding quantization mode or a truncation quantization mode has been made for the various numerical values depending on the mathematical operations that they are involved with. This allows to keep high performance without high latency and resource usage.

With all these optimisations and compromises a candidate network has been chosen. This network gives slightly degraded performance compared to the network implemented in software but still outperforms the OF legacy algorithm. Tests of this network on real FPGA have been performed using a Stratix 10 development kit from Intel. The network was capable of handling 370 channels of the LAr calorimeter on a single FPGA and fulfill the stringent frequency and latency requirements. Further optimisations are still needed to further reduce the resource usage and allow other needed functionalities to be implemented on the FPGA. The final choice of the energy reconstruction algorithm will be made depending of the choice of the FPGA for the final LASP board. However the developed vanilla RNN algorithm with its highly optimised firmware implementation is one of the most promising candidate for the LAr energy reconstruction at the HL-LHC.

Bibliography

- [1] G Aad, S Bentvelsen, G J Bobbink, et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008). Also published by CERN Geneva in 2010, S08003. 437 p. DOI: [10.1088/1748-0221/3/08/S08003](https://doi.org/10.1088/1748-0221/3/08/S08003). URL: <http://cds.cern.ch/record/1129811> (cit. on pp. 12, 41, 42, 45).
- [2] G. Aad et al. “ATLAS pixel detector electronics and sensors”. In: *JINST* 3 (2008), P07007. DOI: [10.1088/1748-0221/3/07/P07007](https://doi.org/10.1088/1748-0221/3/07/P07007) (cit. on p. 37).
- [3] Georges Aad, Anne-Sophie Berthold, Thomas Calvet, et al. “Artificial Neural Networks on FPGAs for Real-Time Energy Reconstruction of the ATLAS LAR Calorimeters”. In: *Computing and Software for Big Science* 5.1 (Oct. 2021), p. 19. ISSN: 2510-2044. DOI: [10.1007/s41781-021-00066-y](https://doi.org/10.1007/s41781-021-00066-y). URL: <https://doi.org/10.1007/s41781-021-00066-y> (cit. on p. 120).
- [4] O. Aberle, I Béjar Alonso, O Brüning, et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2020. DOI: [10.23731/CYRM-2020-0010](https://doi.org/10.23731/CYRM-2020-0010). URL: <https://cds.cern.ch/record/2749422> (cit. on p. 33).
- [5] *About JCOP*. URL: <https://jcop.web.cern.ch/jcop-framework/documentation> (visited on 12/15/2021) (cit. on p. 95).
- [6] M (CERN) Aleksa, W (Pittsburgh) Cleland, Y (Tokyo) Enari, et al. *ATLAS Liquid Argon Calorimeter Phase-I Upgrade: Technical Design Report*. Tech. rep. Final version presented to December 2013 LHCC. Sept. 2013. URL: <https://cds.cern.ch/record/1602230> (cit. on pp. 17, 119).
- [7] *Algorithmic C (AC) Datatypes*. Standard. Mentor graphics, June 2016 (cit. on p. 147).
- [8] C Alpigiani. *Performance of the ATLAS semiconductor tracker*. Tech. rep. Geneva: CERN, July 2014. URL: <https://cds.cern.ch/record/1730503> (cit. on p. 38).
- [9] Timothy Andeen, Jaroslav Ban, Gustaaf Brooijmans, et al. “Data sheet for Nevis ADC”. 2018 (cit. on pp. 68, 107).
- [10] *ATLAS Liquid Argon Calorimeter Phase-II Upgrade: Technical Design Report*. Tech. rep. Geneva: CERN, Sept. 2017. URL: <https://cds.cern.ch/record/2285582> (cit. on p. 120).
- [11] Ahmet Bingul. “The ATLAS TRT and its performance at LHC”. In: *J. Phys. Conf. Ser.* 347 (2012). Ed. by Serkant Ali Cetin, Peter Jenni, Veysi Erkan Ozcan, et al., p. 012025. DOI: [10.1088/1742-6596/347/1/012025](https://doi.org/10.1088/1742-6596/347/1/012025) (cit. on p. 38).

- [12] A Caratelli, S Bonacini, K Kloukinas, et al. “The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments”. In: *JINST* 10 (2015), p. C03034. DOI: [10.1088/1748-0221/10/03/C03034](https://doi.org/10.1088/1748-0221/10/03/C03034). URL: <https://cds.cern.ch/record/2158969> (cit. on pp. 65, 76).
- [13] ATLAS Collaboration. “Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s}=13\text{TeV}$ ”. In: *The European Physical Journal C* 76.5 (May 2016). DOI: [10.1140/epjc/s10052-016-4120-y](https://doi.org/10.1140/epjc/s10052-016-4120-y). URL: <https://doi.org/10.48550/arXiv.1603.05598> (cit. on p. 50).
- [14] ATLAS Collaboration. “Operation and performance of the ATLAS semiconductor tracker in LHC Run 2”. In: *JINST* 17 (Sept. 2021). All figures including auxiliary figures are available at <http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/SCTD-2019-01,P01013>. 55 p. DOI: [10.1088/1748-0221/17/01/P01013](https://doi.org/10.1088/1748-0221/17/01/P01013). arXiv: [2109.02591](https://arxiv.org/abs/2109.02591). URL: <https://cds.cern.ch/record/2780336> (cit. on p. 38).
- [15] *PICMG®3.0 Revision 3.0A dvancedTCA® Base Specification*. Standard. International Organization for Standardization, Apr. 2015. URL: https://www.picmg.org/wp-content/uploads/ECN001_PICMG_ATCA_3_0_R3_0-RELEASED-2015-04-20b.pdf (cit. on p. 59).
- [16] Klaus Hanke, Davide Aguglia, Maria Elena Angoletta, et al. “Status and Plans for the Upgrade of the CERN PS Booster”. In: (2015), THPF090. 3 p. URL: <https://cds.cern.ch/record/2141902> (cit. on p. 31).
- [17] D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory* (1st ed.) 2002. DOI: [10.4324/9781410612403](https://doi.org/10.4324/9781410612403) (cit. on p. 122).
- [18] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. URL: <https://doi.org/10.1162/neco.1997.9.8.1735> (cit. on p. 126).
- [19] IEC. *IEC TR 62541-1:2020: OPC Unified Architecture - Part 1: Overview and concepts*. Tech. rep. IEC, 2020 (cit. on p. 61).
- [20] *Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures*. Standard. International Organization for Standardization, July 2002 (cit. on pp. 65, 75).
- [21] *Keras framework*. <https://keras.io/>. Accessed: 2022-01-30 (cit. on p. 127).
- [22] Jayanth Kuppambatti, Jaroslav Ban, Timothy Andeen, et al. “A radiation-hard dual-channel 12-bit 40 MS/s ADC prototype for the ATLAS liquid argon calorimeter readout electronics upgrade at the CERN LHC. A radiation-hard dual-channel 12-bit 40 MS/s ADC prototype for the ATLAS liquid argon calorimeter readout electronics upgrade at the CERN LHC”. In: *Nucl. Instrum. Methods Phys. Res., A* 855 (June 2017), 38–46. 9 p. DOI: [10.1016/j.nima.2017.01.025](https://doi.org/10.1016/j.nima.2017.01.025). arXiv: [1706.01535](https://arxiv.org/abs/1706.01535). URL: <https://cds.cern.ch/record/2289495> (cit. on pp. 68, 69, 107).

- [23] Madysa, Nico. “AREUS: A Software Framework for ATLAS Readout Electronics Upgrade Simulation”. In: *EPJ Web Conf.* 214 (2019), p. 02006. DOI: [10.1051/epjconf/201921402006](https://doi.org/10.1051/epjconf/201921402006). URL: <https://doi.org/10.1051/epjconf/201921402006> (cit. on pp. 17, 120).
- [24] Wolfgang Mahnke, Stefan-Helmut Leitner, and Matthias Damm. *OPC Unified Architecture*. Springer, Berlin, Heidelberg, 2009 (cit. on p. 61).
- [25] Claudia Marcelloni De Oliveira. “Closing of the ATLAS calorimeters”. General Photo. Dec. 2014. URL: <https://cds.cern.ch/record/1973044> (cit. on p. 44).
- [26] Warren Mcculloch and Walter Pitts. “A Logical Calculus of Ideas Immanent in Nervous Activity”. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 127–147 (cit. on p. 121).
- [27] Esma Mobs. “The CERN accelerator complex - 2019. Complexe des accélérateurs du CERN - 2019”. In: (July 2019). General Photo. URL: <https://cds.cern.ch/record/2684277> (cit. on p. 32).
- [28] P Moreira, R Ballabriga, S Baron, et al. “The GBT Project”. In: (2009). DOI: [10.5170/CERN-2009-006.342](https://cds.cern.ch/record/1235836). URL: <https://cds.cern.ch/record/1235836> (cit. on p. 64).
- [29] P Moreira, A Marchioro, and Kloukinas. “The GBT: A proposed architecure for multi-Gb/s data transmission in high energy physics”. In: (2007). DOI: [10.5170/CERN-2007-007.332](https://cds.cern.ch/record/1091474). URL: <https://cds.cern.ch/record/1091474> (cit. on p. 75).
- [30] Paris Moschovakos, Henk Boterenbrood, Aimilianos Koulouris, et al. “A Software Suite for the Radiation Tolerant Giga-bit Transceiver - Slow Control Adapter”. In: (2019), WEPHA102. 5 p. DOI: [10.18429/JACoW-ICALEPCS2019-WEPHA102](https://cds.cern.ch/record/2778539). URL: <https://cds.cern.ch/record/2778539> (cit. on p. 80).
- [31] Piotr Nikiel, Paris Moschovakos, and Stefan Schlenker. “quasar : The Full-Stack Solution for Creation of OPC-UA Middleware”. In: (2019), MOPHA100. 5 p. DOI: [10.18429/JACoW-ICALEPCS2019-MOPHA100](https://cds.cern.ch/record/2778537). URL: <https://cds.cern.ch/record/2778537> (cit. on p. 61).
- [32] Piotr P. Nikiel and Krzysztof Korcyl. “Object Mapping in the OPC-UA Protocol for Statically and Dynamically Typed Programming Languages”. In: *Computing and Informatics* 37.4 (2018). ISSN: 2585-8807. URL: http://www.cai.sk/ojs/index.php/cai/article/view/2018_4_946/915 (cit. on p. 62).
- [33] Joao Pequena. “Computer generated image of the ATLAS inner detector”. Mar. 2008. URL: <https://cds.cern.ch/record/1095926> (cit. on p. 37).
- [34] Joao Pequena. “Computer generated image of the whole ATLAS detector”. Mar. 2008. URL: <https://cds.cern.ch/record/1095924> (cit. on pp. 10, 35).
- [35] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain”. In: *Psychological Review* (1958), pp. 65–386 (cit. on p. 122).

- [36] J. G. Saraiva. “The ATLAS Tile Calorimeter: Commissioning, calibration and preparation for collisions”. In: *2008 IEEE Nuclear Science Symposium Conference Record* (2008), pp. 2261–2265 (cit. on p. 48).
- [37] *Schematic of a Stratix 10 SX*. <https://www.intel.com/content/www/us/en/products/details/fpga/stratix/10/sx.html>. Accessed: 2022-01-30 (cit. on p. 133).
- [38] Frank Schiller and Tina Mattes. “ANALYSIS OF CRC-POLYNOMIALS FOR SAFETY-CRITICAL COMMUNICATION BY DETERMINISTIC AND STOCHASTIC AUTOMATA”. In: *IFAC Proceedings Volumes* 39.13 (2006). 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp. 944–949. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20060829-4-CN-2909.00157>. URL: <https://www.sciencedirect.com/science/article/pii/S1474667015345390> (cit. on p. 70).
- [39] Stefan Schlenker, Damian Abalo Miron, Ben Farnham, et al. “quasar - A Generic Framework for Rapid Development of OPC UA Servers”. In: (2015), WEB3002. 4 p. DOI: [10.18429/JACoW-ICALEPCS2015-WEB3002](https://doi.org/10.18429/JACoW-ICALEPCS2015-WEB3002). URL: <https://cds.cern.ch/record/2213489> (cit. on pp. 61, 62).
- [40] Alex Sherstinsky. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: <https://doi.org/10.1016/j.physd.2019.132306>. URL: <https://www.sciencedirect.com/science/article/pii/S0167278919305974> (cit. on p. 126).
- [41] Yosuke Takubo. *ATLAS IBL operational experience*. Tech. rep. Geneva: CERN, Nov. 2016. DOI: [10.22323/1.287.0004](https://doi.org/10.22323/1.287.0004). URL: <https://cds.cern.ch/record/2235541> (cit. on p. 38).
- [42] *The history of CERN*. URL: <https://timeline.web.cern.ch> (cit. on p. 30).
- [43] Maurizio Vretenar, J Vollaie, R Scrivens, et al. *Linac4 design report*. Vol. 6. CERN Yellow Reports: Monographs. Geneva: CERN, 2020. DOI: [10.23731/CYRM-2020-006](https://doi.org/10.23731/CYRM-2020-006). URL: <https://cds.cern.ch/record/2736208> (cit. on p. 31).
- [44] Weihao Wu. “FELIX: the New Detector Interface for the ATLAS Experiment”. In: *IEEE Trans. Nucl. Sci.* 66 (June 2018). This manuscript for conference record of 21st IEEE Real Time conference only, 986–992. 7 p. DOI: [10.1109/TNS.2019.2913617](https://doi.org/10.1109/TNS.2019.2913617). arXiv: [1806.10667](https://arxiv.org/abs/1806.10667). URL: <https://cds.cern.ch/record/2627658> (cit. on pp. 73, 74).
- [45] Le Xiao, Xiaoting Li, Datao Gong, et al. “LOCx2, a low-latency, low-overhead, 2 × 5.12-Gbps transmitter ASIC for the ATLAS Liquid Argon Calorimeter trigger upgrade”. In: *JINST* 11 (Sept. 2020), C02013. 9 p. DOI: [10.1088/1748-0221/11/02/C02013](https://doi.org/10.1088/1748-0221/11/02/C02013). arXiv: [2009.08627](https://arxiv.org/abs/2009.08627). URL: <https://cds.cern.ch/record/2266412> (cit. on p. 70).

- [46] Xiandong Zhao, Chonghan Liu, Datao Gong, et al. “Mid-board miniature dual channel optical transmitter MTx and transceiver MTRx”. In: *JINST* 11 (Sept. 2020), C03054. 5 p. DOI: [10 . 1088/1748-0221/11/03/C03054](https://doi.org/10.1088/1748-0221/11/03/C03054). arXiv: [2009 . 08623](https://arxiv.org/abs/2009.08623). URL: <https://cds.cern.ch/record/2730981> (cit. on p. 64).

Acronyms

ADC

Analog to Digital Converter.

ALM

Adaptative Logic Module.

AMC

Advanced Mezzanine card.

ASDBLR

Amplifier-Shaper-Discriminator with BaseLine Restauration.

ASIC

Application-Specific Integrated Circuit.

ATCA

Advanced Telecom Computer Architecture.

ATLAS

A Toroidal LHC Apparatus.

BCID

Bunch Crossing IDentification.

BCR

Bunch Crossing Reset.

CERN

European Organisation for Nuclear Research.

CMOS

Complementary Metal Oxide Semiconductor.

CPPM

Marseille Particles Physics Center.

CRC

Cyclic Redundant Checking.

CTP

Central Trigger Processor.

DCS

Detector Control System.

DDPU

Digital Data Processing Unit.

DP

Data-Points.

DPE

Data-Points Element.

DPT

Data-Points Type.

DSP

Digital Signal Processing.

DSS

Detector Security System.

DTMROC

Digital Time Measurement Read-Out Chip.

EC

External Control.

EMB

Electro-Magnetic Barrel.

EMEC

Electro-Magnetic End Cap.

EMECSpec0

Electro-Magnetic End Cap Special 0.

EMECSPEC1

Electro-Magnetic End Cap Special 1.

EOF

End Of Frame.

EYETS

Extended Year End Technical Stop.

FCAL

Forward CALorimeter.

FEB

Front End Board.

FEB2

Front End Board 2.

FELIX

Front-End Link eXchange.

FEX

Feature EXtractor.

FF

Flip Flop.

FPGA

Field Programmable Gate Array.

GBT

GigaBit Transceiver.

GUI

Graphical User Interface.

HDL

Hardware Description Language.

HDLC

High-level Data Link Control.

HEC

Hadronic End-Cap.

HL-LHC

High Luminosity Large Hadron Collider.

HLS

High Level Synthesis.

HLT

High Level Trigger.

IBL

Insertable B-Layer.

IC

Internal Control.

ITk

Inner Tracker.

JCOP

Joint COntrol Project.

L0A

Level 0 Accept.

L1

Level-1.

L1A

Level 1 Accept.

LARC

LAr Carrier.

LASP

Lar Signal Procccessor.

LDPB

Lar Digital Processing Board.

LDPS

Lar Digital Processing System.

LHC

Large Hadron Collider.

LSB

Layer Summing Board.

LSTM

Long Short Term Memory.

LTDB

LAr Trigger Digitization board.

MDAC

Multiplying Digital-to-Analog Converter.

NSW

New Small Wheel.

OF

Optimal Filtering.

OPC-UA

Open Platform Communications United Architecture.

PCB

printed circuit board.

PDB

Power Distribution Board.

PRBS

Pseudo Random Binary Sequence.

PS

Proton Synchrotron.

PSB

Proton Synchrotron Booster.

QUASAR

Quick opc-UA Server generAtion fRamework.

RLU

Rectified Linear Unit.

RNN

Recurrent Neural Network.

ROD

ReadOut Driver.

SAR

Successive Approximation Register.

SC

Super-Cell.

SCA

Slow Control Adapter.

SCADA

Supervisory Control And Data Acquistion.

SCT

Semi-Conductor Tracker.

SOF

Start Of Frame.

SPS

Super Proton Synchrotron.

TBB

Tower Builder Board.

TDAQ

Trigger and Data AcQuisition.

TRT

Transition Radiation Tracker.

TT

Trigger Tower.

TTC

Timing, Trigger and Control.

WINCC-OA

WINdows Control Command Open Architecture.

XML

eXtensible Markup Language.

ANNEXES

A. SCA channels

Channel	Code	Description
CTRL	0x00	SCA configuration registers
SPI	0x01	Serial Peripheral master Interface
GPIO	0x02	Parallel I/O interface
I2C0	0x03	I2C Serial interface – master 0
I2C1	0x04	I2C Serial interface – master 1
I2C2	0x05	I2C Serial interface – master 2
I2C3	0x06	I2C Serial interface – master 3
I2C4	0x07	I2C Serial interface – master 4
I2C5	0x08	I2C Serial interface – master 5
I2C6	0x09	I2C Serial interface – master 6
I2C7	0x0A	I2C Serial interface – master 7
I2C8	0x0B	I2C Serial interface – master 8
I2C9	0x0C	I2C Serial interface – master 9
I2CA	0x0D	I2C Serial interface – master 10
I2CB	0x0E	I2C Serial interface – master 11
I2CC	0x0F	I2C Serial interface – master 12
I2CD	0x10	I2C Serial interface – master 13
I2CE	0x11	I2C Serial interface – master 14
I2CF	0x12	I2C Serial interface – master 15
JTAG	0x13	JTAG serial master interface
ADC	0x14	Analog to digital converter
DAC	0x15	Digital to analog converter

Table .1.: List of all the channels of the SCA chip

B. List of all pannels of the ATLLARLTDB project

Pannel	Description
clkDesCalibAll.xml	Handle the clkdes calibration of all the boards in parrallel
clkDesCalibShort.xml	Sub pannel of clkDesCalibAll.xml
clkdesScanAll.xml	Handle the clkdes scan of all the boards in parrallel
clkDesScan.xml	Handle the clkdes calibration
clkDesShort.xml	Sub pannel of clkdesScanAll.xml
EnableDisable.pnl	Enable/disable a board
refAdc.pnl	Configuration of the ADC of one board
refCrateGlobalFsm.pnl	Control and monitoring of one crate
refDebugLtdbRead.pnl	Readback all register of the board
refLocx2Scan.pnl	Locx2 scan of one board
refMapping.pnl	Mapping generation and check
refStatusCrate_small.pnl	Monitoring of one crate
ttcCtrl.xml	Monitoring and control of the TTC on one FELIX
refMonGbtx.pnl	Monitoring of the GBTx register
refMonLtdb.pnl	Monitoring of one LTDB
refMonSca.pnl	Monitoring of one SCA
serversConnections_short_full.pnl	Monitoring of the connection between servers
LtdbControl_fsm	Pannel for control and monitoring of all LTDB

Table .2.: List of all panels of the ATLLARLTDB project

