



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ ΕΝΕΡΓΕΙΩΝ

**Σχεδίαση και Ανάπτυξη Firmware στα FPGA των Καρτών
L1DDC για την Αναβάθμιση του Ανιχνευτή ATLAS στο Μεγάλο
Επιταχυντή Αδρονίων του CERN**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Κ. Μπακάλη

Επιβλέπων: Θεόδωρος Αλεξόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2015



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΠΕΙΡΑΜΑΤΙΚΗΣ ΦΥΣΙΚΗΣ ΥΨΗΛΩΝ ΕΝΕΡΓΕΙΩΝ

**Σχεδίαση και Ανάπτυξη Firmware στα FPGA των Καρτών
L1DDC για την Αναβάθμιση του Ανιχνευτή ATLAS στο Μεγάλο
Επιταχυντή Αδρονίων του CERN**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Κ. Μπακάλη

Επιβλέπων: Θεόδωρος Αλεξόπουλος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 8 Οκτωβρίου 2015.

.....
Θ. Αλεξόπουλος
Καθηγητής Ε.Μ.Π.

.....
Ε. Γαζής
Καθηγητής Ε.Μ.Π.

.....
Σ. Μαλτέζος
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2015

.....
Χρήστος Κ. Μπακάλης
Φυσικός Εφαρμογών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική εργασία, πραγματεύεται κατά κύριο λόγο τον προγραμματισμό των Field-Programmable Gate Arrays (FPGA), που χρησιμοποιούνται στο σύστημα των front-end electronics της αναβάθμισης του New Small Wheel (NSW) που θα γίνει το 2018 στον ανιχνευτή ATLAS, στον LHC του CERN. Τα FPGA, είναι μία μεγάλη οικογένεια ολοκληρωμένων κυκλωμάτων, που μπορούν να υλοποιήσουν μία οποιαδήποτε λογική συνάρτηση, ανάλογα τις επιθυμίες του χρήστη του. Τα FPGA χρησιμοποιούνται στις πλακέτες L1DDC (Level-1 Data Driver Card), οι οποίες είναι υπεύθυνες για τη δρομολόγηση των πρωταρχικών δεδομένων για τις συγκρούσεις, όπως αυτά προκύπτουν από τις front-end boards που επεξεργάζονται τα σήματα των ανιχνευτών MicroMegas. Η σχεδίαση των καρτών L1DDC, και ο προγραμματισμός των FPGA τους, έχει πραγματοποιηθεί εξ' ολοκλήρου από την Ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών του Εθνικού Μετσοβίου Πολυτεχνείου. Εκτός από την ανάλυση των λειτουργιών της L1DDC και του firmware του FPGA που βαίνει πάνω της, στην παρούσα εργασία παρατίθενται και γενικές πληροφορίες για ολόκληρο το σύστημα των ηλεκτρονικών του NSW (π. χ. ανάλυση της λειτουργικότητας των GBTx και VMM ASIC), και επομένως αυτό το κείμενο μπορεί να χρησιμοποιηθεί και σαν γενική αναφορά για μελλοντικές εργασίες.

Abstract

This diploma thesis, mainly discusses the programming process of the Field-Programmable Gate Arrays (FPGAs), that are being used in the front-end electronics system of the New Small Wheel (NSW) upgrade which will take place in 2018 at the ATLAS detector of LHC at CERN. FPGAs belong to a wider family of integrated circuits, that are able to implement any logic function, according to the will of the user. The FPGAs are being integrated on the L1DDC (Level-1 Data Driver Card) boards, which are responsible for the routing of the raw data from the particle collisions, as these come out of the front-end boards which refine the signals from the MicroMegas detectors. The design of the L1DDC boards and the programming of their FPGAs, have all been carried out by the Experimental High Energy Physics Group of the National Technical University of Athens. Apart from analyzing the functionality of the L1DDC and the architecture of its FPGA firmware, this text also depicts the general functionality of the entire electronics system of the NSW (e. g. analysis of the VMM and GBTx ASICs), and can be used as a general reference for future projects.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλα τα μέλη της Ομάδας Πειραματικής Φυσικής Υψηλών Ενεργειών του ΕΜΠ., για την πολύτιμη βοήθειά τους. Πιο συγκεκριμένα, θα ήθελα να ευχαριστήσω τον κύριο Θεόδωρο Αλεξόπουλο, που μου έδωσε την ευκαιρία να εργαστώ πάνω σε ένα τόσο ενδιαφέρον θέμα. Ο...Αρχηγός, συντονίζει μία πραγματικά αξιοθαύμαστη ομάδα, που αναδεικνύει το ΕΜΠ και τη ΣΕΜΦΕ στο CERN.

Αυτό που πρέπει να αντιληφθεί κανείς πριν ξεκινήσει την ανάγνωση της παρούσας διπλωματικής, είναι το γεγονός ότι τα FPGA και η VHDL, είναι δύο πεδία που απέχουν *αρκετά* από το πρόγραμμα σπουδών της ΣΕΜΦΕ. Η αλήθεια είναι ότι όσο όρεξη να έχει κανείς να μάθει όλα αυτά τα (ωραία) πράγματα ξεκινώντας από το μηδέν, δεν θα καταφέρει τίποτα χωρίς τη σωστή καθοδήγηση. Έτσι, θα ήθελα να ευχαριστήσω και τον Παναγιώτη Γκουντούμη, ο οποίος όχι μόνο σχεδίασε τις πλακέτες πάνω στις οποίες δούλεψα, αλλά και με βοήθησε στην εκμάθηση της γλώσσας VHDL που χρησιμοποιήθηκε για την ανάπτυξη του firmware. Αν δεν υπήρχε αυτός...τόρα δεν θα δούλευε τίποτα!

Τέλος, θα ήθελα να ευχαριστήσω και τον Γιώργο Ιακωβίδη, ο οποίος με βοήθησε αρκετά στη συγγραφή του παρόντος κειμένου, καθώς με καθοδήγησε για τις αναφορές που έπρεπε να χρησιμοποιήσω, και πήρε το χρόνο του για να μου λύσει αρκετές απορίες, για το ομολογουμένως πολύ περίπλοκο σύστημα ηλεκτρονικών του New Small Wheel. Το κείμενο της διδακτορικής του διατριβής, αποτέλεσε κεντρικό άξονα για τη συγγραφή της παρούσας διπλωματικής εργασίας.

Και εννοείται φυσικά, ότι δεν θα είχα φτάσει ποτέ σε αυτό το σημείο εάν δεν είχα

την υποστήριξη και την αγάπη των γονιών μου. Τους οφείλω τα πάντα, και σε τελική ανάλυση, δεν υπάρχουν λόγια για να τους εκφράσω την ευγνωμοσύνη μου.

Περιεχόμενα

1	Το Πείραμα ATLAS στο CERN	1
1.1	Αναζητώντας τα Στοιχειώδη Σωματίδια	1
1.1.1	Το Καθιερωμένο Πρότυπο	2
1.2	Ο Μεγάλος Επιταχυντής Αδρονίων	4
1.3	Ο Ανιχνευτής ATLAS	8
1.4	Η Αναβάθμιση του New Small Wheel	14
2	Ο Ανιχνευτής MicroMegas	19
2.1	Εισαγωγή - Ανιχνευτές Αερίων	19
2.2	Ο Ανιχνευτής MicroMegas	22
3	Ηλεκτρονικά Συστήματα Ανιχνευτών	27
3.1	Εισαγωγή στα Ηλεκτρονικά των Ανιχνευτών	27
3.2	Η Επεξεργασία των Σημάτων	29
3.2.1	Τα Προβλήματα που Εμφανίζονται	30
3.3	Ηλεκτρονικά Επεξεργασίας Σημάτων	33
4	Το Σύστημα Ηλεκτρονικών του New Small Wheel	49
4.1	Εισαγωγή στα ASIC και στα FPGA	49
4.1.1	Application-Specific Integrated Circuits (ASIC)	51
4.1.2	Field-Programmable Gate Arrays (FPGA)	53
4.2	Γενική Επισκόπηση του Συστήματος Ηλεκτρονικών του NSW	55
4.3	MicroMegas Front-End Board	59
4.3.1	VMM ASIC	61

4.4	Level-1 Data Driver Card	67
4.4.1	Γενική Επισκόπηση της L1DDC	68
5	Το Πρώτο Προτότυπο της Κάρτας L1DDC	75
5.1	Γενικά για την L1DDC Prototype1 - Ο Ρόλος του FPGA	75
5.1.1	GBTx ASIC	79
5.1.2	Artix-7 FPGA	86
5.2	Λοιπές Πλακέτες	87
6	Field-Programmable Gate Arrays	91
6.1	Η Αρχιτεκτονική των FPGA	93
6.1.1	LUTs, Slices, CLBs	94
6.1.2	Hard Blocks	100
6.2	Το Δίκτυο Διασυνδέσεων του FPGA	107
6.3	Η Ροή της Πληροφορίας	111
7	Firmware για το FPGA της Κάρτας L1DDC	117
7.1	I ² C FPGA ↔ GBTx	119
7.1.1	Chipscope ILA/ICON	126
7.2	Ethernet/UDP Interface	127
7.3	E-Link Interface	134
7.4	Debugging Firmware	138
7.4.1	Απλές Διαδικασίες Ελέγχου των E-link	139
7.4.2	Σύνθετες Διαδικασίες Ελέγχου των E-link	145
Παράρτημα Α'	Πρωτόκολλα Επικοινωνίας	151
Α'.1	Low-Voltage Differential Signaling	151
Α'.2	Ethernet - UDP Protocol	153
Α'.3	8b/10b Encoding	155
Α'.4	I ² C Protocol	156
Παράρτημα Β'	VHDL Tutorial	159
Β'.1	Xilinx® ISE Design Suite 14.7	159
Β'.2	Hello World!	162
Β'.3	Serializer	168

B.4 Finite-State Machines	179
Παράρτημα Γ' Block Diagrams	183
Παράρτημα Δ' Δείγματα Firmware σε VHDL	187
Βιβλιογραφία	197

1

Το Πείραμα ATLAS στο CERN

Στο παρόν κεφάλαιο, μετά από μία σύντομη νύξη για τις εδραιωμένες θεωρίες περί Στοιχειωδών Σωματιδίων, θα πραγματοποιηθεί μία γενική επισκόπηση του πειράματος ATLAS που διεξάγεται στο Ευρωπαϊκό Κέντρο Πυρηνικών Ερευνών (CERN: Conseil Européenne pour la Recherche Nucléaire), και της αναβάθμισης του New Small Wheel (NSW), που είναι και το ουσιαστικό κίνητρο για την εργασία αυτή.

1.1 Αναζητώντας τα Στοιχειώδη Σωματίδια

Η Φυσική των Στοιχειωδών Σωματιδίων, λειτουργεί έχοντας σαν κύριο άξονα την εξής ερώτηση: "Από τι είναι φτιαγμένη η ύλη;", και αποπειράται να απαντήσει σε αυτό το ερώτημα, μελετώντας τις δομές των στοιχειωδών σωματιδίων σε όσο το δυνατόν περισσότερο θεμελιώδες επίπεδο. Είναι αρκετά εντυπωσιακό το γεγονός ότι η ύλη του Σύμπαντος αποτελείται τελικά από μικροσκοπικά κομμάτια συστατικών, τα οποία όταν ανασυνδυάζονται μεταξύ τους με διαφορετικούς τρόπους, κατασκευάζουν κάθε φορά και μία διαφορετική έκφανση της ύλης. Αυτό φυσικά δε σημαίνει ότι όλα τα στοιχειώδη σωματίδια είναι διαφορετικά μεταξύ τους· κάθε άλλο. Αν κάποιος παρατηρήσει ένα ηλεκτρόνιο, δε χρειάζεται να παρατηρήσει κανένα άλλο. Είναι σαν να τα έχει δει όλα. Όμως ακόμα και αυτό το γεγονός, δεν κάνει τη ζωή των Φυσικών ευκολότερη. Κανείς δεν έχει υποστηρίξει μέχρι τώρα ότι έχουν λίγη δουλειά να κάνουν!

Οι ρίζες της εικασίας ότι τα πάντα αποτελούνται από μικρά κομμάτια δομικών λί-

θων ύλης, μπορούν να εντοπιστούν ακόμα και στα χρόνια των αρχαίων Ελλήνων Φιλοσόφων, με κύριο υποστηρικτή τον Δημόκριτο (460-370 π.Χ.), ο οποίος πίστευε ότι όλα γύρω του αποτελούντο από μικρά αδιαίρετα τμήματα βασικής ύλης, τα άτομα. Οι σκέψεις αυτές, αν και επρόκειτο μόνο για μεταφυσικές εικασίες, έδωσαν κίνητρο σε πολλούς σύγχρονους επιστήμονες να εκτελέσουν σημαντικά πειράματα για τον προσδιορισμό και τη μελέτη των βασικών τμημάτων της ύλης. Οι πιο καρποφόρες μελέτες έγιναν από τα τέλη του 19ου αιώνα, μέχρι και τα μέσα του 20ου, με μεγάλα ονόματα Φυσικών να εμπλουτίζουν το γνωστικό υπόβαθρο για τα στοιχειώδη σωματίδια. Για παράδειγμα, το 1897, ο J.J. Thomson ανακάλυψε το ηλεκτρόνιο, ενώ το 1899 ο E. Rutherford με το πείραμα σκέδασης σωματιδίων άλφα σε λεπτά φύλλα χρυσού, ενίσχυσε τις εικασίες που ήθελαν την ύλη να αποτελείται από μικροσκοπικά άτομα. Επίσης, η θεωρητική συμβολή του A. Einstein το 1905 με την ανάλυση του φωτοηλεκτρικού φαινομένου που εξήγησε την κβάντωση της ηλεκτρομαγνητικής ακτινοβολίας, και η ταυτόχρονη ανάπτυξη της κβαντομηχανικής, έδωσαν τις βάσεις για μία ολοκληρωμένη θεωρία για τα στοιχειώδη σωματίδια.

1.1.1 Το Καθιερωμένο Πρότυπο

Το τελικό αποτέλεσμα όλων αυτών των επίπονων μελετών (οι περισσότερες εκ των οποίων δεν έχουν αναφερθεί εδώ για λόγους συντομίας), είναι αυτό που ονομάζουν σήμερα οι Φυσικοί *Καθιερωμένο Πρότυπο* (Standard Model of Particle Physics). Πρόκειται για μία γενική θεωρία που εξηγεί με συνέπεια τη δομή των βασικών μορφών ύλης, και τη φύση των μεταξύ τους αλληλεπιδράσεων. Σύμφωνα με αυτό το πρότυπο, η ύλη είναι κατασκευασμένη αποκλειστικά από συνδυασμούς *quark* (κουάρκ), και λεπτονίων. Τα *quark*, η πιο βασική μονάδα ύλης, ζευγαρώνουν είτε ανά δύο για να δημιουργήσουν τα μεσόνια (mesons), ή ανά τρία για να δημιουργήσουν τα βαρυόνια (baryons). Τα *quark* και τα λεπτόνια, αλληλεπιδρούν μεταξύ τους είτε με ηλεκτρομαγνητικές αλληλεπιδράσεις, είτε με ισχυρές ή ασθενείς πυρηνικές αλληλεπιδράσεις. Οι φορείς των αλληλεπιδράσεων αυτών είναι τα φωτόνια (γ), τα W^\pm , Z^0 μποζόνια, και τα γλουόνια. Όλες αυτές οι εξωτικές οντότητες μπορούν να ομαδοποιηθούν στο Σχήμα 1.1.I:

Elementary Particles

Quarks	u up	c charm	t top	Force Carriers
	d down	s strange	b bottom	
Leptons	ν_e e neutrino	ν_μ μ neutrino	ν_τ τ neutrino	W W boson
	e electron	μ muon	τ tau	Z Z boson
3 →	I	II	III	← Generations

Σχήμα 1.1.1: Η λίστα με τα δομικά στοιχεία της ύλης σύμφωνα με το Καθιερωμένο Πρότυπο.

Αξίζει να σημειωθεί, ότι η ύλη σήμερα αποτελείται αποκλειστικά από quark και λεπτόνια πρώτης γενιάς. Αυτά είναι τα up και down quark, και τα ηλεκτρόνια με τα νετρίνο (neutrino) τους. Τα πρωτόνια και τα νετρόνια σχηματίζονται από τα quark αυτά, ενώ τα ηλεκτρόνια περιτριγυρίζουν τους διαφορετικούς συνδυασμούς πρωτονίων και νετρονίων για να σχηματίσουν τελικά τα άτομα, τα χημικά στοιχεία από τα οποία είναι φτιαγμένα όλα όσα μπορεί να φανταστεί κανείς: από τους αστέρες, μέχρι τις πρωτεΐνες στα κύτταρα των έμβιων όντων. Υπάρχουν όμως και αναρίθμητα άλλα σωματίδια, τα οποία σχηματίζονται από τα πιο βαριά quark, αλλά είναι σχεδόν αδύνατο να παρατηρηθούν, καθώς οι υπάρχουσες συνθήκες στο σημερινό Σύμπαν δεν επιτρέπουν στα σωματίδια αυτά να υφίστανται για τόσο χρονικό διάστημα ώστε να είναι παρατηρήσιμα.

Άρα τελικά, η εικόνα αρχίζει να σχηματίζεται: Σύμφωνα με τις καθιερωμένες αντιλήψεις που θέλουν το πρώιμο Σύμπαν να είναι θερμό και με υψηλές ενέργειες, ο στόχος της σύγχρονης φυσικής στοιχειωδών σωματιδίων είναι πλέον ξεκάθαρος: Προκειμένου να μελετηθούν σε βάθος οι δομές και οι αλληλεπιδράσεις μέσα στην ύλη, πρέπει να ξεπεραστούν οι χαμηλές ενέργειες που διέπουν το σημερινό Σύμπαν, ένα Σύμπαν ψυχρό, και εν τέλει...βαρετό σε σχέση με πριν από πολλά δισεκατομμύρια χρόνια, όπου υπήρχε μία πληθώρα σωματιδίων, και που οι δυνάμεις μεταξύ των δεν ήταν διακριτές όπως σήμερα, αλλά όπως δείχνουν πρόσφατες μελέτες, μάλ-

λον ήταν ενοποιημένες [1]. Έτσι λοιπόν, λόγω της ανάγκης για εξερεύνηση σε όλο και μικρότερες τάξεις μεγέθους, για τη μελέτη όλο και βαρύτερων και σπανιότερων σωματιδίων που θα συμπληρώσουν το παζλ του Καθιερωμένου Προτύπου, γεννήθηκε η ανάγκη για όλο και *Υψηλότερες Ενέργειες*. Έτσι οι επιστήμονες ξεκίνησαν να μελετούν θεωρητικά τις Υψηλές Ενέργειες, και να κατασκευάζουν συσκευές που θα τους επιτρέψουν να τις προσεγγίσουν. Μία τέτοια συσκευή, είναι και ο Μεγάλος Επιταχυντής Αδρονίων, που βρίσκεται στο CERN.

1.2 Ο Μεγάλος Επιταχυντής Αδρονίων

Γενικά, μπορεί να πει κανείς ότι οι επιταχυντές είναι τα "μικροσκόπια" για τους Φυσικούς Υψηλών Ενεργείων [3]. Ιστορικά, ως πρώτος επιταχυντής μπορεί να θεωρηθεί ο καθοδικός σωλήνας που κατασκεύασε το 1895 ο Röntgen για να παράξει ακτίνες X, όπου μία διαφορά δυναμικού επιτάχυνε ηλεκτρόνια σε ενέργειες της τάξης των μερικών keV ¹. Σήμερα, οι ενέργειες που αγγίζουν οι επιταχυντές είναι της τάξης των TeV , και δεν επιταχύνουν μόνο ηλεκτρόνια, αλλά και πρωτόνια ή και βαρέα ιόντα. Στα πειράματα αυτά, μία δέσμη σωματιδίων επιταχύνεται είτε σε ευθεία, είτε σε κυκλική τροχιά με τη βοήθεια ηλεκτρομαγνητών που παράγουν πεδία αρκετά ισχυρά ώστε να αυξάνουν την ταχύτητα των σωματιδίων ενώ ταυτόχρονα τα κατευθύνουν σε αυστηρά καθορισμένες τροχιές. Τελικά τα σωματίδια αυτά συγκρούονται είτε σε σταθερούς στόχους, είτε με άλλες δέσμες σωματιδίων, για να παράξουν βαρέα σωματίδια υψηλών ενεργειών, που στη συνέχεια μελετώνται από τους ερευνητές.

Ο Μεγάλος Επιταχυντής Αδρονίων (Large Hadron Collider, LHC), είναι ο μεγαλύτερος και ισχυρότερος επιταχυντής που έχει κατασκευαστεί ποτέ [6], και ξεκίνησε επίσημα τη λειτουργία του το 2008. Πρόκειται για έναν κυκλικό επιταχυντή όπου υπέρλεπτες δέσμες πρωτονίων επιταχύνονται τόσο πολύ που πλησιάζουν την ταχύτητα του φωτός, ώσπου τελικά αλληλοσυγκρούονται μεταξύ τους παράγοντας έτσι εξωτικά σωματίδια. Η πειραματική αυτή διάταξη βρίσκεται εκατοντάδες μέτρα κάτω από την επιφάνεια της Γης, στα σύνορα Γαλλίας-Ελβετίας, ενώ ο κυκλικός σωλήνας

¹Υπενθυμίζεται ότι η μονάδα ενέργειας του eV (ηλεκτρονιοβόλτ, electronvolt) ισοδυναμεί με την ενέργεια που αποκτά ένα ηλεκτρόνιο, όταν αυτό επιταχυνθεί από μία διαφορά δυναμικού της τάξης του $1 V$.

τον οποίο διατρέχουν τα πρωτόνια έχει περιφέρεια 27 km .

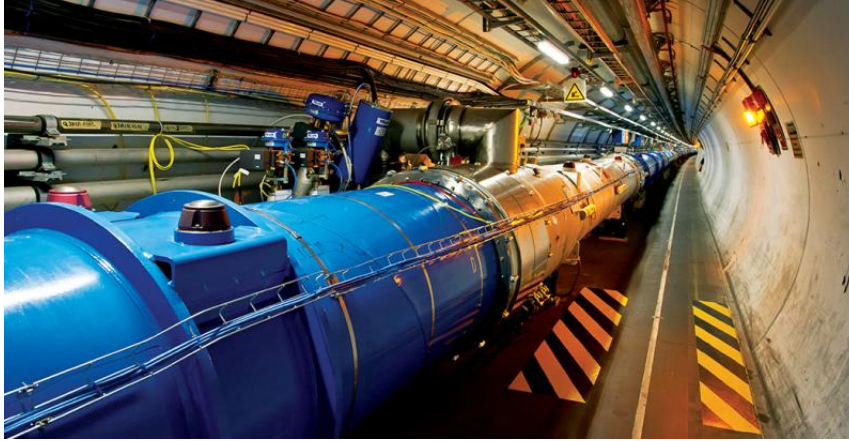


Σχήμα 1.2.1: Αεροφωτογραφία του CERN και του LHC. Είναι ευδιάκριτος ο μεγάλος δακτύλιος που επιταχύνονται τα σωματίδια, ενώ στα δεξιά φαίνεται το αεροδρόμιο και η λίμνη της Γενεύης. Ο μικρότερος δακτύλιος μέσα στον κύριο, είναι ο SPS, ο οποίος διοχετεύει τον κύριο δακτύλιο του LHC με πρωτόνια ενέργειας μερικών εκατοντάδων GeV [6].

Μία σειρά από όλο και μεγαλύτερους δακτυλίους επιταχύνουν προοδευτικά πρωτόνια που προέρχονται από μία απλή φιάλη υδρογόνου, τα οποία πρωτόνια θα καταλήξουν τελικά στον μεγάλο δακτύλιο του LHC, ο οποίος διαχωρίζει δύο δέσμες πρωτονίων και τις ανακυκλώνει για πολλές ώρες σε αντίθετες κατευθύνσεις μέσα του². Οι δέσμες αυτές μπορούν να συγκρουσθούν μεταξύ τους σε συγκεκριμένα σημεία πάνω στον δακτύλιο, και οι ισχυρές αυτές συγκρούσεις απελευθερώνουν μεγάλα ποσά ενέργειας, δηλαδή βαριά σωματίδια, ή φωτόνια υψηλής συχνότητας, που παρέχουν πολύτιμες πληροφορίες για τη δομή της ύλης στους ερευνητές του CERN. Ακριβώς σε αυτά τα διαφορετικά σημεία των συγκρούσεων υπάρχουν οι *ανιχνευτές*, δηλαδή εξειδικευμένες διατάξεις υψηλής τεχνολογίας, που μπορούν να ανιχνεύσουν με ακρίβεια όλα τα σωματίδια που παράγονται από τις συγκρούσεις,

²Η διαδικασία της επιτάχυνσης και ανακατεύθυνσης των δεσμών επιτυγχάνεται με ισχυρά ηλεκτρομαγνητικά πεδία που δημιουργούνται από ηλεκτρομαγνήτες υπεραγωγών που λειτουργούν μόνο σε θερμοκρασίες κοντά στο απόλυτο μηδέν.

και να αναλύσουν πλήρως τις τροχιές και τις ιδιότητες των σωματιδίων αυτών. Οι κυριότεροι ανιχνευτές στον LHC είναι οι: ATLAS, CMS, ALICE και LHCb³.



Σχήμα 1.2.Π: Άποψη της σήραγγας του LHC όπου επιταχύνονται τα πρωτόνια [6].

Από όσα έχουν αναφερθεί μέχρι τώρα, μπορεί να αντιληφθεί κανείς τη σημασία του *ρυθμού* με τον οποίο τα σωματίδια συγκρούονται μεταξύ τους στα πειράματα Φυσικής Υψηλών Ενεργειών. Στην προσπάθεια βελτίωσης των πειραμάτων για εξαγωγή καλύτερων αποτελεσμάτων σε όλο και υψηλότερες ενέργειες, διαπιστώθηκε ότι όσο ψηλότερος ο ρυθμός αλληλεπιδράσεων ϕ (reaction rate), τόσο περισσότερα γεγονότα ανιχνεύονται με αποτέλεσμα να εξάγονται και περισσότερες πληροφορίες από το πείραμα. Ξεκινώντας από τα απλά, δηλαδή από τα πειράματα σταθερού στόχου, ο ρυθμός αλληλεπίδρασης εξαρτάται από το ρυθμό πρόσπτωσης n των σωματιδίων της δέσμης πάνω στο στόχο, την ενεργό διατομή σ της υπό μελέτη αντίδρασης⁴, και το πάχος d του στόχου [3]. Συνδυάζοντας τα μεγέθη αυτά, ορίζεται το μέγεθος της *Φωτεινότητας* \mathcal{L} (luminosity):

$$\phi = \sigma \cdot \mathcal{L} \quad (1.2.1)$$

³Οι ανιχνευτές γενικού ενδιαφέροντος, ATLAS και CMS, ήταν εκείνοι που ανίχνευσαν το 2012 το μποζόνιο Higgs.

⁴Η ενεργός διατομή έχει μονάδες επιφάνειας και εκφράζει ουσιαστικά την πιθανότητα πραγμάτωσης μίας αντίδρασης.

Στα σύγχρονα πειράματα Υψηλών Ενεργειών όπου δεν προτιμάται η μέθοδος σταθερών στόχων, αφού με αλληλοσυγκρουόμενες δέσμες η μέγιστη ενέργεια που επιτυγχάνεται είναι σημαντικά μεγαλύτερη, η φωτεινότητα (και άρα ο ρυθμός αλληλεπιδράσεων που είναι ανάλογος της φωτεινότητας) ορίζεται περισσότερο περίπλοκα. Πλέον η φωτεινότητα αντιπροσωπεύει ένα μέτρο του πλήθους των σωματιδίων ανά cm^2 και s . Αν N_1 και N_2 το πλήθος των σωματιδίων της κάθε δέσμης, και σ_x, σ_y η εγκάρσια και διαμήκης διατομή των δεσμών, τότε η φωτεινότητα \mathcal{L} σχετίζεται με τις παραμέτρους αυτές μέσω του τύπου:

$$\mathcal{L} \propto \frac{N_1 N_2}{\sigma_x \sigma_y} \quad (1.2.2)$$

Έτσι λοιπόν, το πλήθος των αλληλεπιδράσεων N_{exp} που αναμένεται στο πείραμα, ορίζεται από τη σχέση [2]:

$$N_{exp} = \sigma_{exp} \times \int \mathcal{L}(t) dt \quad (1.2.3)$$

Όπου η ποσότητα σ_{exp} ονομάζεται *ενεργός διατομή*, έχει μονάδες μεγέθους επιφάνειας και εκφράζει την πιθανότητα να συμβεί μία συγκεκριμένη αντίδραση.

Σχετικά με τους τύπους που μόλις αναφέρθηκαν και τη σχέση τους με τα πειράματα, αξίζει να αναφερθεί ότι είναι γενικά εύκολο να οριστεί το πλήθος των σωματιδίων της δέσμης σε ένα πείραμα. Η μεγάλη δυσκολία έγκειται στη μείωση της διατομής των δεσμών (στο focus με άλλα λόγια), ένα εγχείρημα που χαρακτηρίζεται από εγγενείς περιορισμούς, καθώς μία δέσμη που αποτελείται μόνο από πρωτόνια για παράδειγμα, αδυνατεί να μείνει συγκεντρωμένη, λόγω των τεράστιων ηλεκτρικών δυνάμεων άπωσης μεταξύ των όμοια φορτισμένων πρωτονίων, που βρίσκονται τόσο κοντά μεταξύ τους μέσα στη δέσμη. Τέτοια τεχνικά ζητήματα καλούνται να λυθούν σε κάθε πείραμα Φυσικής Υψηλών Ενεργειών, πόσο μάλλον στον LHC, όπου οι ενέργειες και η φωτεινότητα είναι τάξεις μεγέθους μεγαλύτερες σε σχέση με παλιότερα εγχειρήματα.

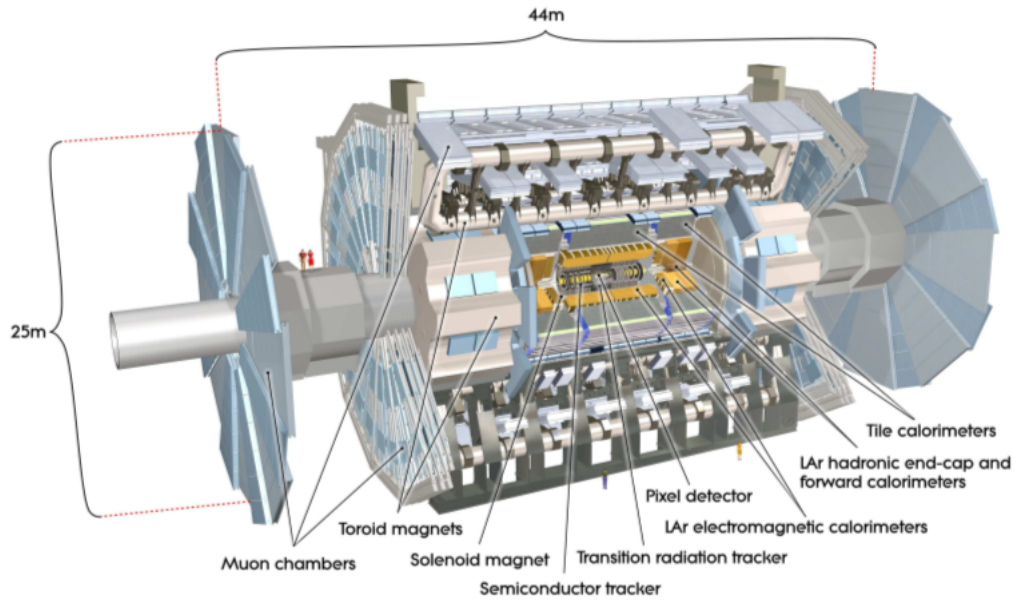
Πιο συγκεκριμένα [9], οι ανιχνευτές ATLAS και CMS, είναι σχεδιασμένοι για να ανιχνεύουν σε φωτεινότητες της τάξης του $\mathcal{L} = 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, ενώ το πείραμα LHCb λειτουργεί σε χαμηλότερες φωτεινότητες ($\mathcal{L} = 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$). Από τη κατα-

σκευή του μέχρι σήμερα, ο LHC παραμένει (και θα παραμείνει για πολλά χρόνια) η μεγαλύτερη και πιο περίπλοκη πειραματική διάταξη που κατασκευάστηκε ποτέ. Ξεκίνησε το 2011-2012 στα 3.5 TeV ενέργεια ανά δέσμη και ανέβηκε στα 4 TeV το 2012, ενώ έχει προσφέρει συνολική φωτεινότητα της τάξης $\int \mathcal{L} dt = 28.26 \text{ fb}^{-1}$ στους ανιχνευτές ATLAS και CMS. Στη συνέχεια, θα εξεταστεί η δομή του ανιχνευτή ATLAS, ο οποίος είναι και ο ανιχνευτής που πραγματεύεται (έμμεσα) η παρούσα εργασία, καθώς οι αναβαθμίσεις που σχεδιάζονται για το μέλλον χρήζουν νέων μελετών στους ανιχνευτές και στα ηλεκτρονικά τους συστήματα, προκειμένου να αντεπεξέλθουν στις νέες απαιτήσεις των επικείμενων πειραμάτων.

1.3 Ο Ανιχνευτής ATLAS

Ο ανιχνευτής ATLAS (A Toroidal LHC ApparatuS), είναι ένας ανιχνευτής γενικής χρήσης, σχεδιασμένος να καλύψει τις ανάγκες πολλών πειραμάτων του LHC, όπως την αναζήτηση του σωματιδίου Higgs, τη μελέτη της υπερσυμμετρίας, την απάντηση ερωτημάτων σχετικά με τη σκοτεινή ύλη και τυχόν ύπαρξη επιπλέον διαστάσεων στο Σύμπαν. Το γενικό του σχήμα είναι κυλινδρικό, με μήκος 45 m , και διάμετρο 25 m . Ζυγίζει περίπου 7000 τόννους, και είναι μέχρι σήμερα ο μεγαλύτερος ανιχνευτής σε όγκο που κατασκευάστηκε ποτέ. Από το 2012, 3000 επιστήμονες από 38 χώρες συνεργάζονται για την υποστήριξη και αναβάθμιση του ανιχνευτή [6]. Στο σχήμα 1.3.1 απεικονίζεται σε τρισδιάστατο μοντέλο μία γενική άποψη του ανιχνευτή.

Οι δέσμες των σωματιδίων περνούν από τον νοητό άξονα κυλινδρικής συμμετρίας του ανιχνευτή (από το κέντρο των κυκλικών "καπακιών") και συγκρούονται στο κέντρο του, παράγοντας νέα σωματίδια. Τα διαφορετικά υποσυστήματα του ATLAS, τοποθετημένα σε στρώματα, καταγράφουν την τροχιά, την ορμή και την ενέργεια των σωματιδίων, ταυτοποιώντας τα πλήρως. Ένας ισχυρός μαγνήτης κάμπει τις τροχιές των φορτισμένων σωματιδίων, επιτρέποντας τη μέτρηση της ορμής τους, ενώ τα δεδομένα που συλλέγονται από τα ηλεκτρονικά συστήματα του ανιχνευτή, υποβάλλονται σε συνεχή ανάλυση ώστε να εξαχθούν οι επιθυμητές πληροφορίες από αυτά. Η όλη διάταξη θυμίζει τη μορφή ενός βαρελιού, και για το λόγο αυτό είναι σύνηθες να αναφέρονται τα διάφορα τμήματα του ανιχνευτή ως "περίβλημα του βαρελιού", ή απλά "βαρέλι" (barrel region), και ως "καπάκια" (end-cap region)

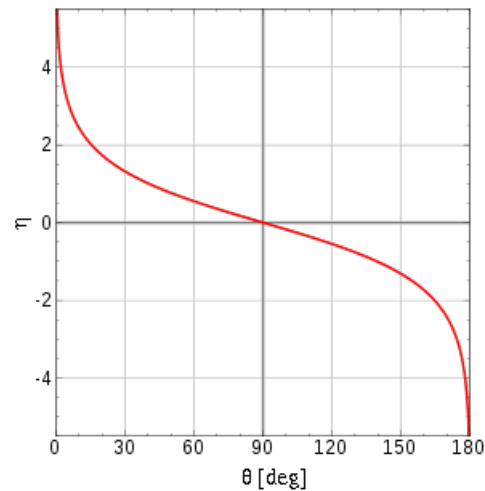


Σχήμα 1.3.I: Τομή του ανιχνευτή ATLAS [6].

Προκειμένου να γίνουν σαφείς οι λεπτομέρειες για τα ανιχνευτικά συστήματα του ATLAS που θα αναφερθούν στη συνέχεια, είναι καλό σε αυτό το σημείο να οριστεί η έννοια της *pseudorapidity* (η). Αν η γωνία σε σχέση με τον άξονα της δέσμης ισούται με θ , τότε η *pseudorapidity* ορίζεται ως [7]:

$$\eta \equiv -\ln \left[\tan \left(\frac{\theta}{2} \right) \right] \quad (1.3.1)$$

Ο ορισμός της ποσότητας η είναι πολύ σημαντικός, καθώς γίνεται έτσι εύκολα αντιληπτή η γεωμετρία που καλύπτει το κάθε ανιχνευτικό υποσύστημα του ανιχνευτή. Πιο συγκεκριμένα, αναφέρεται πιο συχνά η ποσότητα $|\eta|$, που καλύπτει και τα δύο νοητά ημιεπίπεδα που σχηματίζει η ευθεία της δέσμης. Ένα προκύπτον σωματίδιο λοιπόν με μεγάλη τιμή *pseudorapidity*, πιθανόν να αποδράσει από τον ανιχνευτή, αφού θα κινηθεί σχεδόν συγγραμικά με τη δέσμη, ενώ μικρότερες τιμές του $|\eta|$ υπονοούν γωνίες διαφυγής σε σχέση με τη δέσμη της τάξης των $90^\circ - 45^\circ$. Έτσι, είναι φανερό ότι τα καπάκια του ανιχνευτή αντιστοιχούν σε μεγάλες τιμές του $|\eta|$, ενώ το κυλινδρικό περίβλημα του βαρελιού σε μικρότερες.



Σχήμα 1.3.Π: Γράφημα της pseudorapidity σε σχέση με τη γωνία από τον άξονα της δέσμης.

Τα κυριότερα υποσυστήματα του ATLAS είναι τα εξής [9]:

- Το σύστημα μαγνητών
- Ο εσωτερικός ανιχνευτής
- Τα θερμιδόμετρα (ή καλορίμετρα)
- Το μιονικό φασματόμετρο (ή σπεκτρόμετρο)

Ο μαγνήτης είναι ουσιαστικά ένα λεπτό υπεραγώγιμο σωληνοειδές, που περικλείει το κενό ανάμεσα από το σημείο αλληλεπίδρασης/σύγκρουσης σε σχέση με τον εσωτερικό ανιχνευτή. Εκτός από τον εσωτερικό μαγνήτη, τρία μεγαλύτερα σωληνοειδή βρίσκονται τοποθετημένα γύρω από τα καλορίμετρα. Ο εσωτερικός μαγνήτης παράγει ένα μαγνητικό πεδίο της τάξης των 2 T , και επιτρέπει στον εσωτερικό ανιχνευτή (με μήκος 6 m και διάμετρο 2 m) που καλύπτει το εύρος $|\eta| < 2.5$, να αναλύσει τις τροχιές των σωματιδίων που μόλις έχουν δραπετεύσει από τα σημεία σύγκρουσης. Περίπου 1000 σωματίδια προκύπτουν από το σημείο σύγκρουσης κάθε 25 ns . Παρά τη μεγάλη πυκνότητα τροχιών που δημιουργούνται στον ανιχνευτή, επιτυγχάνεται βέλτιστος προσδιορισμός ορμών και vertex⁵, από τα συστήματα των Pixel Detectors και SemiConductor Trackers (SCTs). Γύρω από αυτούς, βρίσκονται

⁵Το "σημείο" αλληλεπίδρασης ονομάζεται vertex

Transition Radiation Trackers (TRT), που αναλύουν την ακτινοβολία μετάπτωσης⁶.

Γύρω από τον εσωτερικό ανιχνευτή, βρίσκονται τα καλορίμετρα, ή αλλιώς θερμιδόμετρα. Τα καλορίμετρα είναι συσκευές σχεδιασμένες με τέτοιο τρόπο ώστε να απορροφούν πλήρως κάποιου τύπου σωματίδια που προσπίπτουν πάνω τους. Έτσι όλη η ενέργεια των σωματιδίων εναποτίθεται στο εσωτερικού του θερμιδόμετρου. Συνήθως, τα καλορίμετρα είναι είτε αδρονικά (ανιχνεύουν δηλαδή μεσόνια ή βαρυόνια), είτε ηλεκτρομαγνητικά (ανιχνεύουν ηλεκτρόνια, ποζιτρόνια και φωτόνια). Το αδρονικό καλορίμετρο χωρίζεται σε τρία μέρη, ανάλογα το εύρος της pseudorapidity που καλύπτει. Υπάρχει το τμήμα που καλύπτει το βαρέλι ($|\eta| < 1.7$), τα καπάκια του ($1.5 < |\eta| < 3.2$), και περιοχές ακόμα περισσότερο κοντά στη δέσμη ($3.1 < |\eta| < 3.9$) [9]. Κάθε τμήμα έχει και λίγο διαφορετική κατασκευή: για παράδειγμα το καλορίμετρο που περιβάλλει το βαρέλι είναι φτιαγμένο από ατσάλι (που λειτουργεί ως απορροφητής) και σπινθηριστές (scintillators) που παράγουν τα σήματα ανίχνευσης [7]. Τα υπόλοιπα αδρονικά καλορίμετρα είναι κατασκευασμένα από Υγρό-Αργό (Liquid-Argon (LAr)). Το ηλεκτρομαγνητικό καλορίμετρο από την άλλη, αποτελείται από δύο ομόκεντρους κυκλικούς δίσκους στα καπάκια, έναν εξωτερικό και έναν εσωτερικό που καλύπτουν $1.375 < |\eta| < 2.5$ και $2.5 < |\eta| < 3.2$ αντίστοιχα. Το ηλεκτρομαγνητικό καλορίμετρο περικλείει το βαρέλι και καλύπτει εύρος $|\eta| < 1.475$. Οι συσκευές αυτές είναι κατασκευασμένες από μολύβδινες πλάκες που απορροφούν τα σωματίδια, από LAr, και από ηλεκτρόδια τύπου Krypton σε σχήμα ακκονρτεόν υπεύθυνα για τη συλλογή των ηλεκτρικών σημάτων [7, 9].

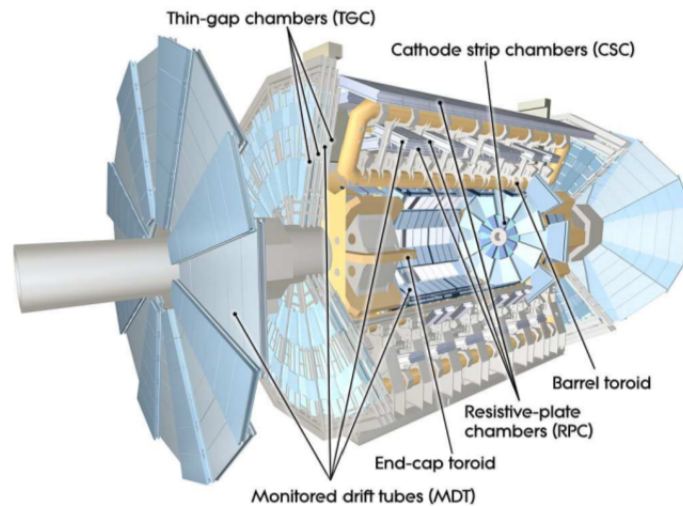
Τα σωματίδια που δεν έχουν απορροφηθεί από τα δύο μεγάλα στρώματα ανιχνευτών που αναφέρθηκαν παραπάνω, είναι συνήθως μόνια (μ), τα οποία ανήκουν στην οικογένεια των λεπτονίων δεύτερης γενιάς. Είναι είτε θετικά είτε αρνητικά φορτισμένα (τα αντιμόνια είναι θετικά φορτισμένα) και είναι αρκετά βαρύτερα από τα συγγενικά τους ηλεκτρόνια⁷. Λόγω της διεισδυτικότητας που τα χαρακτηρίζει, γενικά σε ανιχνευτικές διατάξεις, τα μέρη εκείνα που είναι υπεύθυνα για την ταυτοποίησή τους, τοποθετούνται στα εξωτερικά τμήματα του ανιχνευτή.

Το *Μιονικό Σπεκτροόμετρο* (στο οποίο θα δοθεί και περισσότερη έμφαση καθώς η αναβάθμιση την οποία πραγματεύεται η παρούσα εργασία αφορά αυτό το κομμάτι

⁶Η ακτινοβολία που εκπέμπεται από ένα σωματίδιο όταν αυτό αλλάζει μέσα διάδοσης τα οποία έχουν διαφορετικές διηλεκτρικές ιδιότητες μεταξύ τους [3].

⁷Τα μόνια (μ) έχουν μάζα περίπου $106 \text{ MeV}/c^2$, ενώ τα ηλεκτρόνια (e^-) $511 \text{ keV}/c^2$ [1].

του ATLAS) περιβάλλει τα καλορίμετρα και ορίζει ουσιαστικά το μέγεθος ολόκληρου του ανιχνευτή. Τα συστήματα του μονικού φασματομέτρου, περιλαμβάνουν ανώτερες τεχνολογίες για το triggering⁸ και για την ανακατασκευή των τροχιών των μιονίων (tracking)[9]. Στο Σχήμα 1.3.III είναι φανερή η διάταξη του συστήματος εντοπισμού των μιονίων του ATLAS.



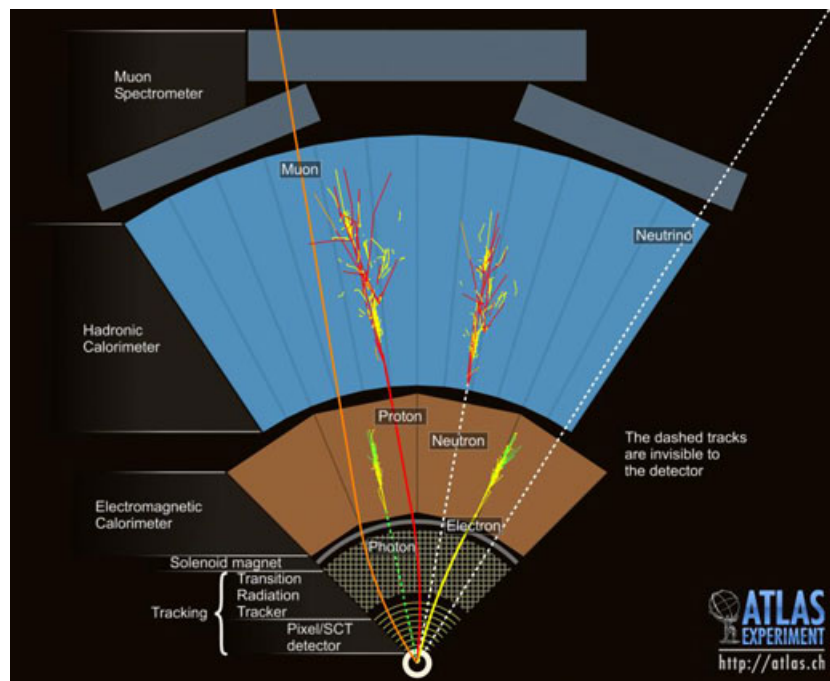
Σχήμα 1.3.III: Άποψη του μονικού φασματομέτρου του ATLAS [7].

Η λειτουργία του φασματομέτρου μιονίων βασίζεται στην κάμψη των τροχιών των μιονίων λόγω του μαγνητικού πεδίου που δημιουργούν οι υπεραγώγιοι τοροειδείς μαγνήτες μέσα στον ανιχνευτή. Στο εύρος $|\eta| < 1.4$, η κάμψη των τροχιών παρέχεται από τον τοροειδή μαγνήτη που περιβάλλει το βαρέλι, ενώ για το εύρος ($1.6 < |\eta| < 2.7$) οι τροχιές κάμπτονται από δύο μικρότερους μαγνήτες που βρίσκονται στα καπάκια του ανιχνευτή. Το προκύπτον μαγνητικό πεδίο είναι ως επί το πλείστον κάθετο στις τροχιές των μιονίων [7], όποια και να είναι η γωνία διαφυγής τους σε σχέση με τη δέσμη. Η ανίχνευση στα μικρά $|\eta|$, γίνεται από τρεις θαλάμους ανίχνευσης (chambers), που είναι ταξινομημένοι σε τρία κυλινδρικά στρώματα γύρω από τον άξονα της δέσμης. Στην περιοχή με μεγάλα $|\eta|$, οι θάλαμοι καλύπτουν τα καπάκια πλήρως, είναι κάθετοι στον άξονα της δέσμης, και είναι και αυτοί οργανωμένοι σε τρία στρώματα.

⁸“Σκανδαλισμός” στα Ελληνικά. Πρόκειται για ένα σήμα που υποδεικνύει ότι ένα σημαντικό γεγονός έχει συμβεί και πρέπει να καταγραφεί από τα ηλεκτρονικά συστήματα του ανιχνευτή.

Σε όλα τα εύρη των $|\eta|$, η τροχιά των μιονίων ανακατασκευάζεται από τα σήματα που αφήνουν τα μίονια σε μία σειρά από Monitored Drift Tubes (MDTs)⁹. Στα καπάκια, εκτός από MDTs, υπάρχουν και Cathode Strip Chambers (CSCs), που είναι πολυσυρματικοί αναλογικοί ανιχνευτές (multiwire proportional chambers), σχεδιασμένοι να υποβοηθούν το tracking των μιονίων, καθώς η περιοχή εκείνη χαρακτηρίζεται από αυξημένο ρυθμό γεγονότων και ακτινοβολία υποβάθρου [7]. Η αρχή λειτουργίας τους είναι συγγενική με εκείνη των MDTs, όμως χαρακτηρίζεται από εντελώς διαφορετική γεωμετρία και μείγμα αερίου. Το σύστημα του triggering από την άλλη, αποτελείται από Resistive Plate Chambers (RPCs) που περιβάλλουν το βαρέλι, και από Thin Gap Chambers (TGCs), στα καπάκια.

Έχοντας πλέον μία γενική εικόνα του τρόπου λειτουργίας του ανιχνευτή ATLAS, παρατίθεται και μία εικόνα που αναπαριστά τις τροχιές των σωματιδίων ενός τυπικού γεγονότος (Σχήμα 1.3.IV).

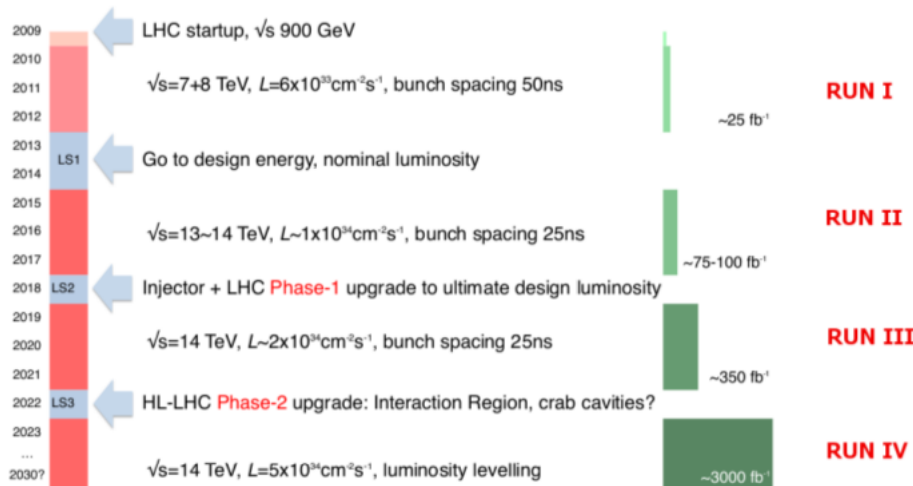


Σχήμα 1.3.IV: Αναπαράσταση ενός γεγονότος στον ATLAS [9].

⁹Η αρχή λειτουργίας τους βασίζεται σε αυτή των ανιχνευτών αερίου, οι οποίοι μελετώνται στο επόμενο Κεφάλαιο

1.4 Η Αναβάθμιση του New Small Wheel

Η κατασκευή του LHC ολοκληρώθηκε με επιτυχία το 2008 όπου η μέγιστη ενέργεια του επιταχυντή έφτασε περίπου τα 1 TeV , και μέχρι το πρώτο κλείσιμο το 2010, προσέφερε συνολικά $\int \mathcal{L} = 29 \text{ fb}^{-1}$. Μετά το πρώτη μεγάλη παύση λειτουργίας (Long Shutdown, LS1) το 2013-2014 για αναβαθμίσεις, η ενέργεια του επιταχυντή αυξήθηκε στα 7 TeV ανά δέσμη, με τη φωτεινότητα να φτάνει στα $\mathcal{L} = 1 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$. Το 2018 έχει προγραμματιστεί η δεύτερη μεγάλη παύση (LS2), όπου η φωτεινότητα θα βελτιωθεί ακόμα περισσότερο ($\mathcal{L} = 2 - 3 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$), η ενέργεια θα φτάσει τα $13 - 14 \text{ TeV}$, και ο ανιχνευτής ATLAS αναμένεται να συλλέγει περίπου 100 fb^{-1} ολοκληρωμένη φωτεινότητα ανά έτος [8]. Το τελικό στάδιο της αναβάθμισης της δέσμης αναμένεται το 2022, (LS3), όπου η φωτεινότητα θα αγγίξει τα $\mathcal{L} = 5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$. Στο Σχήμα 1.4.I δίνεται ένα χρονοδιάγραμμα των αναβαθμίσεων στον LHC.



Σχήμα 1.4.I: Χρονοδιάγραμμα των αναβαθμίσεων του LHC [8].

Για τον ανιχνευτή ATLAS, οι διαδοχικές αυξήσεις στη φωτεινότητα συνεπάγονται και αύξηση στο ρυθμό με τον οποίο τα σωματίδια θα τον διαπερνούν. Προκειμένου να εξαχθούν όλα τα επιθυμητά αποτελέσματα από την επικείμενη αναβάθμιση του LHC το 2018, ο ανιχνευτής ATLAS πρέπει να αναβαθμιστεί και αυτός. Πιο συγκεκριμένα, ιδιαίτερη προσοχή έχει δοθεί στη βελτίωση του Level-1 triggering και του

tracking στο σύστημα μιονίων, καθώς [8]:

- Η αποδοτικότητα των ανιχνευτών που είναι υπεύθυνοι για την ανακατασκευή των τροχιών των μιονίων, κυρίως σε τμήματα με μεγάλα $|\eta|$ στα καπάκια του ανιχνευτή, προβλέπεται ότι θα υποβαθμιστεί με αύξηση της φωτεινότητας, βάσει μέχρι τώρα μετρήσεων. Μεγαλύτερο πρόβλημα θα παρουσιαστεί κυρίως στο εσωτερικό τμήμα των δίσκων (Small Wheels), που έχουν απόσταση μέχρι 7 m από τον άξονα της δέσμης.
- Μια ανάλυση των δεδομένων που πραγματοποιήθηκε το 2012 και συγκέντρωσε τα δεδομένα του triggering από την έναρξη λειτουργίας του LHC, κατέδειξε ότι περίπου το 90 % των μιονικών triggers στα καπάκια του ATLAS ήταν ψεύτικα. Αυτό συνέβαινε καθώς πρωτόνια χαμηλής ενέργειας, που παράγονταν από δευτερεύουσες αντιδράσεις στα υλικά του ανιχνευτή πριν το μιονικό σπεκτρομέτρο στα καπάκια, εισέρχονταν στους ανιχνευτές TGCs (που είναι υπεύθυνοι για το muon triggering), με τέτοιες γωνίες και ορμές που αναγνωρίζονταν από αυτούς ως μόνια. Έτσι υπολογίστηκε πως ο ρυθμός του triggering στα καπάκια είναι μέχρι και 10 φορές μεγαλύτερος από εκείνος στο κυλινδρικό περίβλημα του βαρελιού όπου δεν παράγονται πρωτόνια χαμηλής ενέργειας που φέρνουν το σύστημα σε σύγχυση.

Με την αύξηση της ροής των σωματιδίων (περίπου 15 kHz/cm^2 για $|\eta|=2.7$ [9]) μετά την πρώτη αναβάθμιση, τα προαναφερθέντα προβλήματα αναμένεται να οξυνθούν σημαντικά. Ως εκ τούτου, η ομάδα του ATLAS πρότεινε την πλήρη αντικατάσταση του προβληματικού τμήματος στα καπάκια (του εσωτερικού δίσκου με μεγάλα $|\eta|$ που είναι πιο κοντά στη δέσμη δηλαδή), από ένα νέο ανιχνευτικό σύστημα, το *New Small Wheel* (NSW). Το νέο τμήμα θα καλύπτει ένα εύρος $1.3 < |\eta| < 2.7$, και με τους νέους ανιχνευτές που θα είναι εγκατεστημένοι πάνω του, προβλέπεται ότι θα προσφέρει άριστο χωρικό και χρονικό προσδιορισμό των τροχιών (spatial and time resolution) σε πραγματικό χρόνο, καθώς και πιο συνεπές Level-1 triggering, αφού θα μπορεί πλέον να διακρίνει τα μόνια χαμηλής ενέργειας από τα πρωτόνια που παράγονται από δευτερεύουσες αντιδράσεις και αποτελούν για το σύστημα μιονίων ακτινοβολία υποβάθρου [8, 9]. Ένα νέο σύστημα triggering που θα συνδυάζει δεδομένα από το ηλεκτρομαγνητικό καλορίμετρο μαζί με δεδομένα από το NSW, θα μειώσει την αναμενόμενη (για $\mathcal{L} = 3 \times 10^{34}\text{ cm}^{-2}\text{s}^{-1}$) συχνότητα μιονικών Level-1

trigger, από την προβλεπόμενη τιμή των 100 kHz , που θα επικρατούσε αν η ανιχνευτική διάταξη παρέμενε ως έχει, σε πιο διαχειρίσιμα ποσά της τάξης των 20 Hz , με την εγκατάσταση του NSW [9].

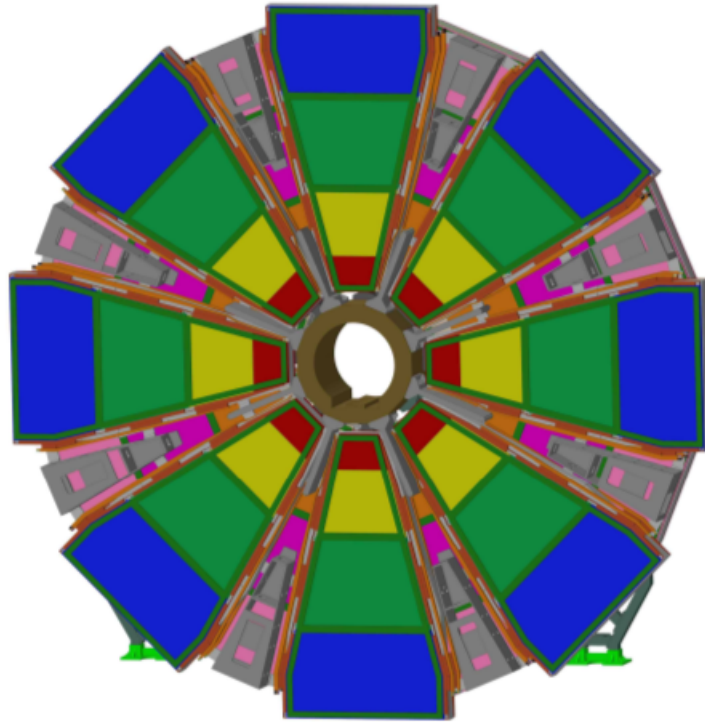
Οι ανιχνευτές που θα αποτελούν το NSW θα είναι δύο ειδών και θα προέρχονται από την γενικότερη κατηγορία των ανιχνευτών αερίου (gaseous detectors). Ο πρώτος είναι ένας πολυσυρματικός θάλαμος (multiwire chamber), που ονομάζεται small strip Thin Gap Chamber (sTGC), ενώ ο δεύτερος ανήκει στην οικογένεια των Micro-Pattern Gaseous Detectors, και ονομάζεται Micromesh Gaseous Structure (MicroMegas). Οι sTGC θα χρησιμοποιούνται περισσότερο για το triggering, ενώ οι ανιχνευτές MicroMegas (MM) θα χρησιμεύσουν κυρίως στην ανακατασκευή τροχιών (tracking). Οι δύο αυτοί τύπου ανιχνευτών θα καλύπτουν συνολικά μία έκταση 1200 m^2 ο καθένας, και θα καταλαμβάνουν πλέον όλη την εσωτερική περιοχή στα καπάκια του ανιχνευτή. Θα είναι τοποθετημένοι ακτινικά σε οκτώ πλήρως αλληλοκαλυπτόμενα επίπεδα, όπου οι MM θα καταλαμβάνουν το εσωτερικό του δίσκου, ενώ οι sTGC θα βρίσκονται στα εξωτερικά μέρη. Οι ανιχνευτές θα είναι τοποθετημένοι με τέτοιο τρόπο ώστε να μην υπάρχουν νεκρές ζώνες (dead regions)¹⁰, αλλά μόνο περιοχές ελαττωμένης ανιχνευτικής απόδοσης [8]. Μία γενική άποψη από τη μορφή του NSW μπορεί να δει κανείς στο Σχήμα 1.4.Π

Σύνοψη

Σε αυτό το Κεφάλαιο, μετά από μία σύντομη εισαγωγή στη Φυσική των Στοιχειωδών Σωματιδίων, πραγματοποιήθηκε μία γενική επισκόπηση του Μεγάλου Επιταχυντή Αδρονίων (LHC) που βρίσκεται στο CERN, και του ανιχνευτή ATLAS στη σχεδίαση του οποίου συμμετέχει ενεργά το Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ). Ο ανιχνευτής αυτός, είναι ο μεγαλύτερος σε όγκο ανιχνευτής Φυσικής Υψηλών Ενεργειών που έχει κατασκευαστεί, όμως επιδέχεται βελτιώσεων προκειμένου να ανταπεξέλθει στις αυξημένες απαιτήσεις που θα προκύψουν μετά από τις αναβαθμίσεις του LHC. Στα επόμενα δύο Κεφάλαια, θα μελετηθούν με περισσότερες λεπτομέρειες ο ανιχνευτής MicroMegas και το περίπλοκο σύστημα ηλεκτρονικών του New Small Wheel, ώστε να είναι ομαλή η μετάβαση στα μετέπειτα μέρη της εργασίας αυτής, στα οποία θα παρουσιαστεί η συμβολή της ομάδας Πειραματικής Φυσικής Υψηλών

¹⁰Πρόκειται για περιοχές που δεν καλύπτονται από ανιχνευτές και αν περάσει κάποιο σωματίδιο από εκεί, θα μείνει απαρατήρητο

Ενεργειών του ΕΜΠ στο σχεδιασμό των ηλεκτρονικών συστημάτων της αναβάθμισης του New Small Wheel του ATLAS.



Σχήμα 1.4.Π: Σχηματική αναπαράσταση του NSW. Οι ανιχνευτές MM και sTGC είναι διατεταγμένοι σε αλληπάλγηλα στρώματα σε κάθε "φέτα" (wedge)[9].

2

Ο Ανιχνευτής MicroMegas

Στο παρόν κεφάλαιο θα γίνει μία λεπτομερέστερη περιγραφή των αρχών λειτουργίας του ανιχνευτή MicroMegas (MM). Ο MicroMegas θα είναι ένα από τα κύρια λειτουργικά στοιχεία του New Small Wheel, το οποίο θα τοποθετηθεί στο πείραμα ATLAS του CERN, στα πλαίσια μίας γενικότερης αναβάθμισης του LHC. Η αναβάθμιση αυτή θα περατωθεί το 2018 και το ΕΜΠ έχει από το ξεκίνημα του σχεδιασμού της σημαντική συμβολή στην έρευνα και στην ανάπτυξη του ανιχνευτή MicroMegas.

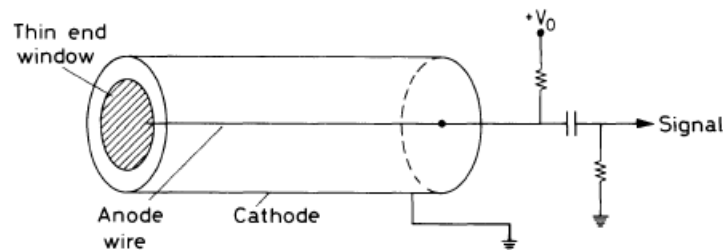
2.1 Εισαγωγή - Ανιχνευτές Αερίων

Τα σωματίδια ανιχνεύονται μέσω της αλληλεπίδρασής τους με την ύλη. Κάθε ανιχνευτική διάταξη έχει την ίδια γενική φιλοσοφία στην κατασκευή της: Η ακτινοβολία αλληλεπιδρά με το μέσο ανίχνευσης (που είναι πολλές φορές αέριας μορφής), και το προϊόν της αλληλεπίδρασης αυτής μετατρέπεται σε ηλεκτρικό σήμα [9]. Στη συνέχεια το σήμα αυτό αναλύεται από τα συστήματα ηλεκτρονικών σε πρώτη φάση, και σε δεύτερη φάση από εξειδικευμένο λογισμικό, ώστε να γίνει πλήρης ταυτοποίηση των σωματιδίων και των τροχιών τους μέσα στον ανιχνευτή.

Όταν ένα φορτισμένο σωματίδιο (π.χ. μόνιο), περάσει μέσα από ένα αέριο μείγμα, τότε το μόνιο θα χάνει συνεχώς ενέργεια μέσω της αλληλεπίδρασής του με τα άτομα του αερίου. Αυτό που συμβαίνει συνήθως, είναι να επέρχεται *ιονισμός*. Κατά τον ιονισμό, ένα προσπίπτον σωματίδιο προσδίδει ενέργεια στα ηλεκτρόνια ενός ατόμου, απελευθερώνοντας τα από το δυναμικό του πυρήνα, με αποτέλεσμα να δημιουργεί-

ται ένα ελεύθερο ηλεκτρόνιο και ένα θετικά φορτισμένο ιόν. Πέρα από το αέριο που το πληροί, ο ανιχνευτής διαθέτει και αντίθετα φορτισμένα μέρη, τα οποία μπορεί να είναι δύο πλάκες, εν είδει πυκνωτή, ή ένα λεπτό σύρμα (άνοδος) σε συνδυασμό με κάποια γειωμένη πλάκα. Αυτή η διάταξη δημιουργεί ένα ηλεκτρικό πεδίο, το οποίο ευνοεί περαιτέρω ιονισμούς μέσα στο αέριο, καθώς τα ηλεκτρόνια που ιονίζονται αρχικά από την προσπίπτουσα ακτινοβολία επιταχύνονται από το εφαρμοζόμενο ηλεκτρικό πεδίο, με αποτέλεσμα να σκεδάζονται πάνω σε άλλα άτομα, ιονίζοντάς τα και αυτά. Έτσι από έναν αρχικό ιονισμό, προκύπτει μία πληθώρα περισσοτέρων, με αποτέλεσμα να πολλαπλασιάζονται τα ηλεκτρόνια και τα θετικά ιόντα. Το φαινόμενο αυτό ονομάζεται *φαινόμενο χιονοστιβάδας* (avalanche effect) ή φαινόμενο Townsend. Τελικά, τα αρνητικά φορτισμένα ηλεκτρόνια και τα θετικά ιόντα καταλήγουν στα αντίθετα φορτισμένα κομμάτια του ανιχνευτή, παράγοντας ένα ηλεκτρικό σήμα.

Ο πρώτος ανιχνευτής που βασιζόταν σε αυτή την αρχή λειτουργίας ήταν ο μετρητής Geiger-Müller [3, 4], που κατασκευάστηκε αρχικά από τους Geiger και Rutherford το 1908 [9]. Στο Σχήμα 2.1.1 απεικονίζεται η πιο απλή μορφή του ανιχνευτή αυτού.

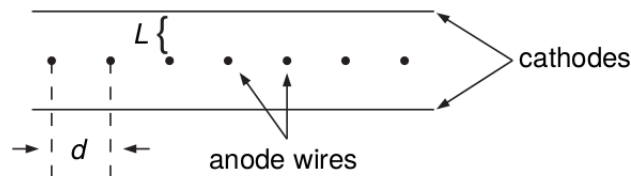


Σχήμα 2.1.1: Ένας μετρητής Geiger. [4]

Ο σωλήνας του ανιχνευτή πληρώνεται με ένα αέριο, συνήθως ευγενές, και στο κεντρικό σύρμα εφαρμόζεται μία θετική τάση $+V_0$, σε σχέση με τα τοιχώματα, τα οποία είναι γειωμένα. Έτσι, δημιουργείται ένα ηλεκτρικό πεδίο η ένταση του οποίου είναι ανάλογη της τάσης και αντιστρόφως ανάλογη της απόστασης από την άνοδο, ενώ επίσης εξαρτάται και από το πάχος του σύρματος της ανόδου. Όταν ιονίζουσα ακτινοβολία διαπεράσει το σωλήνα, λόγω ιονισμού θα δημιουργηθούν ζεύγη ηλεκτρονίων και ιόντων, τα οποία θα διαχωριστούν, και τα ηλεκτρόνια θα κινηθούν γρήγορα προς την άνοδο, ενώ τα ιόντα θα κινηθούν πολύ αργά προς τα τοιχώ-

ματα¹. Υπάρχει μία συγκεκριμένη περιοχή εφαρμοζόμενης τάσης που πολλαπλασιάζει το σήμα αυτό, προκαλώντας φαινόμενο χιονοστιβάδας, και δεν αφήνει τα ηλεκτρόνια να επανασυνδεθούν με τα ιόντα, ούτε προκαλεί ανεξέλεγκτα φαινόμενα ιονισμών που θα δημιουργούσαν ηλεκτρικές εκκενώσεις στον ανιχνευτή καταστρέφοντάς τον. Τελικά, το πολλαπλασιασμένο ηλεκτρικό σήμα που συλλέγεται με την άφιξη των ηλεκτρονίων/ιόντων στα αντίστοιχα μέρη του σωλήνα επεξεργάζεται από τα ηλεκτρονικά, σηματοδοτώντας έτσι ότι ακτινοβολία διαπέρασε τον ανιχνευτή. Η φιλοσοφία αυτή, διέπει όλους τους ανιχνευτές αερίων μέχρι σήμερα.

Βάσει της κατασκευής των απεριθμητών Geiger–Müller, στα τέλη της δεκαετίας του '60 σχεδιάστηκε από τον G. Charpak ο πρώτος πολυσυρματικός αναλογικός θάλαμος (multiwire proportional chamber, MWPC), ο οποίος αποτελείται ουσιαστικά από δύο καθόδους σε μορφή πλάκας, με μία συστοιχία από παράλληλα μεταξύ τους σύρματα ανόδου ανάμεσα από τις δύο πλάκες. Προφανώς, η όλη διάταξη βρίσκεται μέσα σε ένα αέριο μείγμα που λειτουργεί ως το μέσο ανίχνευσης. Ένα απλό σχεδιάγραμμα τέτοιου ανιχνευτή απεικονίζεται στο Σχήμα 2.1.Π.



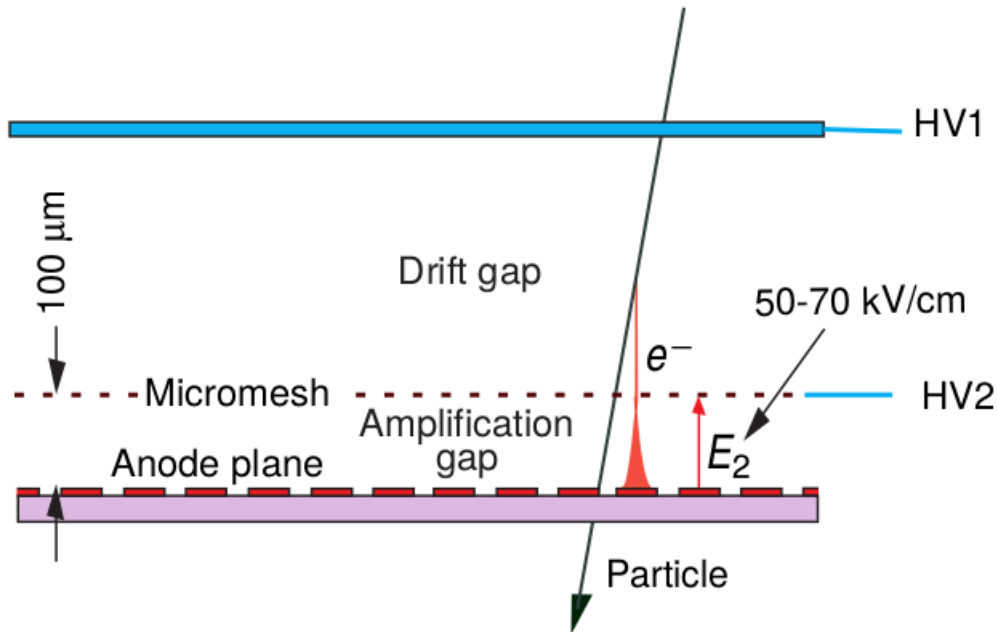
Σχήμα 2.1.Π: Ένας ανιχνευτής MWPC. [3]

Ο βελτιωμένος αυτός ανιχνευτής, εξελίχθηκε περαιτέρω με την εισαγωγή των Micro-Pattern Gaseous Detector (MPGDs) το 1988 [2]. Το κύριο χαρακτηριστικό που διαχωρίζει την οικογένεια αυτή από τους MWPCs, είναι η αντικατάσταση των συρμάτων της ανόδου από λεπτές λωρίδες (*strips*), που βρίσκονται πακτωμένες πάνω σε ένα μονωτικό (ή ημιαγώγιμο) υλικό [3]. Ο σχεδιασμός αυτός τους επιτρέπει να είναι κατά πολύ μικρότεροι σε μέγεθος από τους MWPCs, και να είναι ταυτόχρονα περισσότερο αποδοτικοί ως προς τα φορτία που δημιουργούνται κατά το πέρασμα ενός σωματιδίου από το αέριο μέσο που χρησιμοποιούν ως μέσο ανίχνευσης. Ένας τύπος MPGD, είναι και ο ανιχνευτής MicroMegas.

¹ Η διαφορά στην ταχύτητα έγκειται στη μεγάλη διαφορά μάζας μεταξύ των δύο σωματιδίων.

2.2 Ο Ανιχνευτής MicroMegas

Ο ανιχνευτής *MicroMegas* (MM) κατασκευάστηκε πρώτη φορά από τους Giomataris και Charpak το 1991 [9]. Η γενική διάταξη απεικονίζεται στο Σχήμα 2.2.I.

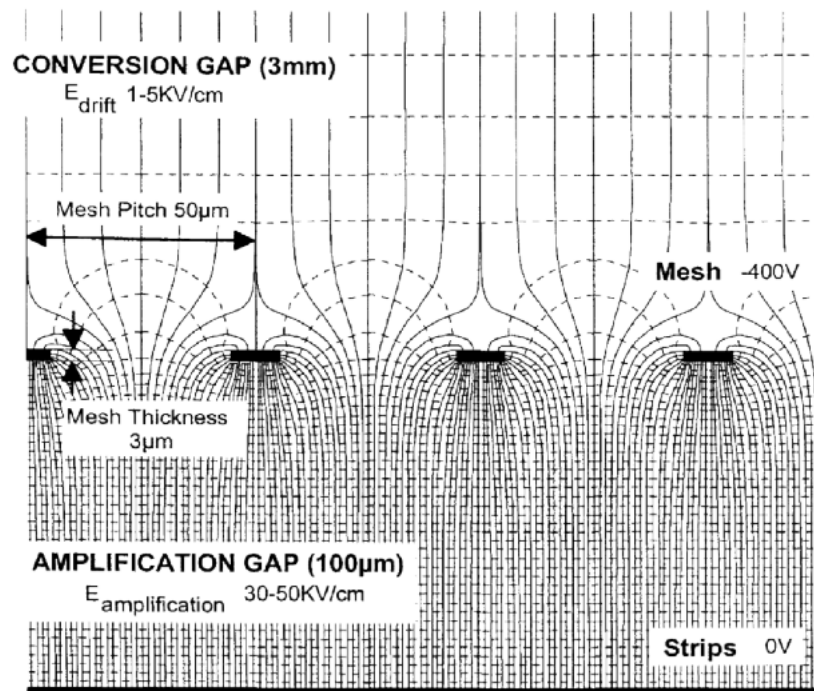


Σχήμα 2.2.I: Η γενική διάταξη ενός ανιχνευτή MM. [2]

Ένας ανιχνευτής MM, μπορεί να χωριστεί σε δύο περιοχές: Στην περιοχή του drift gap, που έχει πάχος από μερικές εκατοντάδες μm , μέχρι $3 mm$. Η παρακάτω στενή περιοχή, ονομάζεται amplification gap, και έχει πάχος που κυμαίνεται στο εύρος $25 - 150 \mu m$. Στο κατώτερο μέρος του ανιχνευτή, βρίσκονται τα *read-out strips*, που έχουν πάχος $5 \mu m$ και πλάτος $150 \mu m$. Είναι κατασκευασμένα από χαλκό και έχουν επίστρωση χρυσού. Η στενή περιοχή όπου γίνεται η ενίσχυση ορίζεται από το επίπεδο που βαίνουν τα strips, και από το επίπεδο του Micromesh, που είναι ένα μεταλλικό πλέγμα, πάχους $3 \mu m$, με ανοίγματα πλάτους $17 \mu m$, το οποίο στηρίζεται σε ένα σύστημα από μικροσκοπικά στυλίδια (quartz fibers/pillars) διαμέτρου $100 \mu m$ [9]. Η διάταξη αυτή βρίσκεται σε ένα αεροστεγές περιβάλλον πληρωμένο με αέριο αργό με μικρές προσμίξεις διοξειδίου του άνθρακα.

Εφαρμόζοντας ένα ισχυρό δυναμικό στα άνω ηλεκτρόδια (HV1 στο Σχήμα 2.2.I), και

ένα λιγότερο ισχυρό στο πλέγμα του micromesh (HV2 στο Σχήμα 2.2.I), δημιουργείται ένα ομογενές ηλεκτρικό πεδίο στο drift gap, της τάξης του $E_{drift} = 1 - 5 \text{ kV/cm}$. Στο amplification gap, υπάρχει ένα ηλεκτρικό πεδίο της τάξης του $E_{amp} = 30 - 50 \text{ kV/cm}$, το οποίο δημιουργείται γειώνοντας τα strips της ανόδου, και εφαρμόζοντας ένα αρνητικό δυναμικό στο πλέγμα. Λόγω της μικρής τους απόστασης, οι δυναμικές γραμμές που θα προκύψουν είναι πολύ πυκνές, υποδεικνύοντας ισχυρό ηλεκτρικό πεδίο, το οποίο ενισχύει σημαντικά το σήμα που θα παράξει ένα σωματίδιο που περνάει από τον ανιχνευτή.



Σχήμα 2.2.II: Χαρτογράφηση του ηλεκτρικού πεδίου σε έναν ανιχνευτή MicroMegas. [9]

Το αποτέλεσμα αυτής της διάταξης είναι να δημιουργείται ένα εξαιρετικό περιβάλλον που πολλαπλασιάζει γρήγορα τα ηλεκτρόνια που προκύπτουν από τον αρχικό ιονισμό, αφού μόλις ένα ηλεκτρόνιο εισέλθει στην περιοχή ενίσχυσης, αποκτάει μεγάλη ενέργεια γρήγορα, ευνοώντας έτσι φαινόμενα χιονοστιβάδας. Ενδεικτικά, ένα ηλεκτρόνιο που περνάει μέσα από τις τρύπες του mesh, επιταχύνεται και δημιουργεί ένα σημαντικά ενισχυμένο σήμα που φτάνει στα strips σε κλάσματα του ns [8]. Τα βραδέα θετικά ιόντα που δημιουργούνται από τους ιονισμούς μέσα στην περιοχή

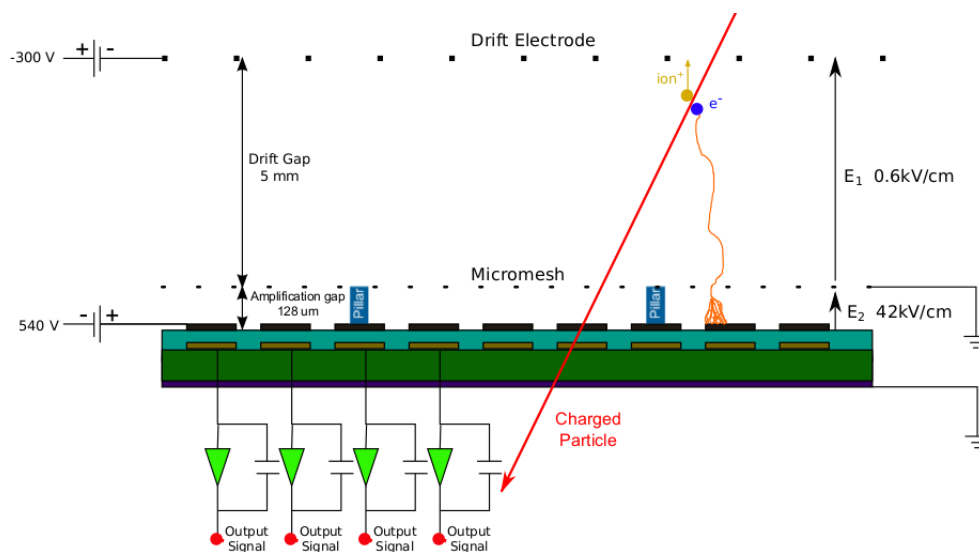
ενίσχυσης, θα κινηθούν αντίθετα, προς το πλέγμα, και θα φτάσουν εκεί περίπου στα 100 ns, που είναι ένας χρόνος αρκετά μικρός σε σχέση με άλλους ανιχνευτές. Ο λόγος για τον οποίο ο ανιχνευτής MM είναι ο πλέον κατάλληλος στο να ανταπεξέλθει σε μεγάλες ροές σωματιδίων που θα κυριαρχούν μετά από τις αλληπάλληλες αναβαθμίσεις στη φωτεινότητα του LHC, είναι ακριβώς αυτός: τα φορτία φτάνουν *γρήγορα* στις ανόδους και τις καθόδους, παράγοντας έτσι τα ηλεκτρικά σήματα με μικρή διαφορά χρόνου σε σχέση με τη στιγμή που το σωματίδιο πέρασε μέσα από τον ανιχνευτή².

Ακριβώς λόγω της βέλτιστης αποδοτικότητας και της απλότητας της κατασκευής του, ο MM επιλέχθηκε από την ομάδα του ATLAS για να στελεχώσει το New Small Wheel κατά την αναβάθμιση του ανιχνευτή. Παρ' όλα τα προτερήματά του όμως, ο MM αντιμετωπίζει και αυτός κάποια ζητήματα. Πιο συγκεκριμένα [8], όταν κατά το φαινόμενο χιονοστιβάδας το πλήθος των ηλεκτρονίων ξεπεράσει το 10^7 (Raether limit), εμφανίζονται σπινθήρες που δημιουργούν διακοπές στην τάση τροφοδοσίας η οποία χρειάζεται σημαντικό χρόνο για να επανέλθει σε φυσιολογικά επίπεδα, ενώ οι σπινθηρισμοί μπορεί επίσης να καταστρέψουν τα strips και το mesh. Η λύση στο πρόβλημα αυτό ήρθε με την εισαγωγή ενός στρώματος από *resistive strips* τοποθετημένο πάνω από το στρώμα των read-out strips [9]. Το στρώμα αυτό αποτελείται από ένα μονωτή πάχους $64\mu m$, πάνω στο οποίο έχουν εναποτεθεί λωρίδες από ειδικό υλικό αντίστασης μερικών $M\Omega/cm$ (resistive strips). Η καινοτομία της λύσης αυτής έγκειται στο γεγονός ότι τα resistive strips έχουν ακριβώς την ίδια γεωμετρική διάταξη με τα read-out strips που βρίσκονται από κάτω, με αποτέλεσμα να μην εξαπλώνουν ομοιόμορφα το φορτίο σε πολλά read-out strips, γεγονός που θα υπονόμεινε την ακρίβεια χωρικής ανάλυσης των τροχιών από τον ανιχνευτή (spatial resolution). Μαζί με τη λύση του στρώματος μόνωσης, μελέτες έδειξαν ότι γειώνοντας το πλέγμα, και εφαρμόζοντας θετική τάση στα resistive strips (αντίστροφα δηλαδή από πριν), δημιουργείται ένα πιο σταθερό ηλεκτρικό πεδίο που δεν επηρεαζόταν τόσο εύκολα από τυχόν σπινθηρισμούς, διατηρώντας έτσι τη βέλτιστη αποδοτικότητά του MM. Τέλος, τα ηλεκτρονικά συστήματα υπεύθυνα για τη συλλογή παλμών, θα συλλέγουν δεδομένα μόνο από τα strips και όχι και από το mesh όπως πριν. Πλέον όλος ο ανιχνευτής

²Αναφορικά, από τη στιγμή που θα συμβεί μία αλληλεπίδραση στο κέντρο του ATLAS, θα περάσουν 29 – 31 ns μέχρι τα μόνια που παρήχθησαν από την αλληλεπίδραση αυτή να φτάσουν στο NSW, και 50 – 75 ns μέχρι να συλλεχθεί το σήμα από τα strips λόγω του ιονισμού που προκάλεσε ένα τέτοιο μόνιο [9].

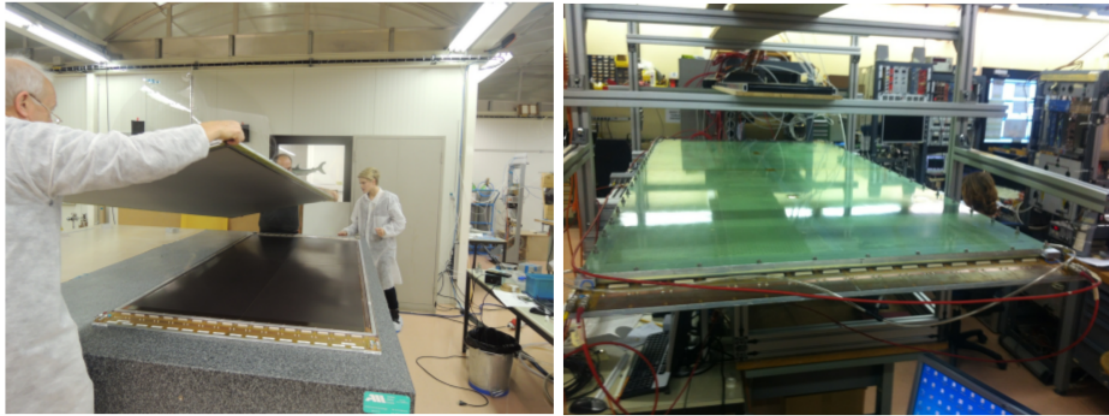
είναι προσαρτημένος πάνω σε μία ηλεκτρονική πλακέτα (PCB³) πάχους 0.5 mm. Σε κάθε τμήμα (sector) του NSW θα βρίσκονται οκτώ διαφορετικές PCBs, κάθε μία από τις οποίες θα έχει 1024 κανάλια. Ένα προς ένα, τα κανάλια αυτά είναι συνδεδεμένα με τα read-out strips, και κάθε πλακέτα PCB, είναι συνδεδεμένη με δύο readout boards, στην οποία βρίσκονται οκτώ read-out chips 64 καναλιών (VMM), που διαβάζουν και επεξεργάζονται τα ηλεκτρικά σήματα [8]. Εδώ όμως θα μπει μία άνω τελεία καθώς περισσότερες λεπτομέρειες για τα ηλεκτρονικά του NSW θα παρατεθούν σε επόμενα Κεφάλαια.

Ακολουθούν σχηματικές αναπαραστάσεις του τροποποιημένου ανιχνευτή, και της διαδικασίας κατασκευής ενός πρωτότυπου διαστάσεων $1 \times 2.4 \text{ m}^2$.



Σχήμα 2.2.III: Ο ανιχνευτής MicroMegas μετά την πρόσθεση των λωρίδων υψηλής αντίστασης (Resistive Strips) και της αλλαγής στις τάσεις τροφοδοσίας στο πλέγμα και στα strips. [9]

³Printed Circuit Board.



Σχήμα 2.2.IV: Η αριστερή φωτογραφία απεικονίζει την κατασκευή ενός μεγάλου πρωτότυπου MicroMegas. Στα δεξιά, το πρωτότυπο βρίσκεται στο εργαστήριο RD51 του CERN.[9]

Σύνοψη

Στο παρόν Κεφάλαιο μελετήθηκε η δομή των ανιχνευτών αερίου, οι οποίοι χρησιμοποιούνται κατά κόρον στη Φυσική Υψηλών Ενέργειών. Στη συνέχεια αναφέρθηκαν λεπτομέρειες για τον MicroMegas, έναν ανιχνευτή που αναπτύσσεται και από το ΕΜΠ σε συνεργασία με την υπόλοιπη ομάδα του ATLAS. Στο επόμενο Κεφάλαιο θα περιγραφεί το σύστημα ηλεκτρονικών της αναβάθμισης του New Small Wheel. Το ΕΜΠ έχει επίσης ενεργό ρόλο στη σχεδίαση και ανάπτυξη του συστήματος των ηλεκτρονικών, ενώ το κύριο μέρος της παρούσας εργασίας πραγματεύεται ουσιαστικά ένα κομμάτι του συστήματος αυτού.

3

Ηλεκτρονικά Συστήματα Ανιχνευτών

Αν και υπάρχουν πολλοί διαφορετικοί τύποι ανιχνευτών (φωτοπολλαπλασιαστές, ανιχνευτές στερεάς κατάστασης, ανιχνευτές αερίων), όπου ο κάθε ένας χρησιμοποιεί διαφορετικές αρχές για να εντοπίσει τα σωματίδια που το διαπερνούν, όλοι κάποια στιγμή μετατρέπουν την πληροφορία του εντοπισμού σε ηλεκτρικό σήμα. Ένα ακόμα βασικό κομμάτι που παίζει κεντρικό ρόλο στις ανιχνευτικές διατάξεις λοιπόν, είναι τα ηλεκτρονικά συστήματα που συλλέγουν τα σήματα (read-out electronics) των ανιχνευτών και τα μεταφράζουν με τέτοιο τρόπο ώστε να μπορούν να τα επεξεργαστούν ειδικοί αλγόριθμοι ανάλυσης δεδομένων. Σε αυτό το κεφάλαιο, θα δοθεί αρχικά μία γενική επισκόπηση των ηλεκτρονικών ενός ανιχνευτή, και θα γίνει αναφορά στους πολυάριθμους σκοπέλους που πρέπει να αποφευχθούν κατά τη σχεδίαση ενός τέτοιου συστήματος. Τελικά, θα μελετηθεί το κάθε ηλεκτρονικό εξάρτημα (ή υποσύστημα) ξεχωριστά, προκειμένου ο αναγνώστης να εμπεδώσει τον τρόπο λειτουργίας όλου του μηχανισμού.

3.1 Εισαγωγή στα Ηλεκτρονικά των Ανιχνευτών

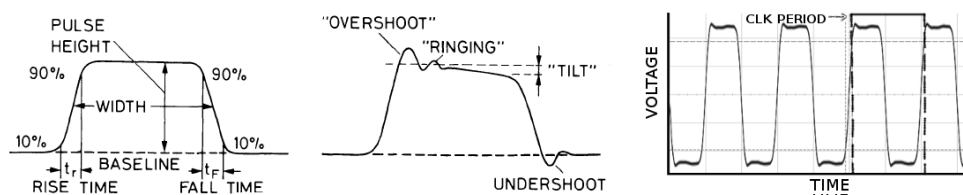
Σε αυτό το κομμάτι θα μελετηθούν οι αρχές των ηλεκτρονικών (front-end electronics) που διαβάζουν και επεξεργάζονται (signal processing) τα πρωταρχικά σήματα των ανιχνευτών. Ο σκοπός τους σε γενικές γραμμές συνοψίζεται στα εξής [3]:

- Εξαγωγή του σήματος από τους αισθητήρες του ανιχνευτή. Τυπικά, το σήμα αυτό είναι ένας βραχύς παλμός ηλεκτρικού φορτίου (της τάξης των fC ή

ακόμα και aC).

- Βελτιστοποίηση του χρόνου απόκρισης του συστήματος ανάγνωσης ώστε να μετρηθεί σωστά η ενέργεια που αντιστοιχεί στον παλμό, ο ρυθμός γεγονότων, και ο ακριβής χρόνος άφιξης του παλμού.
- Διάκριση παλμών που αντιστοιχούν σε πραγματικά γεγονότα (discrimination) από τον εγγενή ηλεκτρονικό θόρυβο.
- Μετατροπή του επεξεργασμένου αναλογικού σήματος σε ψηφιακό, και αποστολή σε άλλα υποσυστήματα για περαιτέρω ανάλυση.

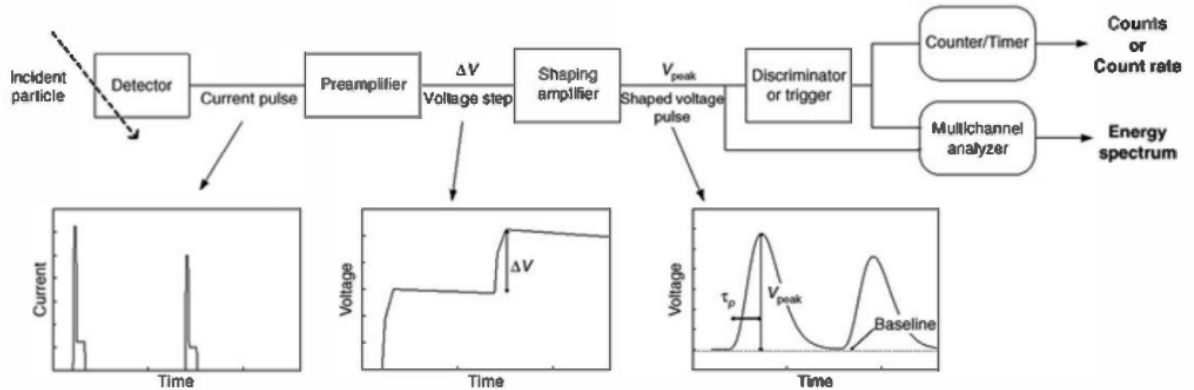
Οι ηλεκτρικοί παλμοί, και τα σήματα γενικότερα, φέρουν πληροφορία σε δύο μορφές: αναλογική ή ψηφιακή. Ένα αναλογικό σήμα κωδικοποιεί με συνεχή τρόπο την πληροφορία, μεταβάλλοντας τη συχνότητά του, το πλάτος του, και τη γενικότερη μορφολογία του. Για παράδειγμα, οι σπινθηριστές παράγουν σήματα των οποίων το πλάτος είναι ανάλογο της ενέργειας που εναπόθεσε ένα σωματίδιο σε αυτούς. Οι ψηφιακοί παλμοί από την άλλη, χαρακτηρίζονται μόνο από δύο καταστάσεις, το λογικό μηδέν και το λογικό ένα, και μολονότι φαίνεται πως δεν έχουν τις δυνατότητες των αναλογικών σημάτων ως προς τη μετάδοση πληροφορίας, είναι τελικά πιο αξιόπιστοι, καθώς μένουν σχετικά ανεπηρέαστοι από τυχόν παραμορφώσεις κατά τη μετάδοσή τους από το ένα σημείο στο άλλο [4]. Ο γενικός κανόνας υποδεικνύει ότι τα αναλογικά σήματα που προέρχονται από τους ανιχνευτές, αφού υποστούν επεξεργασία, κάποια στιγμή θα μετατραπούν σε ψηφιακά.



Σχήμα 3.1.1: Αριστερά και Κέντρο: Βασική ορολογία γύρω από του παλμούς [4]. Σημαντικός όρος είναι και το *leading* (ή *rising*) *edge*, που είναι ο αριστερός κλάδος ενός παλμού με ανοδική τάση, και το *falling edge*, που είναι ο δεξιός κλάδος ενός παλμού που πέφτει από τη μέγιστη τάση, πίσω στη *baseline*. Δεξιά: Τυπικός παλμός ρολογιού όπως φαίνεται σε έναν παλμογράφο. Πρόκειται για έναν περιοδικό τετραγωνικό παλμό, η περίοδος του οποίου (*clk period*) είναι αντίστροφη της συχνότητας του ρολογιού ($f = 1/T$).

3.2 Η Επεξεργασία των Σημάτων

Η βασική μορφολογία της αλυσίδας των ηλεκτρονικών συστημάτων ενός ανιχνευτή δίνεται στο Σχήμα 3.2.1:



Σχήμα 3.2.1: Η αλυσίδα επεξεργασίας ηλεκτρικών παλμών σε έναν ανιχνευτή, μαζί με τα αποτελέσματα της μετά από κάθε βήμα [5].

Όταν το προσπίπτον σωματίδιο εναποθέσει ενέργεια στον ανιχνευτή, αυτή την ενέργεια ο ανιχνευτής θα τη μετατρέψει σε έναν παλμό ρεύματος¹, ο οποίος συνήθως είναι σύντομος, και μικρός σε ένταση. Για το λόγο αυτό, το σήμα περνάει πρώτα από έναν Προενισχυτή (*Preamplifier*). Οι προενισχυτές που χρησιμοποιούνται σε τέτοιες διατάξεις είναι charge-sensitive, δηλαδή δέχονται έναν βραχύ παλμό ρεύματος (φορτίου) στην είσοδό τους, και παράγουν μία βηματική κυματομορφή τάσης (voltage step), το ύψος ΔV της οποίας είναι ανάλογο του φορτίου στην είσοδο. Ο *Ενισχυτής Διαμόρφωσης* (*Shaping Amplifier*), λαμβάνει αυτό το σήμα και το μορφοποιεί με τη χρήση φίλτρων, μετατρέποντάς το σε πιο διαχειρίσιμες μορφές (όπως Gaussιανές), με μέγιστη τάση V_{peak} ανάλογη του ύψους του βήματος ΔV . Επομένως η μέγιστη τάση είναι ανάλογη του αρχικού φορτίου, άρα και της ενέργειας που απορρόφησε ο ανιχνευτής από το σωματίδιο. Τελικά, οι επεξεργασμένοι παλμοί στέλνονται σε έναν *Διευκρινιστή* (*Discriminator*), ο οποίος ανάλογα το πλάτος της κυματομορφής που του δίνεται, παράγει και τους αντίστοιχους λογικούς παλμούς στην έξοδό του. Τέλος, ένας *Μετρητής* (*Counter*) ή ένας *Πολυκαναλικός Αναλυτής* (*Multichannel*

¹Όπως εξετάστηκε σε προηγούμενο Κεφάλαιο, ο ανιχνευτής MicroMegas τελεί αυτή τη μετατροπή συλλέγοντας τα φορτία που προέκυψαν από αλληπάλληλους ιονισμούς.

Analyzer (MCA)) λαμβάνει τους λογικούς παλμούς του discriminator ή και την έξοδο του shaping amplifier μαζί, ώστε να καταμετρήσει τα γεγονότα (counter), ή να τα κατηγοριοποιήσει ανάλογα με την ενέργειά τους (MCA) [5]. Ο αναγνώστης παραπέμπεται στο Σχήμα 3.2.1 για μία γενική επισκόπηση.

3.2.1 Τα Προβλήματα που Εμφανίζονται

Αν και η διαδικασία που μόλις εξετάσθηκε φαίνεται να είναι απλή, στην πραγματικότητα υπάρχουν πολλοί παράγοντες που επηρεάζουν το αποτέλεσμα σε κάθε βήμα της επεξεργασίας ενός σήματος, υπονομεύοντας έτσι τον γενικό στόχο των ηλεκτρονικών ενός ανιχνευτή· να μην παραμορφώνουν δηλαδή τα σήματα που λαμβάνουν, ώστε να μεταδίδουν την πληροφορία ακέραια. Σε κάθε περίπτωση σχεδίασης ενός τέτοιου συστήματος ηλεκτρονικών, ο χρήστης πρέπει να λαμβάνει τους εξής παράγοντες υπόψιν [5]:

- Τα γεγονότα που καταγράφει ο ανιχνευτής είναι τυχαία κατανεμημένα στο χρόνο. Αυτό σημαίνει φυσικά, πως και οι παλμοί που θα παράγει, θα είναι και αυτοί μη περιοδικοί. Κατά συνέπεια, η χρονική διαφορά μεταξύ δύο παλμών μπορεί να είναι μεγάλη, ή μικρή, σε σύγκριση με τη σταθερά χρόνου τ του κυκλώματος. Στην περίπτωση λοιπόν που δύο ηλεκτρικά σήματα καταγράφονται πολύ κοντά το ένα με το άλλο, συνήθως επέρχεται παρεμβολή, γεγονός που οδηγεί σε *συσσώρευση (pulse pile-up)*.
- Ο ηλεκτρονικός θόρυβος (*electronic noise*), είναι ένα εγγενές φαινόμενο σε κάθε ηλεκτρικό κύκλωμα, ενώ η αναλογία Σήματος προς Θόρυβο (*Signal to Noise ratio*) S/N , πρέπει να ελέγχεται προσεκτικά έτσι ώστε να περιορίζεται η παραμόρφωση των χρήσιμων πληροφοριών που φέρουν οι παλμοί, από εξωτερικούς παράγοντες. Ιδίως σε εφαρμογές ανιχνευτικών διατάξεων, το πρόβλημα του θορύβου είναι πολύ σημαντικό, καθώς όχι μόνο τα προς επεξεργασία σήματα είναι αναλογικά (άρα και πιο ευαίσθητα), αλλά είναι και σχετικά αδύναμα.
- Οι μετρήσεις του πλάτους μίας κυματομορφής γίνονται πάντα με σημείο αναφοράς το μηδέν, το οποίο όμως δυστυχώς, δεν είναι απόλυτο. Μπορεί σε κάποιο μέρος της επεξεργασίας ενός παλμού, να συμβεί το επονομαζόμενο και *baseline shift*, ένα φαινόμενο κατά το οποίο το κατώτερο σημείο αναφοράς

(baseline) του παλμού μετατοπίζεται προς τα πάνω ή προς τα κάτω, οδηγώντας έτσι σε λανθασμένες μετρήσεις του πλάτους σε μετέπειτα στάδια της αλυσίδας επεξεργασίας.

- Τα σήματα που παράγει ο ανιχνευτής μπορεί μεν να είναι ανάλογα της ενέργειας, όμως η αναλογία αυτή δεν είναι απόλυτη, αλλά εξαρτάται και από δευτερεύοντες παράγοντες, όπως για παράδειγμα από τη θερμοκρασία και από την τάση τροφοδοσίας. Για το λόγο αυτό μεγάλη προσοχή πρέπει να δίνεται στη σταθεροποίηση του φάσματος (*spectrum stabilization*), η οποία επιτυγχάνεται μόνο με περιορισμό των διακυμάνσεων των παρασιτικών αυτών παραγόντων.
- Σε πολλές εφαρμογές, απαιτείται το ηλεκτρικό σύστημα να έχει ακαριαία απόκριση στους παλμούς που λαμβάνει από το ανιχνευτικό σύστημα. Τα κυκλώματα αυτά, που είναι υπεύθυνα να παράγουν ηλεκτρικούς παλμούς στενά συνδεδεμένους με το *πότε ακριβώς* έλαβε χώρα ένα γεγονός στον ανιχνευτή, ονομάζονται *triggers*. Ο σχεδιασμός τέτοιων κυκλωμάτων, ιδίως σε πειράματα με μεγάλη ροή σωματιδίων, δεν είναι ιδιαίτερα εύκολος.

Για να αντιληφθεί κανείς τις δυσκολίες που προκύπτουν κατά το σχεδιασμό κυκλωμάτων που οφείλουν να διατηρούν την ακεραιότητα των σημάτων που προκύπτουν από ανιχνευτικές διατάξεις σε ένα πείραμα Φυσικής Υψηλών Ενεργειών, θα πρέπει εκτός από τους παράγοντες που αναφέρθηκαν παραπάνω, να συμπεριλάβει και τον παράγοντα της *ταχύτητας* των παλμών που καλείται ένα κύκλωμα να επεξεργαστεί. *Γρήγορα* ονομάζονται τα σήματα εκείνα των οποίων ο χρόνος ανόδου (rise time² είναι μερικά *ns*, ενώ ένα *αργό* σήμα, έχει rise time μερικών εκατοντάδων *ns* ή και περισσότερο. Ένα περιοδικό σήμα (π.χ. ένας παλμός ρολογιού) μεγάλης συχνότητας για παράδειγμα, μπορεί να θεωρηθεί γρήγορο σήμα. Δυστυχώς, ένα γρήγορο σήμα είναι πολύ πιο επιρρεπές σε παραμορφώσεις λόγω μικρών, ανεπιθύμητων *παρασιτικών κυκλωμάτων* που δύνανται να εμφανιστούν κατά τη μετάδοσή του. Τα παρασιτικά αυτά κυκλώματα αποτελούνται από υψηλές αντιστάσεις, αυτεπαγωγές, και χωρητικότητες που εμφανίζονται στις διασυνδέσεις μεταξύ διαφορετικών κυκλωμάτων από τα οποία περνάει το σήμα. Το ισοδύναμο κύκλωμα ενός τέτοιου

²Ο χρόνος που απαιτείται για έναν παλμό να φτάσει στο 90% της έντασής (ή πλάτους) του, ξεκιτώντας από τα 10% της έντασής του. (βλ. Σχ. 3.1.1)

παρασιτικού μηχανισμού, μπορεί για παράδειγμα να είναι ένα κύκλωμα RC , με μία σταθερά χρόνου $\tau = RC$, η οποία αν είναι πολύ μεγαλύτερη σε σχέση με το rise time του παλμού, προκαλεί σημαντική παραμόρφωση του σήματος. Το αποτέλεσμα είναι ότι ακόμα και αν απλά στείλει κανείς ένα γρήγορο σήμα μέσα από μία διασύνδεση, να προκαλείται αυτόματα μη αποδεκτή αλλοίωση μετά από μόλις λίγα εκατοστά στη γραμμή [4].

Στενά συνδεδεμένη με την ταχύτητα ενός παλμού είναι και η *συχνότητά (frequency)* του. Ανάλογα με το εύρος ζώνης (bandwidth) των σημάτων που διατρέχουν ένα κύκλωμα, θα λάβει κανείς και άλλα αποτελέσματα. Στις εφαρμογές που μελετώνται εδώ, προφανώς το εύρος ζώνης είναι απρόβλεπτο, καθώς τα σήματα που μπορεί να παράξει ένας ανιχνευτής δεν μπορεί να πει κανείς ότι διέπονται ακριβώς από ντετερμινιστικά κριτήρια. Γενικά, κανένα πραγματικό κύκλωμα δεν μπορεί να αποκρίνεται με τον ίδιο τρόπο σε όλες τις συχνότητες, καθώς όλα τα μέρη του κυκλώματος, είτε πραγματικά είτε παρασιτικά, θα δρουν ως *φίλτρα*³. Ο τύπος του φίλτρου (χαμηλοπερατό, ζωνοδιαβατό κ.λπ.) καθορίζει το πως θα παραμορφώνει τα σήματα τα οποία δέχεται. Περισσότερα για τα φίλτρα θα αναφερθούν παρακάτω στο παρών Κεφάλαιο.

Είναι λοιπόν πλέον φανερά τα εμπόδια που εμφανίζονται κατά τη σχεδίαση των ηλεκτρονικών που επεξεργάζονται τα πρωταρχικά σήματα τα οποία παράγει ένας ανιχνευτής σε κάποιο πείραμα. Τις περισσότερες φορές, προσεκτικές μελέτες για τη φύση των σημάτων πρέπει να λάβουν χώρα πριν από τη σχεδίαση των ηλεκτρονικών συστημάτων, ώστε να ληφθούν αποφάσεις για τους συμβιβασμούς στους οποίους πρέπει να προβεί μία ερευνητική ομάδα, προκειμένου να πετύχει τους στόχους της. Επίσης, τέτοιες μελέτες είναι απαραίτητες ώστε να επιλεγθούν τα κατάλληλα ηλεκτρονικά εξαρτήματα που θα αποδίδουν βέλτιστα βάσει των περιστάσεων. Για το λόγο αυτό, στη συνέχεια θα παρουσιαστούν με λίγο περισσότερες λεπτομέρειες τα συστήματα που χρησιμοποιούνται συνηθέστερα στα ανιχνευτικά συστήματα της Φυσικής Υψηλών Ενεργειών, τα οποία, όπως θα διαπιστώσει ο αναγνώστης σύντομα, βρίσκουν εφαρμογή και στην αναβάθμιση του New Small Wheel.

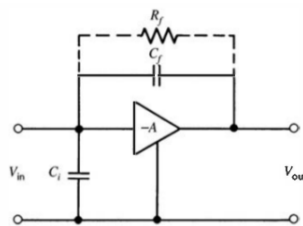
³Φίλτρο ονομάζεται η ηλεκτρονική εκείνη διάταξη που βάσει κατασκευής, απορρίπτει ένα συγκεκριμένο εύρος συχνοτήτων, μειώνοντας δραστικά το πλάτος του.

3.3 Ηλεκτρονικά Επεξεργασίας Σημάτων

Υπενθυμίζεται ότι τα κύρια ηλεκτρονικά εξαρτήματα που χρησιμοποιούνται για την επεξεργασία των παλμών που προκύπτουν από έναν ανιχνευτή είναι οι Προενισχυτές (Preamplifiers), οι Ενισχυτές Διαμόρφωσης (Shaping Amplifiers), οι Διευκρινιστές (Discriminators), οι Μετρητές (Counters) και οι Πολυκαναλικοί Αναλυτές (Multichannel Analyzers). Σε αυτό το κομμάτι θα γίνει μία λεπτομερέστερη ανάλυση αυτών των υποσυστημάτων, μαζί με μία αναφορά στους Μετατροπείς από Αναλογικά σε Ψηφιακά σήματα (Analog to Digital Converters (ADC)), στα συστήματα υπεύθυνα για το triggering, και στους Μετατροπείς Χρόνου-σε-Τάση (Time-to-Amplitude Converters (TAC)).

Preamplifiers

Το κύριο έργο των προενισχυτών είναι να ενισχύουν τα αδύναμα σήματα των ανιχνευτών και να τα διοχετεύουν στα υπόλοιπα ηλεκτρονικά υποσυστήματα [4]. Ταυτοχρόνως, θα πρέπει να επιβαρύνουν τους παλμούς με όσο το δυνατόν λιγότερο θόρυβο. Για το λόγο αυτό, ο προενισχυτής τοποθετείται συνήθως πολύ κοντά στον ανιχνευτή, ελαχιστοποιώντας έτσι το μήκος του καλωδίου που μεταφέρει τον προς ενίσχυση παλμό⁴. Υπενθυμίζεται επίσης, ότι οι προενισχυτές που χρησιμοποιούνται περισσότερο σε ανιχνευτικές διατάξεις, είναι τύπου charge-sensitive, δηλαδή ενισχύουν παλμούς φορτίου, μετατρέποντάς τους σε παλμούς τάσης. Το κύκλωμα ενός τέτοιου προενισχυτή παρατίθεται στο Σχήμα 3.3.I.



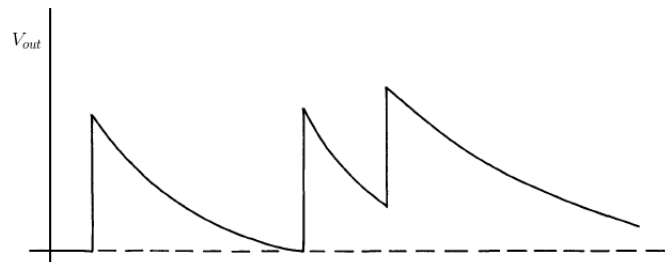
Σχήμα 3.3.I: Απλοποιημένο διάγραμμα ενός προενισχυτή [5].

⁴Το αποτέλεσμα αυτού είναι να μειώνεται ο θόρυβος που προκαλείται από ηλεκτρομαγνητικά πεδία που βρίσκονται στο περιβάλλον, και να ελαττώνεται η χωρητικότητα του καλωδίου (cable capacitance), γεγονός που αυξάνει το λόγο S/N .

Η τάση εξόδου του ενισχυτή, V_{out} θα είναι ανάλογη του ύψους του παλμού φορτίου στην είσοδο⁵, αρκεί η διάρκεια του παλμού εισόδου, να είναι κατά πολύ μικρότερη της σταθεράς χρόνου του κυκλώματος $\tau = R_f C_f$. Επίσης αξίζει να σημειωθεί ότι ο χρόνος ανόδου (rise time) του παλμού της τάσης εξόδου, δεν καθορίζεται μόνο από τη διάρκεια του παλμού εισόδου, αλλά και από παρασιτικά κυκλώματα που εμφανίζονται στις διασυνδέσεις. Πιο συγκεκριμένα, η σταθερά του χρόνου παρασιτικών κυκλωμάτων που δημιουργούνται λόγω ατελειών στη διασύνδεση του προενισχυτή με τον ανιχνευτή (η αντίσταση που εμφανίζεται δημιουργεί σύζευξη με την χωρητικότητα στην είσοδο C_i), αν είναι συγκρίσιμη με τη διάρκεια του παλμού στην είσοδο, θα καθυστερήσει σημαντικά την άνοδο του παλμού τάσης εξόδου [5].

Άλλα προβλήματα, συνυφασμένα κυρίως με την τιμή την αντίστασης R_f (βλ. Σχ. 3.3.I), είναι η ύπαρξη εγγενούς ηλεκτρονικού θορύβου στην αντίσταση, και η στενή σχέση της R_f με το φαινόμενο του pile-up, το οποίο μπορεί να οδηγήσει σε κορεσμό της διαδικασίας της ενίσχυσης. Γενικά, κάθε αντίσταση συνεισφέρει στο θόρυβο ενός σήματος που το διαπερνά, λόγω του φαινομένου που ονομάζεται *Θόρυβος Johnson* (Johnson noise). Πρόκειται για ένα είδος θορύβου με ομογενή συνεισφορά σε όλο το εύρος συχνοτήτων ενός σήματος, η ένταση του οποίου εξαρτάται από την τιμή της αντίστασης και τη θερμοκρασία, και οφείλεται σε θερμικές διακυμάνσεις μέσα στον αντιστάτη [18]. Προκειμένου να περιορίσει κανείς τη συνεισφορά του θορύβου Johnson σε έναν προενισχυτή, μπορεί να αυξήσει την τιμή της αντίστασης R_f , όμως αυτό με τη σειρά του θα οδηγήσει σε αύξηση της σταθεράς χρόνου τ , γεγονός ανεπιθύμητο. Ένας προενισχυτής παράγει αναλογικά σήματα τάσης που τείνουν να είναι βηματικής μορφής, δηλαδή με απότομες αυξήσεις. Το ύψος του "σκαλοπατιού" είναι ανάλογο της έντασης του παλμού εισόδου. Ο προενισχυτής αφού παράξει το επιθυμητό σήμα, θα προσπαθήσει να μηδενίσει την τάση εξόδου, μία διαδικασία η χρονική διάρκεια της οποίας είναι ανάλογη της σταθεράς χρόνου του κυκλώματος $R_f C_f$. Η τάση εξόδου θα μειώνεται εκθετικά, παράγοντας μία εκθετική ουρά, το μήκος της οποίας αυξάνεται με αύξηση του τ . Αν ο προενισχυτής λάβει και άλλο σήμα στην είσοδό του πριν αυτή η ουρά εξαλειφθεί, το επόμενο σήμα εξόδου θα συσσωρευτεί πάνω στο μη-μηδενικό παλιό σήμα, οδηγώντας σε σταδιακό *pile-up*, ένα φαινόμενο πολύ σύνηθες, ιδίως σε πειράματα με υψηλό ρυθμό γεγονότων. Η συσσώρευση αυτή απεικονίζεται στο Σχήμα 3.3.II.

⁵ $V_{out} \simeq -\frac{Q}{C_f}$



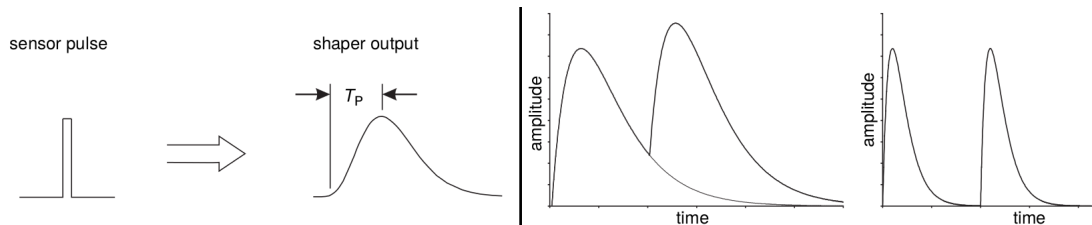
Σχήμα 3.3.Π: Διάγραμμα τάσης εξόδου προς χρόνο. Η τάση εξόδου δεν προλαβαίνει να μηδενιστεί (η εκθετική μείωση είναι εμφανής) και ο επόμενος παλμός προστίθεται στον παλιό [4].

Αν και ο Shaping Amplifier είναι σχεδιασμένος με τέτοιο τρόπο ώστε να αφαιρεί τυχόν συσσωρεύσεις, στην περίπτωση που το pile-up γίνει υπερβολικά έντονο (συμβούν δηλαδή αλληπάλληλες συσσωρεύσεις), τότε ο προενισχυτής θα ξεπεράσει το όριο της τάσης εξόδου του, και θα παράξει μη-αποδεκτά αλλοιωμένους παλμούς. Για πειράματα με υψηλή συχνότητα γεγονότων, ο μόνος τρόπος να αποφευχθεί αυτό είναι να μειωθεί η τιμή της αντίστασης R_f , άρα και ο χρόνος που θα χρειάζεται η τάση να μηδενιστεί στην έξοδο. Αυτό όμως αυξάνει το θόρυβο Johnson. Η αντίθεση των δύο αυτών φαινομένων έχει οδηγήσει πολλούς σχεδιαστές στη πλήρη αφαίρεση της αντίστασης R_f του προενισχυτή, γεγονός που μειώνει δραστικά το χρόνο αποκατάστασης του σήματος εξόδου, και το θόρυβο Johnson. Μία τέτοια διάταξη όμως απαιτεί τεχνικές active reset, δηλαδή κάποιο λογικό κύκλωμα που θα επαναφέρει την τάση εξόδου του προενισχυτή σε αποδεκτά επίπεδα όταν αυτή ξεπεράσει κάποιο όριο που είναι κοντά στο επίπεδο κορεσμού του συστήματος [5].

Shaping Amplifiers

Οι παλμοί τάσης που προκύπτουν από τον προενισχυτή, πρέπει να υποστούν ακόμα περισσότερη επεξεργασία, ώστε να αλλάξει το σχήμα τους σε μία προκαθορισμένη μορφή. Αυτή τη μορφή θα τη δώσει στους παλμούς ένας ενισχυτής διαμόρφωσης, ο οποίος πρέπει να μετατρέπει τα σήματα του προενισχυτή με τέτοιο τρόπο ώστε να διατηρείται πλήρως η πληροφορία που εμπεριέχεται σε αυτά. Σε γενικές γραμμές (και με απλά λόγια), ένας μορφοποιητής παλμών μπορεί να δεχθεί ένα βραχύ (στενό) σήμα και να το απλώσει, εξομαλύνοντας (ή στρογγυλεύοντας) ταυτόχρονα την κορυφή του στο χρόνο που παρουσιάζεται το ολικό του μέγιστο. Μπορεί επίσης

να διαχωρίσει δύο παλμούς οι οποίοι έχουν εξέλθει από τον προενισχυτή με μικρή χρονική διαφορά, και παρουσιάζουν pile-up [3]. Οι δύο αυτές λειτουργίες που παρουσιάζουν με χονδροειδή τρόπο τη λειτουργικότητα των ενισχυτών διαμόρφωσης παρατίθενται στο Σχήμα 3.3.III.

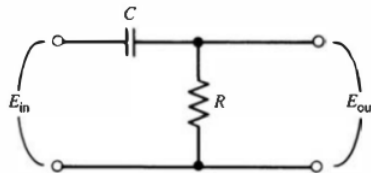


Σχήμα 3.3.III: Στα αριστερά απεικονίζεται το "άπλωμα" ενός σύντομου παλμού, και στα δεξιά ο διαχωρισμός δύο παλμών με pile-up. Ο πρώτος παλμός πλέον έχει επιστρέψει στην απόλυτη baseline πριν εμφανιστεί ο δεύτερος [3].

Σε κάθε περίπτωση, ο ενισχυτής οφείλει να διατηρεί την πληροφορία που ενδιαφέρει περισσότερο τον σχεδιαστή του συστήματος. Αν για παράδειγμα πρέπει από τον παλμό να εξαχθεί πληροφορία για το χρόνο (το πότε ακριβώς δηλαδή συνέβη ένα γεγονός στον ανιχνευτή), τότε απαιτείται γρήγορη απόκριση από το διαμορφωτή. Αν πρέπει όμως να εξαχθεί πληροφορία για το πόσο φορτίο συνέλεξαν οι αισθητήρες (δηλαδή πόση ενέργεια εναπόθεσε η ιοντίζουσα ακτινοβολία στον ανιχνευτή), τότε πρέπει να διατηρηθεί το πλάτος του παλμού ακέραιο. Τέλος, αξίζει να σημειωθεί ότι μία άλλη λειτουργία του shaping amplifier είναι να βελτιστοποιεί την αναλογία καθαρού σήματος προς θόρυβο S/N . Για ένα δοθέν φάσμα ηλεκτρονικού θορύβου, υπάρχει ένα συγκεκριμένο σχήμα παλμού όπου ο θόρυβος ελαχιστοποιείται. Οι μακριές ουρές στους παλμούς για παράδειγμα, που είναι συνήθως αποτέλεσμα μεγάλης σταθεράς χρόνου $\tau = RC$ στον προενισχυτή, ευνοεί παραμορφώσεις λόγω θορύβου. Αναφορικά, μορφές σημάτων που θεωρούνται ότι υπερτερούν σε αυτό τον τομέα είναι η τριγωνική ή η Γκαουσιανή [4].

Περνώντας σε περισσότερες λεπτομέρειες, αξίζει να αναφερθεί ότι όλες αυτές τις απαραίτητες λειτουργίες, οι ενισχυτές διαμόρφωσης τις επιτυγχάνουν συνήθως με τη χρήση φίλτρων, ποικίλων μορφών. Τα πιο απλά από αυτά τα κυκλώματα είναι κυκλώματα RC , τα οποία δρουν ως διαφοριστές (*differentiators*), ή ως ολοκληρωτές (*integrators*). Η ονοματολογία αυτή προκύπτει από τις εξισώσεις τάσεων εισόδου-

εξόδου που τα διέπουν. Ο διαφοριστής CR , είναι ουσιαστικά ένα υψιπερατό φίλτρο (δηλαδή το φάσμα των υψηλών συχνοτήτων περνάει από αυτόν χωρίς να αποσβένεται), και το ισοδύναμο κύκλωμα του διαφοριστή με έναν πυκνωτή C και μία αντίσταση R απεικονίζεται στο Σχήμα 3.3.IV⁶.



Σχήμα 3.3.IV: Ο διαφοριστής CR . Υψιπερατό φίλτρο [5].

Αν $\tau = RC$ η σταθερά χρόνου του κυκλώματος του διαφοριστή, τότε μετά από μερικές απλές πράξεις, καταλήγει κανείς στη σχέση μεταξύ τάσης εξόδου και εισόδου που διέπεται από την εξής διαφορική:

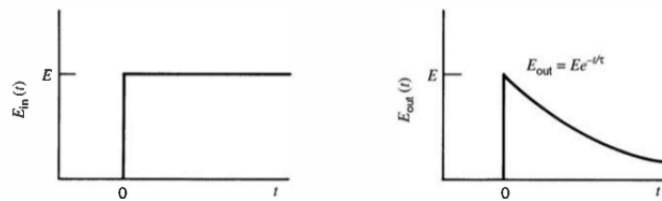
$$V_{out} \simeq \tau \frac{dV_{in}}{dt} \quad (3.3.1)$$

Εξ ου και το όνομα διαφοριστής. Η απόκριση του συστήματος αυτού σε μία βηματική συνάρτηση τάσης με μέγιστη τάση V είναι δε η εξής:

$$V_{out} = V e^{-t/\tau} \quad (3.3.2)$$

Το αποτέλεσμα παρατίθεται στη γραφική παράσταση του Σχήματος 3.3.V.

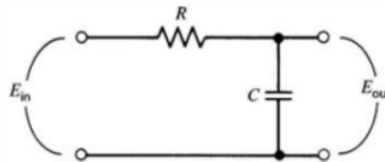
⁶Σημειώνεται ότι στα επόμενα σχήματα που ελήφθησαν από τη βιβλιογραφία [5], ο συμβολισμός E αντιστοιχεί σε τάσεις, που συνήθως συμβολίζονται με V .



Σχήμα 3.3.V: Απόκριση ενός διαφοριστή CR σε μία βηματική συνάρτηση τάσης [5].

Αυτή η περίπτωση προσεγγίζει τη διαμόρφωση ενός γρήγορου παλμού με μικρό rise time και με μεγάλη εκθετική ουρά απόσβεσης. Η απότομη αύξηση τάσης θα διατηρηθεί ανέπαφη, καθώς το rise time είναι μικρότερο του τ (υπενθυμίζεται ότι εν προκειμένω μελετάται ένα υψιπερατό φίλτρο, και οι υψηλές συχνότητες ουσιαστικά διέπονται από μικρούς χρόνους ανόδου). Η γρήγορη αύξηση θα περάσει, όμως το αργό εκθετικό σβήσιμο θα αποκοπεί [5].

Ο ολοκληρωτής RC τώρα, είναι πρακτικά ένα χαμηλοπερατό φίλτρο, δηλαδή αποκόβει τις υψηλές συχνότητες με μικρούς χρόνους ανόδου (rise times) και αφήνει ανέπαφες τις χαμηλές. Το ισοδύναμο κύκλωμα του ολοκληρωτή παρατίθεται στο Σχήμα 3.3.VI:



Σχήμα 3.3.VI: Ο ολοκληρωτής RC . Χαμηλοπερατό φίλτρο [5].

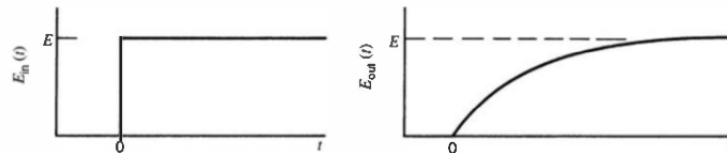
Αν $\tau = RC$ η σταθερά χρόνου του κυκλώματος του ολοκληρωτή, τότε μετά από μερικές απλές πράξεις, καταλήγει κανείς στη σχέση μεταξύ τάσης εξόδου και εισόδου που διέπεται από την εξής ολοκληρωτική εξίσωση:

$$V_{out} \simeq \frac{1}{\tau} \int V_{in} dt \quad (3.3.3)$$

Εξ ου και το όνομα ολοκληρωτής. Η απόκριση του συστήματος αυτού σε μία βηματική συνάρτηση τάσης με μέγιστη τάση V είναι δε η εξής:

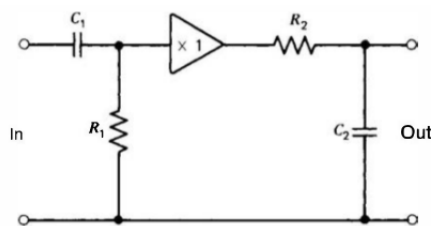
$$V_{out} = V(1 - e^{-t/\tau}) \quad (3.3.4)$$

Το αποτέλεσμα παρατίθεται στη γραφική παράσταση του Σχήματος 3.3.VII.



Σχήμα 3.3.VII: Απόκριση ενός ολοκληρωτή RC σε μία βηματική συνάρτηση τάσης [5].

Μέχρι τώρα όμως, δεν μπορεί να πει κανείς ότι οι κυματομορφές που προκύπτουν από τα δύο προαναφερθέντα φίλτρα είναι ακριβώς οι επιθυμητές για την ανάλυση παλμών από έναν ανιχνευτή. Για παράδειγμα, ο παλμός που προκύπτει από τον διαφοριστή CR στη βηματική συνάρτηση (Σχήμα 3.3.V) διατηρεί τη μέγιστη τάση του για πολύ μικρό χρονικό διάστημα, δυσχεραίνοντας έτσι μία ανάλυση που απαιτεί μέτρηση μέγιστης τάσης για καθορισμό ενέργειας. Επίσης, αν υπάρχει θόρυβος που επηρεάζει μόνο τα υψηλά φάσματα συχνοτήτων, η χρήση μόνο ενός διαφοριστή (υψηλερατό φίλτρο) για διαμόρφωση σήματος δεν θα τον αποκλείσει. Για το λόγο αυτό, χρησιμοποιούνται συνδυασμοί και των δύο τύπων φίλτρου (CR διαφοριστή και μετά RC ολοκλήρωση του σήματος), προκειμένου να αποκλειστεί μεγαλύτερο φάσμα ηλεκτρονικού θορύβου, και προς περαιτέρω εξομάλυνση των σημάτων. Ένα κύκλωμα $CR - RC$, παρατίθεται στο Σχήμα 3.3.VIII:



Σχήμα 3.3.VIII: Ο διαμορφωτής $CR - RC$. Ένας τελεστικός ενισχυτής διαχωρίζει τα δύο επιμέρους κυκλώματα. [5].

Τέλος, αξίζει να αναφερθεί και ένα ακόμα βήμα στη διαμόρφωση, που λαμβάνει χώρα εάν μετά από το κύκλωμα $CR - RC$ προστεθεί και μία συστοιχία από ολοκληρωτές RC . Ο διαμορφωτής $CR - RC$ από μόνος του βελτιώνει κατά πολύ το λόγο S/N αφού αποκλείει το θόρυβο από μεγάλο φάσμα συχνοτήτων, όμως ο παλμός επιδέχεται περαιτέρω βελτίωσης, καθώς ο λόγος S/N θα αυξηθεί κατά 18% εάν μετατραπεί σε Γκαουσιανή μορφή [4]. Η μορφή αυτή προσεγγίζεται από από την αλληλουχία $CR - RC - RC - RC - \dots$. Έτσι προκύπτει μία ημι-Γκαουσιανή μορφή η οποία είναι σαφώς η πιο επιθυμητή σε σχέση με ο,τι έχει αναφερθεί μέχρι τώρα.

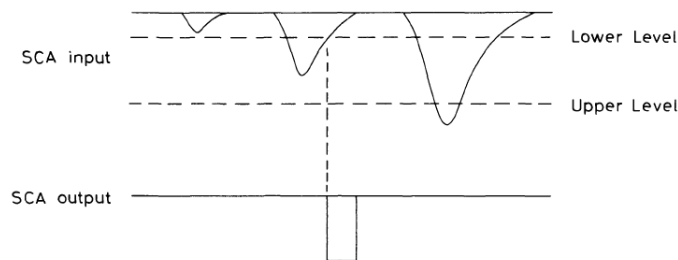
Discriminators

Μετά από την επεξεργασία των παλμών, συνήθως τοποθετείται ένα κύκλωμα το οποίο να αναγνωρίζει τους παλμούς που αντιστοιχούν σε γεγονότα. Αυτό το κύκλωμα ονομάζεται διευκρινιστής. Ο πιο απλός τύπος διευκρινιστή είναι ο ολοκληρωτικός διευκρινιστής (*integral discriminator*), ο οποίος παράγει έναν λογικό παλμό μόνο αν το αναλογικό σήμα που δέχεται στην είσοδό του έχει πλάτος μεγαλύτερο από ένα πολύ συγκεκριμένο κατώτερο κατώφλι [4, 5]. Αν η μέγιστη τάση του σήματος στην είσοδο είναι μικρότερη από αυτό το κατώτερο επίπεδο, τότε δεν εμφανίζεται τίποτα στην έξοδο του διευκρινιστή. Το κύκλωμα αυτό, χρησιμεύει και σαν *trigger primitive*⁷), δεδομένου ότι ο λογικός παλμός παράγεται άμεσα⁸ σε σχέση με το *leading edge* (βλ. Σχ. 3.1.Ι του παλμού που ξεπερνάει το κατώφλι του discriminator. Αυτή η διαδικασία ονομάζεται *leading edge triggering*. Τέλος, αξίζει να αναφερθεί ότι το κατώτατο όριο του διευκρινιστή μπορεί να οριστεί από το χρήστη, και η επιλογή της τιμής του εξαρτάται από πολλούς παράγοντες, όπως η ένταση του ηλεκτρονικού θορύβου που υπάρχει σαν υπόβαθρο, ή η ευαισθησία του ανιχνευτή. Σε πολλές εφαρμογές, το κατώφλι ορίζεται μόλις πάνω από το θόρυβο του συστήματος, προκειμένου να καταμετρώνται όλοι οι παλμοί ανεξαρτήτως μεγέθους, που προκύπτουν από τον ανιχνευτή [4]. Εκτός από τον προαναφερθέντα διευκρινιστή, υπάρχει και ο διαφορικός διευκρινιστής (*differential discriminator*), ο οποίος ονομάζεται και μονοκαναλικός αναλυτής (*single channel analyzer (SCA)*). Η κύρια διαφορά του σε σχέση

⁷Παράγει δηλαδή τα πρωταρχικά σήματα που είναι χρήσιμα για το triggering ενός ανιχνευτικού συστήματος. Περισσότερες λεπτομέρειες για το θέμα αυτό μπορούν να βρεθούν στο επόμενο Κεφάλαιο, όπου μελετάται το σύστημα triggering του NSW.

⁸Αν όχι άμεσα, τουλάχιστον με μία σταθερή χρονική διαφορά.

με τον integral discriminator, είναι ότι ο SCA έχει παράθυρο λειτουργίας και όχι απλό κατώτατο όριο. Δηλαδή, ο SCA παράγει λογικό παλμό αν το σήμα εισόδου ξεπερνάει ένα κατώτατο όριο, αλλά δεν ξεπερνάει και ένα άλλο, ανώτατο όριο. Τα όρια αυτά, όπως και πριν, μπορούν να οριστούν από το χρήστη κατά βούληση. Η λειτουργία του απεικονίζεται στο Σχήμα 3.3.IX.



Σχήμα 3.3.IX: Η βασική λειτουργία του SCA. Το πρώτο σήμα δεν ξεπερνάει το κατώτατο όριο, το τρίτο ξεπερνάει το κατώτατο, αλλά και το ανώτατο. Μόνο το δεύτερο ανήκει στο εύρος λειτουργίας (παράθυρο) του διευκρινιστή. [4].

Οι διατάξεις αυτές είναι καίριας σημασίας για την αλυσίδα επεξεργασίας των σημάτων ενός ανιχνευτή, καθώς η έξοδός τους, λειτουργεί σαν είσοδος για τους μετρητές και για τους πολυκαναλικούς αναλυτές, οι οποίοι καταμετρούν τα γεγονότα, ή προβαίνουν σε λειτουργίες φασματοσκοπίας αντίστοιχα.

Counters

Οι μετρητές είναι απλές ηλεκτρονικές συσκευές που δέχονται τους λογικούς παλμούς από τους διευκρινιστές και τους καταμετρούν. Μπορεί για παράδειγμα να αποτελείται από έναν απλό ψηφιακό καταχωρητή (*register*), του οποίου η αποθηκευμένη τιμή να αυξάνεται κατά ένα για κάθε λογικό παλμό που του στέλνεται από τον discriminator. Πιο συγκεκριμένα, ένας τρόπος λειτουργίας για τον counter, είναι να συσσωρεύει παλμούς στην είσοδό του, μέχρι ένα σημείο κορεσμού, όπου η περίοδος συσσώρευσης τερματίζεται. Ένας εξωτερικός χρονομετρητής έχει μετρήσει ανεξάρτητα τη χρονική διάρκεια αυτής της διαδικασίας, από τον πρώτο μέχρι και τον τελευταίο παλμό, και έτσι υπολογίζεται ο ρυθμός καταμέτρησης. Το πλεονέκτημα αυτής της μεθόδου έγκειται στο γεγονός ότι μπορεί να παράξει στατιστικώς αποδεκτά αποτελέσματα εύκολα, καθώς το σημείο του κόρου προγραμματίζεται από

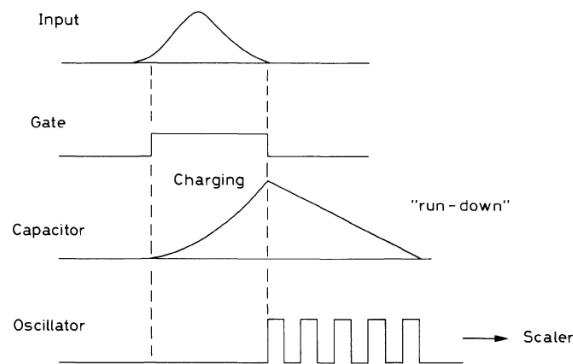
το χρήστη αναλόγως. Το μειονέκτημα από την άλλη, είναι η ύπαρξη του εξωτερικού χρονομετρητή, ο οποίος συνήθως υλοποιείται από ένα κύκλωμα κρυστάλλου που ταλαντώνεται (ο πιο συνήθης τρόπος δημιουργίας ψηφιακού ρολογιού). Αν για παράδειγμα η ποιότητα του κρυστάλλου είναι κατώτερη, τυχόν θερμοκρασιακές μεταβολές μπορεί να επηρεάσουν την περίοδο ταλάντωσής του, άρα και την περίοδο του παλμού που παράγει [5].

Analog to Digital Converters (ADC)

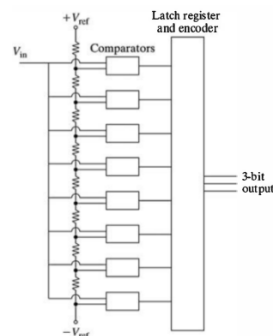
Κάποια στιγμή κατά τη διάρκεια επεξεργασίας των παλμών που προκύπτουν από έναν ανιχνευτή, θα χρειαστεί να μετατραπούν τα αναλογικά σήματα σε ψηφιακά δεδομένα. Ο κωδικοποιητής από αναλογικό σε ψηφιακό (ADC), είναι επομένως ένα πολύ σημαντικό υποσύστημα, που δρα ως ο συνδετικός κρίκος μεταξύ αναλογικών και ψηφιακών (ή λογικών) κυκλωμάτων. Αν για παράδειγμα ένας διαμορφωτής παλμών παράγει σήματα τάσης από 0 έως 10 V τα οποία και στέλνει σε έναν ADC, στην περίπτωση που η έξοδος του ADC έχει μήκος 10 bits, τότε μπορεί να εκφράσει σε δεκαδικό τους αριθμούς από το 0 έως το 1023. Αν δεχτεί 10 V στην είσοδο, θα παράξει το δυαδικό αντίστοιχο του αριθμού 1000, αν δεχτεί 3 V θα στείλει το 300 σε δεκαδικό κ.ο.κ.

Μία μέθοδος με την οποία ο ADC μετράει το μέγιστο ύψος ενός παλμού είναι η μέθοδος Wilkinson [4]. Αυτή η τεχνική περιλαμβάνει έναν πυκνωτή ο οποίος φορτίζεται από τον παλμό εισόδου. Όταν ξεκινάει η εκφόρτιση του πυκνωτή, ένας μετρητής συσσωρεύει τιμές στην ψηφιακή μνήμη του, με τη χρήση ενός εξωτερικού χρονομέτρου. Όταν ο πυκνωτής εκφορτιστεί πλήρως, η καταμέτρηση σταματάει, και προφανώς, όση περισσότερη ώρα διαρκούσε η εκφόρτιση, τόσο μεγαλύτερη ήταν και η μέγιστη τάση του παλμού εισόδου όπως και η τιμή που μετρήθηκε από το κύκλωμα του μετρητή. Η τιμή αυτή διοχετεύεται στην έξοδο του ADC. Η μέθοδος περιγράφεται γραφικά στο Σχήμα 3.3.X. Μία άλλη ευρέως διαδεδομένη τεχνική μετατροπής αναλογικού σήματος σε ψηφιακό, είναι η υλοποίηση ενός παράλληλου ή flash ADC. Αυτός ο κωδικοποιητής, επιστρατεύει μία συστοιχία από κυκλώματα σύγκρισης (comparators) συνδεδεμένα παράλληλα μεταξύ τους με κοινή τάση αναφοράς. Οι comparators συγκρίνουν την τάση εισόδου με μία ονομαστική τάση αναφοράς, ορισμένη αυστηρά για κάθε κύκλωμα σύγκρισης. Στην περίπτωση που ο παλμός εισόδου υπερκαλύπτει την τάση αναφοράς, εγγράφεται το λογικό μηδέν

από τον comparator σε έναν καταχωρητή. Λόγω της παράλληλης συνδεσμολογίας, οι comparators δρουν όλοι ταυτόχρονα, και αφού ολοκληρωθεί η σύγκριση, διαβάζεται η εγγραφή στη μνήμη του καταχωρητή, ο οποίος μετατρέπει τη συμβολοσειρά της μνήμης του σε αντίστοιχες τιμές στα bits της εξόδου. Αν και πρόκειται για πολύ γρήγορη τεχνική, το μεγάλο μειονέκτημά της, έγκειται στην απαίτηση για πολλά κυκλώματα σύγκρισης (2^n comparators για n bits εξόδου), γεγονός που περιορίζει το πλήθος των bits εξόδου, άρα και τη διακριτική ικανότητα (resolution) του κωδικοποιητή. Στο Σχήμα 3.3.XI παρατίθεται μία αναπαράσταση του κυκλώματος ενός flash ADC.



Σχήμα 3.3.X: Η μέθοδος Wilkinson. Το σήμα εισόδου φορτίζει έναν πυκνωτή, του οποίου ο χρόνος εκφόρτισης υπολογίζεται από ένα κύκλωμα καταμέτρησης (scaler) [4].



Σχήμα 3.3.XI: Ένας flash ADC με έξοδο τριών bit και οκτώ comparators [5].

Multichannel Analyzers (MCA)

Οι πολυκαναλικοί αναλυτές (Multichannel Analyzers (MCA)) είναι περίπλοκες συσκευές που κατηγοριοποιούν τους παλμούς της εισόδου τους σε κανάλια σύμφωνα με το πλάτος τους, ενώ καταχωρούν ταυτόχρονα και το συνολικό πλήθος κάθε κατηγορίας. Το περιεχόμενο όλων των καναλιών (ύψος παλμού και σύνολο παλμών που αντιστοιχεί σε κάθε ύψος) δίνει ένα πλήρες φάσμα ενεργειών όπως αυτό ανιχνεύθηκε [4]. Η καρδιά αυτών των υποσυστημάτων βρίσκεται στον ADC τους, ο οποίος μετατρέπει τον παλμό εισόδου σε ψηφιακό σήμα, που είναι ουσιαστικά μία συμβολοσειρά από bits που θα αντιστοιχούν σε μία διεύθυνση μνήμης. Μόλις καταχωρηθεί ο παλμός στο κανάλι αυτό, αυξάνεται και η τιμή του μετρητή που είναι συνυφασμένος με το κανάλι.

Προφανώς, η αποδοτικότητα του ADC είναι καίριας σημασίας για την εύρυθμη λειτουργία του MCA. Σε τέτοιες εφαρμογές, ο ADC συνήθως έχει τη δυνατότητα (εκτός από το να καταγράφει με ακρίβεια το ύψος των παλμών που δέχεται) να εντοπίζει άμεσα ότι κατέφτασε ένας παλμός που αντιστοιχεί σε γεγονός που ανιχνεύθηκε (peak sensing ADC). Καθ' όλη τη διάρκεια επεξεργασίας του παλμού αυτού, ο ADC δεν μπορεί να δεχθεί άλλα σήματα, επομένως υπάρχει ένας νεκρός χρόνος όπου χάνονται γεγονότα (dead-time). Ο χρόνος αυτός, χρησιμοποιώντας ένα εξωτερικό ψηφιακό ρολόι, μπορεί να μετρηθεί με ακρίβεια, δίνοντας έτσι τη δυνατότητα στο χρήστη να υπολογίσει πόσο *πραγματικό* χρόνο (δηλαδή χρόνο που μπορεί να καταγράφει ο MCA, ο οποίος ονομάζεται live-time) θέλει να αφήσει στον MCA για να καταγράψει ένα φάσμα [5].

Triggering

Πέρα από τα κυκλώματα που η κύρια λειτουργία τους είναι να διαμορφώνουν τους παλμούς του ανιχνευτή και να μετρούν το ύψος τους, άρα και την ενέργεια που εναποτέθηκε, υπάρχουν και κυκλώματα εξειδικευμένα στο να καταγράφουν *χρονικές πληροφορίες* για τα σήματα που προκύπτουν από ένα ανιχνευτικό σύστημα. Η πιο θεμελιώδης μεθοδολογία μέτρησης χρόνου, είναι η παραγωγή ενός λογικού παλμού, το leading edge του οποίου υποδεικνύει ακριβώς τη στιγμή που ένα αναλογικό σήμα καταγράφηκε από το σύστημα. Τα κυκλώματα που υλοποιούν αυτή τη λειτουργία ονομάζονται *μονάδες καταγραφής χρόνου (time pick-off units)*, ή *triggers* [5].

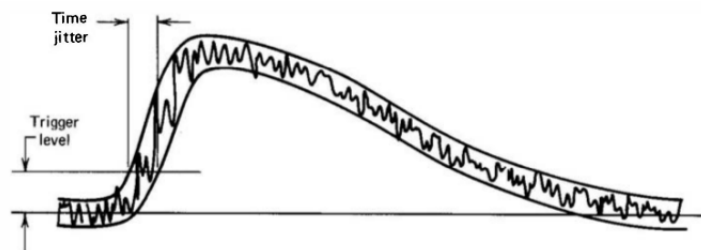
Η αποδοτικότητα της καταγραφής του πότε, δεν εξαρτάται μόνο από την ποιότητα των ηλεκτρονικών που αναλαμβάνουν αυτή τη δουλειά, αλλά και από τα χαρακτηριστικά του ίδιου του ανιχνευτή. Για παράδειγμα, ανιχνευτές που παράγουν πολυάριθμους φορείς πληροφορίας (ζεύγη ιόντων-ηλεκτρονίων για τον MicroMegas), παρουσιάζουν συνήθως πολύ καλή χρονική διακριτική ικανότητα (timing resolution). Σημαντικό ρόλο παίζει επίσης και το εύρος των πλατών των παλμών που δημιουργούνται από τους αισθητήρες του ανιχνευτή. Αν τα πλάτη κινούνται σε μικρά εύρη, τότε πολλές διαφορετικές τεχνικές μέτρησης χρόνου δίνουν αποδεκτά αποτελέσματα. Από την άλλη, αν οι μέγιστες τάσεις που εμφανίζονται καλύπτουν ένα πιο μεγάλο φάσμα πιθανών τιμών, τότε σχεδόν πάντα το timing resolution είναι κατώτερης ποιότητας.

Ένας ακόμα σημαντικός παράγοντας που επηρεάζει αρνητικά την αποδοτικότητα της μέτρησης χρόνου σε ένα σύστημα ηλεκτρονικών είναι εκείνος του jitter. Το jitter είναι ένα είδος ηλεκτρονικού θορύβου, στενά συνυφασμένο με τη φάση της κυματομορφής [18]. Ένα περιοδικό σήμα για παράδειγμα, αναλογικό ή ψηφιακό, εμφανίζει jitter όταν οι δεσμοί της κυματομορφής του (τα σημεία εκείνα που τέμνουν τη baseline, ή την τάση αναφοράς) δεν ισαπέχουν μεταξύ τους αλλά η μεταξύ τους απόσταση διακυμαίνεται τυχαία. Με άλλα λόγια το σήμα δεν είναι εκεί που οφείλει να είναι, τη στιγμή που πρέπει. Αυτές οι ελάχιστες διακυμάνσεις στη φάση των παλμών μπορεί να οδηγήσουν σε σημαντικές ανακρίβειες στη μέτρηση χρονικών ιδιοτήτων. Η πηγή του jitter, μπορεί να εντοπιστεί σε μικρές αυξομειώσεις της έντασης και του γενικότερου σχήματος ενός παλμού, όπως αυτός προκύπτει από τους αισθητήρες, τους προενισχυτές, και γενικότερα από όλα τα προαναφερθέντα ηλεκτρονικά υποσυστήματα μίας ανιχνευτικής διάταξης. Αν για παράδειγμα οι φορείς φορτίου που παράγουν τα σήματα στον ανιχνευτή είναι σχετικά λίγοι στον αριθμό, η κατά συνέπεια σημαντική στατιστική τους διακύμανση θα οδηγήσει σε παραγωγή παλμών ασταθούς μορφολογίας, γεγονός που θα συνεισφέρει αργότερα στο jitter των προς επεξεργασία σημάτων.

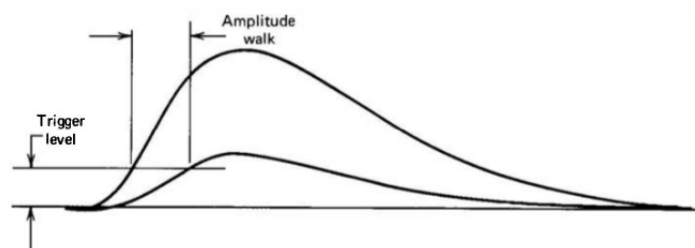
Ανάλογα τις δυσκολίες που ανακύπτουν λόγω θορύβου, και ανάλογα τη φύση των παλμών που διαχειρίζεται ένα σύστημα ηλεκτρονικών, θα χρησιμοποιηθεί και διαφορετική τεχνική καταγραφής χρόνου (time pick-off). Αναφορικά, η συνηθέστερη μέθοδος, εκείνη του Leading edge triggering (που είναι και η απλούστερη), ενεργοποιείται αν ένας παλμός ξεπεράσει κάποιο αυστηρά ορισμένο κατώτατο όριο. Η

μέθοδος αυτή όμως, είναι επιρρεπής σε ανακρίβειες αν η μορφή των παλμών παρουσιάζει σημαντικές διακυμάνσεις, ή αν το jitter είναι υπερβολικά έντονο. Το πρόβλημα γίνεται εμφανές στα δύο σχήματα που παρατίθενται παρακάτω (Σχήματα 3.3.XII και 3.3.XIII).

Οι λεπτές ισορροπίες που διέπουν τη διαδικασία ορθού καθορισμού των triggers, έχουν οδηγήσει σε ελαφρώς διαφορετικές τεχνικές προσδιορισμού του χρόνου εμφάνισης ενός παλμού οι οποίες όμως δεν θα αναφερθούν εδώ. Αξίζει όμως να σημειωθεί πως η βελτιστοποίηση της μεθόδου του triggering είναι από τα πιο σημαντικά κομμάτια της διαδικασίας σχεδιασμού ενός ηλεκτρονικού συστήματος που επεξεργάζεται τις πληροφορίες των ανιχνευτών, γεγονός που θα γίνει εμφανές στο επόμενο Κεφάλαιο, όπου ανάμεσα στα άλλα θα μελετηθεί και το σύστημα read-out για την αναβάθμιση του NSW.



Σχήμα 3.3.XII: Η επιρροή του time jitter στην παραγωγή του trigger. Ο λογικός παλμός του trigger καθυστερεί να εμφανιστεί ή εμφανίζεται συντομότερα από το κανονικό, λόγω των μικροδιακυμάνσεων από το jitter [5].

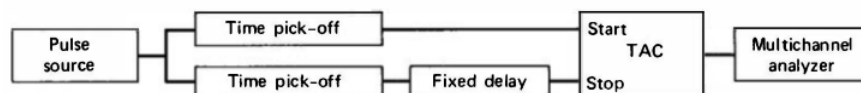


Σχήμα 3.3.XIII: Η επιρροή του σχήματος του παλμού στην παραγωγή του trigger. Δύο παλμοί που εμφανίζονται την ίδια στιγμή, δίνουν trigger σε εντελώς διαφορετικές χρονικές στιγμές, καθώς ο ένας έχει πολύ πιο μικρό rise time από τον άλλο (πιο απότομο leading edge) με αποτέλεσμα να φτάνει στο κατώφλι νωρίτερα [5].

Time-to-Amplitude Converters (TAC)

Το τελευταίο ηλεκτρονικό υποσύστημα που θα μελετηθεί, είναι πολύ σημαντικό και όπως θα διαπιστωθεί αργότερα, παίζει καθοριστικό ρόλο και στα σύστημα read-out του NSW. Πρόκειται για τους μετατροπείς χρόνου-σε-πλάτος ή TAC. Το ηλεκτρονικό αυτό υποσύστημα, παράγει έναν αναλογικό παλμό, το μέγιστο πλάτος του οποίου είναι ανάλογο της χρονικής διαφοράς μεταξύ δύο σημάτων (Start και Stop) που δέχεται στην είσοδό του [4, 5]. Ο παλμός της εξόδου μπορεί να αναλυθεί περαιτέρω από έναν MCA, ο οποίος θα παράξει τελικά ένα φάσμα χρόνων. Επίσης, ο παλμός αυτός μπορεί να οδηγηθεί άμεσα σε έναν ADC, ο οποίος θα στείλει την ψηφιακή πληροφορία σε άλλο υποσύστημα υπεύθυνο για τη χρονική ανάλυση των παλμών.

Η μέθοδος λειτουργίας του TAC βασίζεται σε έναν πυκνωτή, ο οποίος αρχικά είναι πλήρως φορτισμένος. Αν ένας παλμός περάσει στον TAC, τότε παράγεται ένα λογικό σήμα trigger (που μπορεί να έχει παραχθεί από έναν leading edge trigger για παράδειγμα) το οποίο είναι το Start. Ο πυκνωτής τότε ξεκινάει να εκφορτίζεται, και συνεχίζει να το κάνει μέχρι να λάβει το σήμα Stop (το οποίο για παράδειγμα μπορεί να το δίνει ένας εξωτερικός counter που μετράει μία συγκεκριμένη περίοδο, ή γενικώς ένα κύκλωμα που παράγει ένα καθυστερημένο σήμα σε ορισμένο χρόνο). Το συνολικό φορτίο που θα συσσωρευτεί από την εκφόρτιση, θα είναι ανάλογο του χρονικού διαστήματος μεταξύ των δύο σημάτων που ξεκίνησαν και σταμάτησαν την εκφόρτιση. Μετά το Stop, ο πυκνωτής επαναφορτίζεται και είναι έτοιμος για τον επόμενο κύκλο μετρήσεων. Ο παλμός της εξόδου, το μέγιστο πλάτος του οποίου θα υποδεικνύει το χρόνο, διοχετεύεται συνήθως σε έναν ADC ο οποίος μετατρέπει την πληροφορία της έντασης του παλμού σε ψηφιακό σήμα, έτοιμο για επεξεργασία ή αποθήκευση. Μία απλοποιημένη αναπαράσταση δίνεται στο Σχήμα 3.3.XIV.



Σχήμα 3.3.XIV: Απλοποιημένο σχεδιάγραμμα συστήματος TAC. Ένας trigger δίνει το σήμα του Start, και το Stop δίνεται από ένα σήμα που καθυστερεί να εμφανιστεί για ορισμένο χρονικό διάστημα. Ο προκύπτον παλμός εδώ διοχετεύεται σε έναν MCA για κατηγοριοποίηση [5].

Σύνοψη

Στο Κεφάλαιο αυτό, πραγματοποιήθηκε μία εκτενέστερη ανάλυση των ηλεκτρονικών συστημάτων που διαβάζουν τα σήματα ενός ανιχνευτή. Οι δυσκολίες που συναντάει κανείς κατά το σχεδιασμό ενός τέτοιου μηχανισμού είναι πολλές και σημαντικές. Επίσης, η επίλυση ενός συγκεκριμένου προβλήματος, μπορεί να οδηγήσει στην εμφάνιση ή και στην όξυνση κάποιου άλλου, με αποτέλεσμα αυτό που συμβαίνει συνήθως είναι να προβαίνει κανείς σε συμβιβασμούς όταν υλοποιεί τέτοια περίπλοκα συστήματα. Ζητήματα όπως ο ηλεκτρονικός θόρυβος, η ανομοιογένεια στη μορφολογία των παλμών, τα παρασιτικά κυκλώματα στις διασυνδέσεις και το jitter, είναι μερικά από τα ανεπιθύμητα φαινόμενα που κάνουν αυτή τη διαδικασία τόσο δύσκολη και επίπονη. Στη συνέχεια, θα παρουσιαστεί ένα τέτοιο read-out system για την αναβάθμιση του NSW, μαζί με το υπόλοιπο σύνολο των ηλεκτρονικών που είναι υπεύθυνο για την αναδιανομή των ψηφιακών δεδομένων, το οποίο είναι και το κομμάτι που πραγματεύεται έμπρακτα η παρούσα διπλωματική εργασία.

4

Το Σύστημα Ηλεκτρονικών του New Small Wheel

Σε αυτό το Κεφάλαιο, αρχικά θα παρουσιαστούν μερικές λεπτομέρειες για δύο οικογένειες εξελιγμένων ολοκληρωμένων κυκλωμάτων τα οποία τελούν τις εργασίες που περιγράφηκαν στο Κεφάλαιο 3 και συμπεριλαμβάνονται στο σύστημα ηλεκτρονικών του New Small Wheel (NSW). Στη συνέχεια, αφού δοθεί μία γενική επισκόπηση των πλακετών που θα χρησιμοποιηθούν στα πλαίσια της αναβάθμισης του ανιχνευτή ATLAS, θα μελετηθεί η κάθε πλακέτα ξεχωριστά, με έμφαση στην κάρτα L1DDC, τη σχεδίαση και τον προγραμματισμό της οποίας ανέλαβε η ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών του Ε.Μ.Π. Αξίζει επίσης να σημειωθεί ότι στο παρόν κεφάλαιο μελετώνται κυρίως τα ηλεκτρονικά των ανιχνευτών MicroMegas (MM) (αν και δεν υπάρχουν σημαντικές διαφορές σε σχέση με το σύστημα των sTGC).

4.1 Εισαγωγή στα ASIC και στα FPGA

Ανατρέχοντας στην ενότητα 3.3, μπορεί κανείς να μελετήσει τη δομή των ηλεκτρονικών υποσυστημάτων που τελούν τις διεργασίες της επεξεργασίας των πρωταρχικών σημάτων, όπως αυτά προκύπτουν από τους αισθητήρες ενός ανιχνευτή. Παράλληλα λοιπόν με την κατασκευή πολλών διαφορετικών και πολύπλοκων ανιχνευτών, εμφανίστηκε η ανάγκη για παγίωση των τεχνικών με τις οποίες συλλέγονται τα δεδομένα από αυτούς. Για το λόγο αυτό, στα τέλη της δεκαετίας του '60 αναπτύχθηκαν

τα συστήματα NIM και CAMAC¹ [4]. Τα δύο αυτά πρότυπα συλλογής δεδομένων, είναι ουσιαστικά η υλοποίηση των read-out electronics σε μορφή αυτοτελών μονάδων (modules), με αυστηρά καθιερωμένες προδιαγραφές. Οι μονάδες αυτές, που διαθέτουν συγκεκριμένη συνδεσμολογία, τοποθετούνται σε εξειδικευμένα crates, τα οποία διανέμουν τα σήματα εισόδου στα NIM και CAMAC modules. Έτσι η επεξεργασία των σημάτων προτυποποιήθηκε, και η σχεδίαση των ηλεκτρονικών για ανιχνευτές πέρασε στα χέρια εταιριών που κατασκεύαζαν τα συστήματα NIM και CAMAC.



Σχήμα 4.1.1: Μερικά NIM modules [4].

Με το πέρασμα των χρόνων όμως, ήρθε και η αύξηση της πολυπλοκότητας των ανιχνευτικών διατάξεων· οι εγγενείς περιορισμοί των συστημάτων αυτών άρχισε λοιπόν να γίνεται έκδηλη. Ευτυχώς, τη λύση στο πρόβλημα ήρθε να λύσει η ανάπτυξη στον τομέα της μικροηλεκτρονικής, η οποία άνοιξε το δρόμο για την κατασκευή *Ολοκληρωμένων Κυκλωμάτων (Integrated Circuits)*, με πραγματικές διαστάσεις μερικών εκατοστών, που όμως στο εσωτερικό τους έκρυβαν εκατομμύρια τρανζίστορ. Με την τεχνολογία CMOS², γεννήθηκε η δυνατότητα κατασκευής μικροηλεκτρονικών διατάξεων, ικανών να εκτελέσουν εκατομμύρια διεργασίες, μέσω ενός και μοναδικού chip. Έτσι, εκτός από τους ισχυρούς μικροεπεξεργαστές (Central Processing Unit (CPU)) που χρησιμοποιούνται σε κάθε προσωπικό υπολογιστή σήμερα, αναπτύχθηκαν και τα ASIC και FPGA, δύο οικογένειες ολοκληρωμένων κυκλωμάτων

¹NIM: Nuclear Instrument Module, CAMAC: Computer Automated Measurement And Control.

²CMOS: Complementary Metal Oxide Semiconductors. Πρόκειται για έναν συγκεκριμένο τρόπο σχεδίασης ηλεκτρονικών ψηφιακών κυκλωμάτων, που χρησιμοποιεί τρανζίστορ μεγέθους μερικών *nm* για να υλοποιήσει οποιοδήποτε λογικό κύκλωμα.

που βρίσκουν σημαντικές εφαρμογές στα πειράματα Φυσικής Υψηλών Ενεργειών.

4.1.1 Application-Specific Integrated Circuits (ASIC)

Τα *Ολοκληρωμένα Κυκλώματα Συγκεκριμένων Εφαρμογών (Application-Specific Integrated Circuits (ASICs))*, είναι εξειδικευμένες διατάξεις μικροηλεκτρονικής, σχεδιασμένες με τέτοιο τρόπο ώστε να εκτελούν συγκεκριμένες διεργασίες, τις οποίες ορίζει ο σχεδιαστής τους. Ήδη από τη δεκαετία του '80, πολλές από τις λειτουργίες των συστημάτων NIM και CAMAC, άρχισαν να τελούνται από τα ASIC [5]. Τα chip αυτά, αν σχεδιαστούν σωστά, μπορούν να προβούν σε ενίσχυση σημάτων, σύγκριση τάσεων, κωδικοποίηση από αναλογικό σε ψηφιακό, και πολλά άλλα. Πλέον, σε μεγάλα πειράματα, όλες οι διαδικασίες που έχουν περιγραφεί στο προηγούμενο Κεφάλαιο, εκτελούνται ως επί το πλείστον από ASIC.

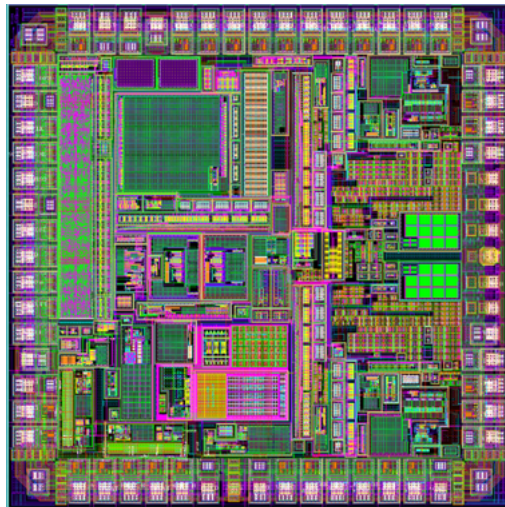
Όπως μπορεί να μαντέψει κανείς βέβαια, η διαδικασία σχεδίασης ενός ASIC που θα χρησιμοποιηθεί σε ένα πείραμα είναι επίπονη και χρονοβόρα. Το γεγονός αυτό δεν οφείλεται μόνο στις δυσκολίες σχεδίασης που εμφανίζονται κατά την ανάπτυξη συστημάτων ηλεκτρονικών για read-out, οι οποίες έχουν αναλυθεί στο προηγούμενο Κεφάλαιο, αλλά και στο ότι η αρχιτεκτονική αυτών των chip είναι εξαιρετικά περίπλοκη, και η σωστή ανάπτυξη ενός τέτοιου ολοκληρωμένου κυκλώματος από το μηδέν, απαιτεί μία μεγάλη ομάδα εξειδικευμένου προσωπικού. Πιο συγκεκριμένα [5], τα ASIC χρησιμοποιούνται σε μία ανιχνευτική διάταξη όταν:

- Το πλήθος των καναλιών³ του ανιχνευτή είναι πολυάριθμα (πολλές χιλιάδες).
- Η ενέργεια που πρέπει να καταναλώνεται είναι περιορισμένη (π.χ. όταν ανά κανάλι πρέπει να δαπανάται ισχύς το πολύ της τάξης των 10 mW).
- Υπάρχει περιορισμός χώρου.

Είναι γεγονός ότι και οι τρεις άνω λόγοι, ισχύουν για την περίπτωση της αναβάθμισης του New Small Wheel. Κατ' αρχάς, το συνολικό πλήθος των καναλιών, ξεπερνάει τα 2 εκατομμύρια [9]. Επίσης, λόγω του μεγέθους του πειράματος, είναι φανερό ότι οποιαδήποτε μείωση στην ισχύ που απαιτείται για την εκτέλεση του πειράματος (η

³Με τον όρο "κανάλι" εννοείται ο δίαυλος επικοινωνίας μεταξύ ενός αισθητήρα (ενός strip για την περίπτωση του MicroMegas (βλ. εν. 2.2), και του συστήματος ανάλυσης σημάτων

ενέργεια που καταναλώνουν οι μαγνήτες που επιταχύνουν και κατευθύνουν τις δέσμες στον LHC είναι από μόνη της αγγίζει τα 120 MW) είναι ευπρόσδεκτη. Επίσης, όταν τα κανάλια είναι τόσα πολλά, είναι φανερό ότι οι διατάξεις που θα επεξεργάζονται τα σήματα θα πρέπει αντίστοιχα να είναι μικρά σε μέγεθος, γιατί πολύ απλά...δεν θα χωράνε να τοποθετηθούν κοντά στον ανιχνευτή. Τέλος, αξίζει να σημειωθεί ότι υπάρχουν τεχνικές ανάπτυξης ASIC, οι οποίες επιτρέπουν στα chip που θα κατασκευαστούν να είναι πλήρως ανεκτικά ως προς την ακτινοβολία *radiation tolerant*. Αυτή η ιδιότητα είναι πολύ χρήσιμη για τα πειράματα στο CERN, καθώς οι συνθήκες που επικρατούν κοντά στις αλληλοσυγκρουόμενες δέσμες, είναι τουλάχιστον αφιλόξενες για τα συστήματα ηλεκτρονικών (περίπου 1700 Gy για μεγάλες τιμές του η [11]). Το πρόβλημα το δημιουργούν ως επί το πλείστον τα αδρόνια (κυρίως τα νετρόνια) μεγάλης ενέργειας ($E > 20\text{ MeV}$), που προκαλούν στα τρανζίστορ *Single Event Upsets (SEU)*⁴. Για την αναβάθμιση του NSW, όπως θα διαπιστωθεί και στη συνέχεια, έχουν σχεδιαστεί και κατασκευαστεί αρκετά ASIC, τα οποία όλα πληρούν τις προδιαγραφές του πειράματος ATLAS.



Σχήμα 4.1.Π: Η μήτρα ενός ASIC.

⁴SEU: Όταν μία ιοντίζουσα ακτινοβολία αλληλεπιδρά με τα δομικά μέλη ενός chip, προκαλώντας σφάλματα στους υπολογισμούς.

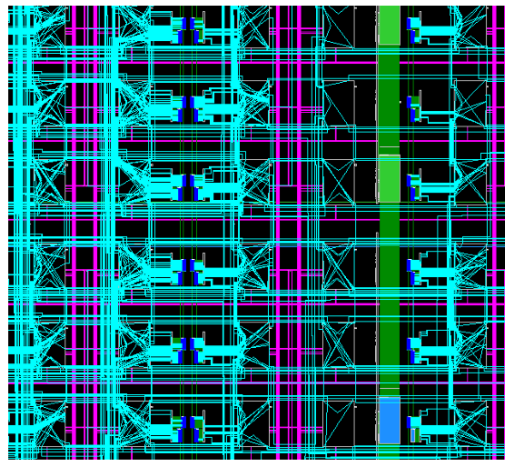
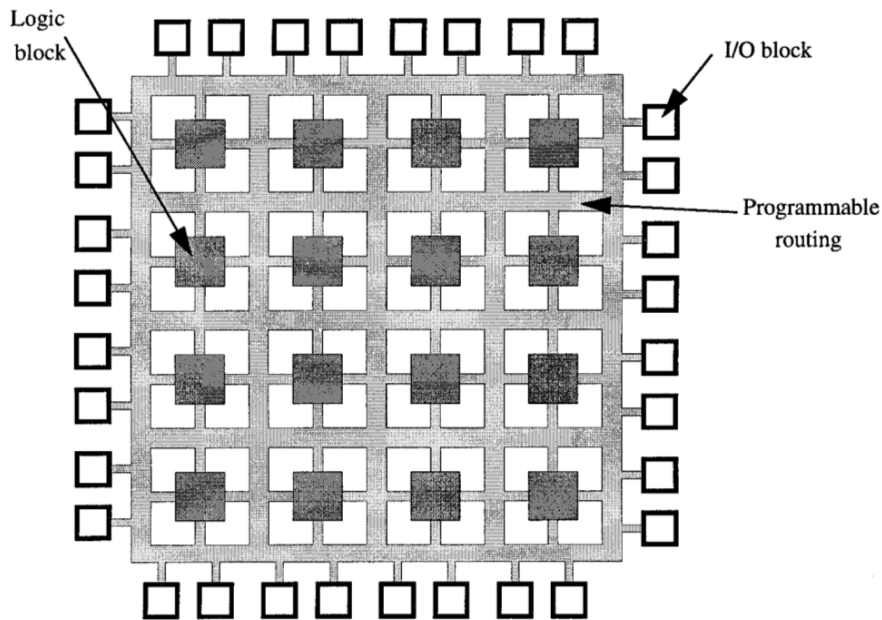
4.1.2 Field-Programmable Gate Arrays (FPGA)

Οι Προγραμματιζόμενες Συστοιχίες Πυλών, *Field-Programmable Gate Arrays (FPGA)*, είναι ολοκληρωμένα κυκλώματα που αποτελούνται από χιλιάδες πανομοιότυπα Προγραμματιζόμενα Λογικά Μπλοκ *Configurable Logic Blocks (CLB)*, τα οποία είναι συνδεδεμένα μεταξύ τους μέσω πολυάριθμων διαύλων. Οι διάυλοι αυτοί, μπορούν και αυτοί να ενεργοποιούνται και να απενεργοποιούνται κατά τη βούληση του σχεδιαστή-προγραμματιστή, δημιουργώντας έτσι ένα δυναμικό σύνολο, που μπορεί να τελεί οποιαδήποτε λογική συνάρτηση. Τα σημερινά FPGA επιστρατεύουν εκατομμύρια λογικά μπλοκ σε ένα και μόνο ολοκληρωμένο κύκλωμα, το οποίο επικοινωνεί με την πλακέτα στην οποία βαίνει με εκατοντάδες, ή και χιλιάδες I/O Pins⁵. Το σημαντικότερο πλεονέκτημα ενός FPGA έναντι όλων των υπολοίπων ολοκληρωμένων κυκλωμάτων λοιπόν, είναι το γεγονός ότι μπορούν και προγραμματίζονται από τον χρήστη, μέσω γλωσσών οι οποίες περιγράφουν hardware (και όχι software, όπως οι πιο γνωστές γλώσσες π.χ. Java, C, FORTRAN). Το πρόγραμμα το οποίο παράγεται από μία τέτοια γλώσσα ονομάζεται *firmware*. Όλα τα FPGA, όταν βγουν από τη γραμμή παραγωγής, δεν εμπεριέχουν κανένα είδους firmware, και περιμένουν να τα προγραμματίσει ο χρήστης προκειμένου να τελέσουν κάποια λειτουργία. Ο προγραμματισμός αυτός δε, μπορεί να απορριφθεί και να αντικατασταθεί από κάποιον άλλο, για όσες φορές θελήσει ο χρήστης. Έτσι ο χαρακτήρας και η λειτουργικότητα του FPGA αλλάζει, ανάλογα το πρόβλημα στο οποίο εφαρμόζεται. Το πλεονέκτημα των FPGA σε σχέση με τα ASIC είναι ακριβώς αυτό: τα ASIC, επειδή προορίζονται για συγκεκριμένο σκοπό, από τη στιγμή του σχεδιασμού τους και έπειτα, δεν μπορούν να μεταβάλλουν τη λειτουργικότητά τους, ενώ τα FPGA χαρακτηρίζονται από μία συνολική ευελιξία. Από την άλλη όμως, τα ASIC καταναλώνουν γενικά λιγότερη ενέργεια από τα FPGA, ενώ αν σχεδιαστούν σωστά είναι ανεκτικά στην ακτινοβολία, ένα γεγονός που δεν ισχύει για τα FPGA.

Τα FPGA χρησιμοποιούνται και αυτά με τη σειρά τους κατά κόρον σε πειράματα Φυσικής Υψηλών Ενεργειών. Όπως θα διαπιστωθεί και στη συνέχεια, υπάρχουν πολυάριθμα FPGA σε διαφορετικά σημεία του συστήματος ηλεκτρονικών που σχεδιάζεται για την αναβάθμιση του NSW. Η παρούσα διπλωματική εργασία, πραγματεύεται τον προγραμματισμό των FPGA που είναι ενσωματωμένα σε κάποιες πλα-

⁵Input/Output pins. Οι είσοδοι/έξοδοι ενός ολοκληρωμένου κυκλώματος. Τα εξωτερικά σήματα διαβιβάζονται μέσα στο chip μέσω των pins.

κέτες των ηλεκτρονικών του NSW. Για το λόγο αυτό, για περισσότερες λεπτομέρειες σχετικά με την αρχιτεκτονική των FPGA, και πως αυτά προγραμματίζονται, ο αναγνώστης παραπέμπεται στο Κεφάλαιο 6, και στο αντίστοιχο Παράρτημα.



Σχήμα 4.1.ΠΙ: Πάνω: Σχηματική αναπαράσταση της αρχιτεκτονικής ενός FPGA. Είναι εμφανή τα λογικά μπλοκ, τα I/O pins, και οι δρομολογήσεις ανάμεσα από τα CLB [22]. Κάτω: Απεικόνιση των ενεργοποιημένων λογικών μπλοκ μαζί με τις διασυνδέσεις τους, έπειτα από τον προγραμματισμό ενός FPGA.

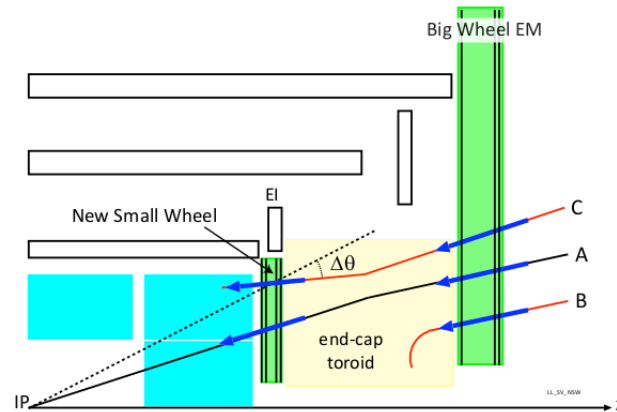
4.2 Γενική Επισκόπηση του Συστήματος Ηλεκτρονικών του NSW

Σε αυτή την ενότητα, θα μελετηθεί η γενικότερη δομή που διέπει ολόκληρο το σύστημα ηλεκτρονικών που θα υποστηρίξει την αναβάθμιση του New Small Wheel (NSW). Ανατρέχοντας στην ενότητα 1.4, ο αναγνώστης μπορεί να βρει τους λόγους για τους οποίους δημιουργήθηκε η ανάγκη αναβάθμισης του εσωτερικού δίσκου του μιονικού σπεκτρομέτρου στον ανιχνευτή ATLAS. Σε γενικές γραμμές, μπορεί να πει κανείς ότι μέσω παρατηρήσεων και υπολογισμών με τα μέχρι τώρα δεδομένα του ATLAS, έχει προβλεφθεί ότι μετά την αναβάθμιση του LHC, που συνεπάγεται αύξηση της φωτεινότητας άρα και της ροής των σωματιδίων στους ανιχνευτές, η γενική αποδοτικότητα του μιονικού σπεκτρομέτρου στα καπάκια του ανιχνευτή θα εκφυλιστεί. Πιο συγκεκριμένα, αναμένεται ότι με την αύξηση του ρυθμού αλληλεπιδράσεων λόγω των διαδοχικών αναβαθμίσεων, η χωρική διακριτική ικανότητα (spatial resolution) του παρόντος ανιχνευτή θα υποβαθμιστεί, ενώ το σύστημα του μιονικού triggering, που ήδη παρουσιάζει προβλήματα, σίγουρα θα κριθεί ως παραπάνω από ανεπαρκές. Επομένως, με την αναβάθμιση των ανιχνευτών του NSW, απαιτείται και ένα εξελιγμένο σύστημα ηλεκτρονικών το οποίο σε γενικές γραμμές οφείλει:

- Να συλλέγει τα δεδομένα του triggering και να τα στέλνει στο τμήμα επεξεργασίας trigger του CERN γρήγορα και αποδοτικά.
- Να αποθηκεύει την ενέργεια που εναποθέτουν τα σωματίδια στο σύστημα ανιχνευσης, μαζί με το πότε συνέβη αυτό.

Έτσι ορίζονται οι βασικοί άξονες λειτουργίας του συστήματος ηλεκτρονικών που βρίσκεται πάνω στους ανιχνευτές (*Front-end Electronics (FE)*): γρήγορη συλλογή δεδομένων για το triggering, ακριβής μέτρηση ενέργειας και χρόνου, και διανομή των δεδομένων αυτών στο σύστημα ηλεκτρονικών που παρεμβάλλεται μεταξύ του ανιχνευτή και του κέντρου υπολογιστών του CERN (*Back-end Electronics (BE)*). Επιπλέον, θα πρέπει να λαμβάνουν εντολές για αλλαγές του τρόπου λειτουργίας τους από το κέντρο ελέγχου (*control room*) του ATLAS, και να στέλνουν πίσω δεδομένα για εξωτερικούς παράγοντες (π.χ. θερμοκρασία) που επικρατούν στους ανιχνευτές (*configuration and monitoring*).

Ως προς τη διαδικασία του trigger, το νέο σύστημα που θα παράγει το triggering του NSW, σε συνεργασία με το σύστημα του Big Wheel, πρέπει να λειτουργεί με τέτοιο τρόπο ώστε να αποκλείει τα ψεύτικα trigger που προέρχονται από τροχιές που δε συμβαδίζουν με το σημείο αλληλεπίδρασης στο κέντρο του ανιχνευτή [8]. Η όλη διαδικασία περιγράφεται στο Σχήμα 4.2.I:



Σχήμα 4.2.I: Σχηματική αναπαράσταση του συστήματος trigger στα καπάκια του ATLAS μετά την αναβάθμιση του 2018. Ο ήδη υπάρχων Big Wheel, δέχεται όλες τις τροχιές σωματιδίων που απεικονίζονται. Με την εγκατάσταση του NSW, μόνο η περίπτωση 'A' γίνεται δεκτή καθώς μόνο αυτή περνάει και από τα δύο συστήματα triggering. Η περίπτωση 'B' θα απορριφθεί καθώς ο NSW δε θα βρει την τροχιά του σωματιδίου που πέρασε από εκείνο το σημείο του Big Wheel. Η τροχιά 'C' θα απορριφθεί επειδή συνδυάζοντας τις γωνίες και από τα δύο συστήματα, το σύστημα επεξεργασίας δεν καταλήγει σε μία ευθεία που να καταλήγει στο σημείο αλληλεπίδρασης [8].

Η διαδικασία επεξεργασίας του triggering περιλαμβάνει συλλογή δεδομένων για τα σημεία αλληλεπίδρασης και υπολογισμός των γωνιών (αξιμουθιακή ϕ και πολική $\Delta\theta$) που έχουν οι τροχιές σε σχέση με το σύστημα συντεταγμένων των ανιχνευτών (βλ. Σχ. 4.2.I). Αυτές οι γωνίες, υπολογίζονται τόσο από τους sTGC όσο και από τους MM, οι οποίοι μέσω των strip τους που έχουν πλάτος μόλις 0.5 mm , μπορούν να παρέχουν άμεσες και ακριβείς πληροφορίες που οδηγούν στον υπολογισμό γωνιών (με ακρίβεια $< 1 \text{ mrad}$) των τροχιών. Η διαδικασία υπολογισμού βέβαια, πρέπει να ολοκληρώνεται μέσα σε χρόνο 1025 ns από τη στιγμή που εντοπίζεται η πρώτη αλληλεπίδραση, ώστε να προλαμβάνει τα δεδομένα που θα προκύψουν από την ανάλυση του Big Wheel [8]. Με το συνδυασμό των πληροφοριών για τις τροχιές και από τα

δύο συστήματα, το μιονικό trigger θα μπορεί πλέον αν ανταπεξέρχεται στις περιστάσεις. Πέρα από τη βελτίωση του triggering, οι νέες τεχνολογίες των ανιχνευτών που θα χρησιμοποιηθούν στο NSW, αυξάνουν τη χωρική και διακριτική ικανότητα και βελτιστοποιούν τις μετρήσεις για την ενέργεια που εναποθέτουν τα σωματίδια στους ανιχνευτές. Λόγω αυτού του γεγονότος, η λεπτομερής ανακατασκευή των τροχιών, και ο υπολογισμός των ορμών (τα οποία πραγματώνονται σε μεταγενέστερα στάδια μετά την αποθήκευση των δεδομένων (off-line data analysis)), θα γίνεται με ακόμα μεγαλύτερη ακρίβεια μετά την αναβάθμιση του NSW.

Αντιλαμβάνεται λοιπόν κανείς το κεντρικό ρόλο των ηλεκτρονικών σε όλη αυτή την περίπλοκη διαδικασία ακριβούς και γρήγορου υπολογισμού του triggering και στην ανακατασκευή των τροχιών, τα οποία ηλεκτρονικά οφείλουν όχι μόνο να μεταδώσουν βέλτιστα τις σχετικές πληροφορίες, αλλά και να αντέξουν τις αντίξοες συνθήκες ακτινοβολήσης που θα επικρατούν πάνω στους ανιχνευτές όπου και θα βρίσκονται. Το σύστημα των front-end ηλεκτρονικών του NSW λοιπόν, που αυτή τη στιγμή βρίσκεται σε προηγμένα στάδια σχεδίασης και ανάπτυξης (η οποία θα ολοκληρωθεί το 2018 μαζί με την αναβάθμιση του ATLAS), απαρτίζεται από τρεις ηλεκτρονικές πλακέτες (Printed Circuit Boards (PCB)). Σε αυτό το σημείο θα δοθεί η γενική ιδέα της λειτουργίας των καρτών αυτών [8, 9, 10], ενώ παρακάτω παρατίθεται και μία σχηματική αναπαράσταση της μεταξύ τους συνδεσμολογίας (Σχήμα 4.2.ΙΙ):

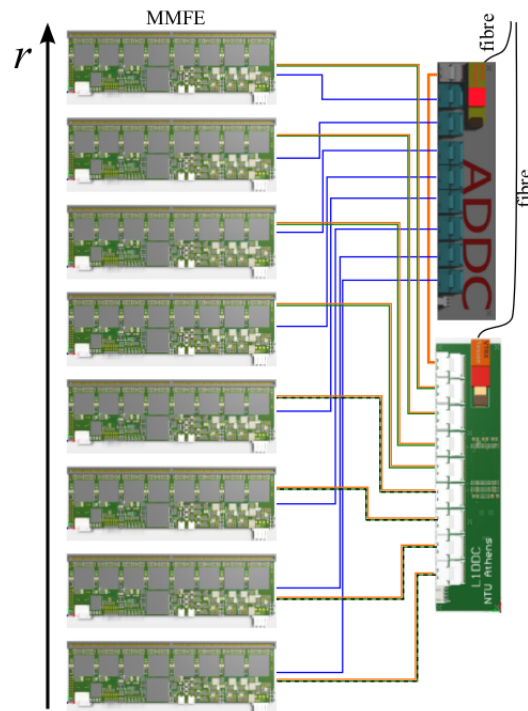
- **MicroMegas Front-End Board (MMFE, ή και MMFE8)**. Πρόκειται για την ηλεκτρονική πλακέτα που βρίσκεται πάνω στους ανιχνευτές και είναι υπεύθυνη για τη συλλογή των πρωταρχικών σημάτων από τους αισθητήρες. Κάθε κάρτα MMFE περιλαμβάνει οκτώ ASIC υπεύθυνα για το read-out, που ονομάζονται VMM. Κάθε VMM έχει 64 κανάλια, όπου το κάθε ένα αντιστοιχεί σε ένα strip του ανιχνευτή MicroMegas. Εκτός από τα VMM ASIC, στην πλακέτα αυτή βγαίνουν και άλλα δύο ASIC υπεύθυνα για την διανομή σημάτων monitoring/control και των ψηφιακών δεδομένων που προκύπτουν από την ανάλυση των παλμών. Συνολικά θα κατασκευαστούν 4096 κάρτες MMFE για την αναβάθμιση του NSW.
- **Level-1 Data Driver Card (L1DDC)**. Πρόκειται για την ηλεκτρονική πλακέτα που δέχεται τα σειριακά δεδομένα από οκτώ συνεχόμενες MMFE, και τα συμπύσσει σε ένα link οπτικής ίνας το οποίο καταλήγει στο σύστημα των back-end electronics και συγκεκριμένα στο δίκτυο FELIX (Front End Link eXchange).

Αυτό το επιτυγχάνει με τη χρήση ενός ASIC ονόματι *GBTx*. Εκτός από τη συλλογή και αποστολή δεδομένων για τους παλμούς που συλλέγει ο ανιχνευτής, η κάρτα αυτή είναι υπεύθυνη και στο να αποστέλλει τα δεδομένα για τη διαμόρφωση της λειτουργικότητας των MMFE (configuration signals/data), που προέρχονται από το κέντρο ελέγχου, και να παρέχει στις MMFE το σύστημα χρονισμού (BC clock). Η κάθε κάρτα L1DDC όμως, δε συνδέεται μόνο με τις οκτώ MMFE, αλλά και με την τρίτη πλακέτα, την ADDC που θα περιγραφεί στη συνέχεια. Η κάρτα L1DDC είναι αυτή που θα μελετηθεί εκτενέστερα αφού αποτελεί το αντικείμενο της παρούσας διπλωματικής εργασίας, καθώς εκτός από το *GBTx* περιλαμβάνει και ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα FPGA. Συνολικά θα κατασκευαστούν 1024 κάρτες L1DDC για την αναβάθμιση του NSW (512 για τους MicroMegas και 512 για τους sTGC).

- **ART Data Driver Card (ADDC)**. Πρόκειται για την ηλεκτρονική πλακέτα η οποία δέχεται τα δεδομένα ART (Address-in-real-time data) από τις οκτώ MMFE με τις οποίες συνδέεται. Διαθέτει και αυτή με τη σειρά της δύο *GBTx* ASIC, τα οποία συλλέγουν τα δεδομένα ART, και τα πακετάρουν για να τα στείλουν μέσω ενός link οπτικής ίνας στα back-end electronics, και συγκεκριμένα στο κομμάτι που επεξεργάζεται τα δεδομένα του triggering (*Trigger Processor*). Η κάρτα αυτή επικοινωνεί και με μία L1DDC, η οποία της στέλνει configuration data από το κέντρο ελέγχου, και της παρέχει σήματα χρονισμού. Συνολικά θα κατασκευαστούν 1024 κάρτες ADDC για την αναβάθμιση του NSW (512 για τους MicroMegas και 512 για τους sTGC).

Η επικοινωνία μεταξύ των καρτών γίνεται σειριακά, μέσω ζευγών διαφορικών γραμμών (καλωδίων) που ονομάζονται *e-links*. Το κάθε *e-link*, αποτελείται από τρία ζεύγη διαφορικών γραμμών (differential lines). Στη μία γραμμή μεταδίδεται το σήμα του ρολογιού (Clk+, Clk-), στην άλλη πραγματοποιείται η μετάδοση των δεδομένων (Dout+, Dout-), και στην τελευταία η λήψη των δεδομένων (Din+, Din-). Το κάθε *e-link* υποστηρίζει τρεις διαφορετικές ταχύτητες μετάδοσης δεδομένων (80, 160, 320 *Mbps*) οι οποίες μπορούν να οριστούν από το χρήστη. Το επίπεδο τάσεων των *e-link* ακολουθεί ως επί το πλείστον τα πρότυπα LVDS και SLVS [9, 10]. Περισσότερες πληροφορίες για τα LVDS/SLVS (όπως και για άλλα πρότυπα επικοινωνίας που εφαρμόζονται στα ηλεκτρονικά του NSW) μπορούν να βρεθούν στο επόμενο Κεφάλαιο.

Προκειμένου να γίνει μία λεπτομερέστερη ανάλυση των front-end electronics του NSW,



Σχήμα 4.2.Π: Σχηματική αναπαράσταση της συνδεσμολογίας μεταξύ των front-end electronics του NSW. Αριστερά είναι οι οκτώ πλακέτες MMFE, και στα δεξιά η κάρτα L1DDC συνδέεται με τις MMFE και την ADDC η οποία με τη σειρά της είναι και αυτή συνδεδεμένη με τις MMFE. Οι κάρτες L1DDC και ADDC συνδέονται μέσω οπτικής ίνας με το σύστημα των back-end electronics, και συγκεκριμένα με το δίκτυο FELIX και τον trigger processor [10].

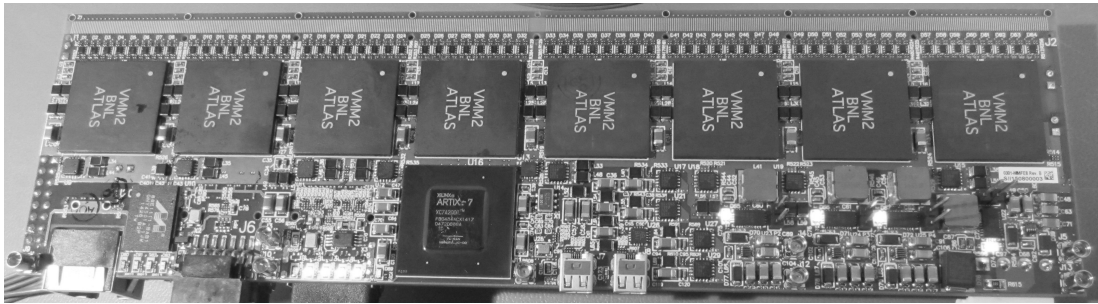
ώστε να ξεκαθαριστεί ο τρόπος λειτουργίας ολόκληρου του συστήματος, στη συνέχεια θα μελετηθεί η κάθε⁶ ηλεκτρονική πλακέτα ξεχωριστά.

4.3 MicroMegas Front-End Board

Η πλακέτα *MicroMegas Front-End Board (MMFE8)*, είναι εκείνη που συνδέεται με τον ανιχνευτή *MicroMegas*, και συλλέγει τα ηλεκτρικά σήματα (που είναι ουσιαστικά μία συγκέντρωση των φορτίων που προέρχονται από αλληπάλληλους ιονισμούς, βλ. εν. 2.2) από τα strips. Έχει διαστάσεις $215 \times 60 \text{ mm}^2$. Πρόκειται ουσιαστικά για τον

⁶Για λόγους συντομίας, θα γίνει μόνο μία σύντομη αναφορά στον τρόπο λειτουργίας της ADDC

ενδιάμεσο μεταξύ του ανιχνευτή και των δύο καρτών που διαχειρίζονται τα δεδομένα για το trigger (ADDC) και τα δεδομένα για την ενέργεια και το χρόνο εμφάνισης των παλμών που θα χρησιμεύσουν αργότερα στην ανακατασκευή των τροχιών και των ορμών (L1DDC) [12]. Το νούμερο "8" στο όνομα υπονοεί ότι κάθε MMFE8 έχει πάνω της οκτώ VMM ASIC, το κάθε ένα εκ των οποίων συνδέεται με 64 κανάλια του MicroMegas (άρα 64 strips), και προβαίνει σε λειτουργίες ανάλυσης σημάτων, όπως ενίσχυση, διαμόρφωση σχήματος, εύρεση κορυφών, και ψηφιοποίηση. Εκτός από τα οκτώ VMM ASIC, η τελική κάρτα MMFE8, θα διαθέτει και άλλα δύο ASIC: Το SCA (*Slow Control ASIC*), και το ROC *Read-Out Companion*. Το SCA δέχεται τα σήματα του configuration της MMFE και στέλνει σήματα για την παρακολούθηση των συνθηκών που επικρατούν στον ανιχνευτή (monitoring). Η διασύνδεση αυτή γίνεται με την κάρτα L1DDC. Το ROC, συλλέγει τα δεδομένα από τα VMM, τα συμπύσσει, και τα αποστέλλει σειριακά στην L1DDC. Ο ρυθμός μετάδοσης/λήψης δεδομένων είναι σταθερός στα 80 Mbps για τα e-link του SCA (configuration/monitoring data). Από την άλλη, αν η MMFE βρίσκεται στο εσωτερικό τμήμα του MicroMegas wedge (βλ. σχ. 1.4.Π), ο ρυθμός μετάδοσης δεδομένων του ROC είναι στα 320 Mbps ενώ αν βρίσκεται στο εξωτερικό, είναι στα 160 Mbps. Ο λόγος για τον οποίο συμβαίνει αυτό είναι επειδή η υψηλότερη ροή σωματιδίων στο εσωτερικό τμήμα (μεγαλύτερες τιμές του $|\eta|$) απαιτεί και υψηλότερο εύρος ζώνης μετάδοσης δεδομένων για τις συγκρούσεις [10]. Αυτή τη στιγμή, τα πρωτότυπα της MMFE8 που έχουν κατασκευαστεί δεν διαθέτουν ROC/SCA ASIC, αλλά στη θέση τους υπάρχει ένα FPGA της εταιρείας Xilinx® (Artix XC7A200T-2FBG484), για το οποίο αναπτύσσεται firmware που να προσομοιώνει τη λειτουργικότητα των δύο συνοδευτικών ASIC. Το FPGA θα χρησιμεύσει στο να δοκιμαστεί η απόδοση της πλακέτας σε πειραματικές συνθήκες με μικρότερη ροή σωματιδίων (όπως Test Beams και πειράματα με κοσμικές ακτίνες), άρα και αμελητέα επίπεδα ακτινοβόλησης. Οι τελικές κάρτες που θα χρησιμοποιηθούν στο NSW δεν θα διαθέτουν FPGA καθώς οι συνθήκες που θα επικρατούν στον ανιχνευτή ATLAS μετά την αναβάθμισή του μπορεί να καταστρέψουν τα FPGA (1700 Gy δόση και 0.4 T μαγνητικό πεδίο [11]). Η έρευνα-ανάπτυξη και σχεδίαση για τις κάρτες MMFE8 πραγματοποιείται στο πανεπιστήμιο της Αριζόνα των ΗΠΑ. Στη συνέχεια παρατίθεται φωτογραφία της πλακέτας και λεπτομερέστερη περιγραφή του τρόπου λειτουργίας του VMM ASIC.



Σχήμα 4.3.1: Φωτογραφία ενός πρωτότυπου MMFE8. Διακρίνονται τα οκτώ VMM ASIC στο πάνω μέρος της πλακέτας, και το FPGA (Xilinx® Artix-7) στο κάτω. [12]

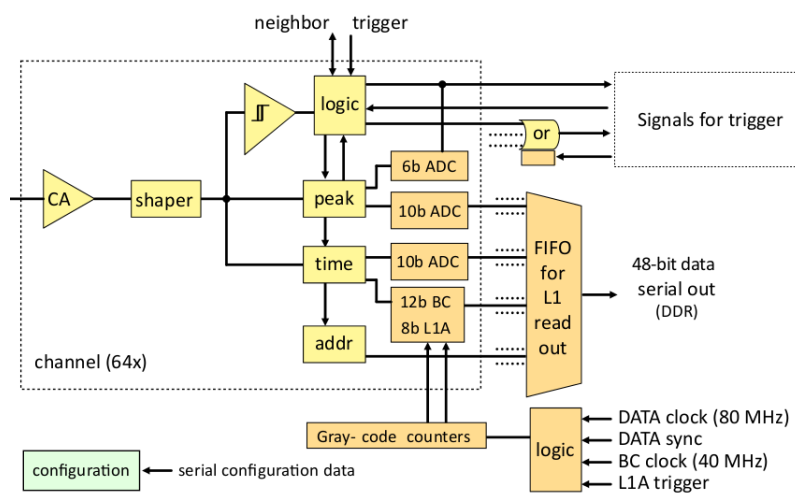
4.3.1 VMM ASIC

Το VMM, είναι ένα front-end readout ASIC με 64 κανάλια εισόδου, που αναπτύσσεται για την αναβάθμιση του μιονικού σπεκτρομέτρου του ATLAS από το Brookhaven National Laboratory (BNL) στο Upton, NY των ΗΠΑ. Κάθε πλακέτα MMFE8 έχει πάνω της οκτώ VMM ASIC, και κατά συνέπεια συλλέγει δεδομένα από 512 strips του MicroMegas. Το VMM, έχει διαστάσεις $13.5 \times 8.4 \text{ mm}^2$ και περιέχει περίπου πέντε εκατομμύρια τρανζίστορ σχεδιασμένα με τεχνολογία CMOS στα 130 nm [9]. Κάθε κανάλι που είναι συνδεδεμένο με ένα read-out strip, υλοποιεί έναν ενισχυτή φορτίου (charge amplifier), έναν ενισχυτή διαμόρφωσης (shaping amplifier) ο οποίος δρα ως φίλτρο, έναν διευκρινιστή (discriminator) με κατώτερο κατώφλι και εντοπισμό κορυφής, και έναν TAC. Τα αναλογικά σήματα που παράγονται από αυτά τα ηλεκτρονικά υποσυστήματα, ψηφιοποιούνται από τρεις διαφορετικούς ADC, με εξόδους 6,8 και 10 bit. Τα (ψηφιακά πλέον) δεδομένα που προκύπτουν από την ανάλυση των παλμών, αποθηκεύονται σε μία FIFO⁷ η οποία αποστέλλει τα δεδομένα στο ROC ASIC της MMFE8 [13, 14]. Υπενθυμίζεται ότι προς το παρόν, το ρολό του ROC (και του SCA), το παίζει ένα FPGA. Ο αναγνώστης παραπέμπεται στην ενότητα 3.3 προκειμένου να μελετήσει τον τρόπο λειτουργίας των προαναφερθέντων υποσυστημάτων, αλλά και για να διαπιστώσει την ομοιότητα που παρουσιάζουν τα ηλεκτρονικά υποσυστήματα που χρησιμοποιούνται εδώ και δεκαετίες στην ανάλυση σημάτων ανιχνευτών, με αυτά που έχουν υλοποιηθεί σε αυτό το ASIC των

⁷FIFO: First In First Out. Πρόκειται ουσιαστικά για μία μνήμη (data buffer), που αποθηκεύει ψηφιακά αλφαριθμητικά, τα οποία και επεξεργάζεται δίνοντας προτεραιότητα στην παλαιότερη εγγραφή. Η λειτουργικότητά της μελετάται περισσότερο στο Κεφάλαιο 6.

λίγων χιλιοστών.

Σε γενικές γραμμές, το VMM προσφέρει ακριβείς μετρήσεις για το φορτίο που εναποτίθεται στα strips, αλλά και για το πότε συνέβη αυτή η συλλογή φορτίου, σε σχέση με το BC clock⁸. Επίσης, παρέχει και άμεσα δεδομένα που χρησιμοποιούνται για το triggering, ενώ η διαδικασία του readout από την L1DDC είναι σχετικά σπάνια, και εξαρτάται από το αν ένα γεγονός θεωρηθεί έγκυρο από τον trigger processor. Η διαδικασία αυτή θα ξεκαθαριστεί στη συνέχεια, μετά από μία σχηματική αναπαράσταση των ηλεκτρονικών υποσυστημάτων του VMM (Σχήμα 4.3.II):

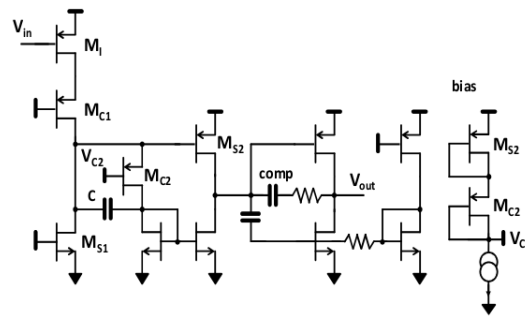


Σχήμα 4.3.II: Σχηματική αναπαράσταση της αρχιτεκτονικής του VMM. [9]

Όταν ένας παλμός εισέλθει στο VMM, ο προενισχυτής μετατρέπει το συνολικό φορτίο σε έναν αναλογικό παλμό τάσης ο οποίος μορφοποιείται από έναν ημι-Γκαουσιανό shaper, που παράγει το τελικό σήμα τάσης το μέγιστο πλάτος του οποίου είναι ανάλογο του φορτίου που εναποτέθηκε στο κανάλι. Το σχεδιάγραμμα σε επίπεδο CMOS του προενισχυτή παρατίθεται στο Σχήμα 4.3.III.

Το επεξεργασμένο σήμα από τον shaper εισέρχεται στη συνέχεια στον discriminator, ο οποίος έχει ρυθμιζόμενο κατώτατο κατώφλι. Μόλις κάποιος παλμός ξεπεράσει

⁸Το Bunch Crossing (BC) clock είναι ένα ρολόι με περίοδο 25 ns το rising edge του οποίου ταυτίζεται με τη στιγμή που τα πρωτόνια στη δέσμη διασταυρώνονται μεταξύ τους. Ταυτίζεται επομένως με τη στιγμή που δημιουργούνται τα γεγονότα στο κέντρο του ATLAS.



Σχήμα 4.3.III: Σχεδιάγραμμα του ενισχυτή του VMM σε επίπεδο τρανζίστορ. [13]

το κατώφλι αυτό, ενεργοποιείται μία μέθοδος εντοπισμού της κορυφής του παλμού (peak detection). Τη στιγμή που εντοπίζεται η κορυφή και καταγράφεται το ύψος της, ενεργοποιείται ο Time-to-Amplitude Converter (TAC), ο οποίος λαμβάνει το σήμα Start και ξεκινάει να εκφορτίζει έναν πυκνωτή. Το σήμα Stop έρχεται με το επόμενο tick του BC clock όπου και ο πυκνωτής παύει να εκφορτίζεται. Με αυτό τον τρόπο, υπολογίζεται το ύψος του παλμού (άρα και το φορτίο που συλλέχθηκε, άρα και η ενέργεια που εναπόθεσε το σωματίδιο), και ο χρόνος που εμφανίστηκε η κορυφή του, σε σχέση με τα rising edge δύο διαδοχικών tick του BC clock. Αξίζει επίσης να σημειωθεί πως όταν ένα σήμα ξεπερνάει το κατώφλι του discriminator, ενεργοποιείται και ένα κύκλωμα (neighbor logic) που ειδοποιεί τα γειτονικά κανάλια να αγνοήσουν τον δικό τους discriminator, και να καταγράψουν έτσι και αλλιώς τις ενέργειες των παλμών τους, όσο μικροί και αν είναι αυτοί. Αυτή η έξυπνη μεθοδολογία επιτρέπει στο να τεθεί ψηλά το επίπεδο του κατωφλίου, χωρίς ουσιαστικά να χάνονται πληροφορίες, αφού όταν καταγραφεί μεγάλος παλμός σε ένα κανάλι, τα διπλανά θα αποθηκεύσουν με τη σειρά τους την ενέργεια που συνέλεξαν, η οποία σίγουρα δεν θα πρέπει να αγνοηθεί, όσο μικρή και αν είναι αυτή [8, 9, 13, 14].

Τρία διαφορετικά κυκλώματα μετατροπής από αναλογικό σε ψηφιακό (Analog to Digital Converters (ADC)), δέχονται τα αποτελέσματα της ανάλυσης των σημάτων, και τα ψηφιοποιούν. Πιο συγκεκριμένα, για τη μέτρηση του χρόνου, ο 8-bit ADC, ψηφιοποιεί τον παλμό του πυκνωτή από τον TAC, και συμπύσσει το ένα byte που παράγει, με ένα αλφαριθμητικό μήκους 12-bit, το οποίο δημιουργείται από έναν Gray Code counter που αυξάνεται από το BC clock. Ουσιαστικά, η έξοδος του TAC, παράγει την πληροφορία της ακριβούς μέτρησης του χρόνου, και ο counter, ο οποίος φτάνει στη μέγιστη τιμή του και μηδενίζεται σε πιο αραιά χρονικά διαστήματα, πα-

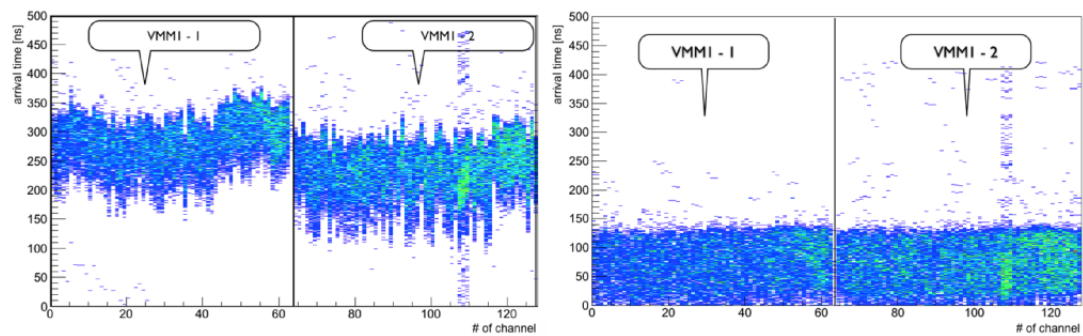
ρέχει μία σχετικά πιο χονδροειδή εκτίμηση του χρόνου [14]. Συνδυάζοντας όμως και τα δύο αποτελέσματα, κατασκευάζεται ένα αλφαριθμητικό μήκους 20 bit, που φέρει την πληροφορία του ακριβούς χρόνου (timestamp) που εμφανίστηκε ένας παλμός με διακριτική ικανότητα της τάξης του ns . Για τη μέτρηση της ενέργειας τώρα, ο 10-bit ADC λαμβάνει το σήμα που φέρει την πληροφορία του μέγιστου ύψους του παλμού από τον peak detector, και το ψηφιοποιεί με μεγάλη ακρίβεια σε περίπου 200 ns (σχετικά μεγάλο χρονικό διάστημα). Ο ADC των 6 bit από την άλλη, προβαίνει και αυτός σε μετατροπή του σήματος του peak detector, αλλά με λιγότερη ακρίβεια σε σχέση με τον ADC των 10 bit (πρόκειται για ADC τύπου flash που έχει αναφερθεί στο Κεφάλαιο 3). Η όλη διαδικασία ολοκληρώνεται σε 25 ns . Το αν θα χρησιμοποιηθεί ο πιο ακριβής ADC ή ο λιγότερο ακριβής για την κατασκευή του ψηφιακού πακέτου που θα εμπεριέχει την πληροφορία της συνολικής ενέργειας, εξαρτάται από τις προτιμήσεις του χρήστη. Σε κάθε περίπτωση πάντως, το τελικό πακέτο αποτελείται από 38 bit. Το πρώτο είναι το flag⁹ για το readout, το δεύτερο είναι το flag που υποδεικνύει ότι ξεπεράστηκε το κατώφλι του discriminator, μετά ακολουθούν τα 6 bit της διεύθυνσης του καναλιού, μετά τα 10 bit του ύψους του παλμού, και τέλος τα 20 bit του timestamp. Η τελική συμβολοσειρά αποθηκεύεται σε μία FIFO η οποία έχει βάθος τεσσάρων συμβολοσειρών. Αυτά τα δεδομένα, είναι τα read-out data που καταλήγουν τελικά στην L1DDC. Για την ακρίβεια, το ROC συλλέγει τα πακέτα από όλα τα VMM της MMFE στην οποία βαίνει, δηλαδή προβαίνει σε multiplexing¹⁰ συνολικά 512 διαφορετικών καναλιών σε ένα σειριακό output, που είναι το e-link της L1DDC.

Πριν μελετηθεί η διαδικασία με την οποία παράγονται και επεξεργάζονται τα ART data, αξίζει να σημειωθούν μερικές δευτερεύουσες λειτουργίες του VMM που ολοκληρώνουν την εικόνα της αρχιτεκτονικής του. Μια τέτοια επιπρόσθετη λειτουργία, είναι η ύπαρξη κυκλωμάτων για το *calibration* (βαθμονόμηση) του κάθε καναλιού. Πιο συγκεκριμένα, έχει βρεθεί πως το κάθε κανάλι του VMM παρουσιάζει διακυμάνσεις ως προς την ενίσχυση κατά τη μετατροπή των παλμών φορτίου σε παλμούς τάσης (fine gain). Προκειμένου να γίνουν σωστές μετρήσεις στις κυματομορφές των παλμών, οι σχετικές αυτές διακυμάνσεις πρέπει να είναι γνωστές. Για αυτό το λόγο, το κάθε κανάλι του VMM διαθέτει ένα εξωτερικό κύκλωμα που διοχετεύει με παλ-

⁹Ένα bit που λειτουργεί ουσιαστικά σαν boolean μεταβλητή.

¹⁰Η διαδικασία ομαδοποίησης πολλών σημάτων σε ένα.

μούς το κανάλι (pulser), προκειμένου να μετρηθεί η ακριβής απόκριση του καναλιού [8]. Εκτός από το gain, είναι επίσης σημαντικό να γίνει βαθμονόμηση και του επιπέδου του κατωφλίου του discriminator¹¹, γεγονός που επιτυγχάνεται επίσης με τη χρήση του pulser. Βαθμονόμηση επίσης επιδέχεται και ο TAC [9], ο οποίος παρουσιάζει μικροδιαφορές ως προς τη λειτουργικότητά του ανά κανάλι. Με τη χρήση του pulser, διοχετεύονται συγκεκριμένοι παλμοί στο σύστημα, οι οποίοι ορίζονται από το χρήστη. Στη συνέχεια τα αποτελέσματα του TAC επιστρέφονται πίσω στο κέντρο ελέγχου, όπου υπολογίζονται κάποιες χρονικές σταθερές για κάθε κανάλι. Αυτές οι σταθερές, θα ληφθούν υπόψιν αργότερα στην ανάλυση του χρόνου όπως αυτός δίνεται από τον κάθε TAC, ώστε οι μετρήσεις κατά τη διάρκεια του πειράματος να είναι ακριβέστερες. Το αποτέλεσμα του TAC calibration απεικονίζεται στο Σχήμα 4.3.IV.



Σχήμα 4.3.IV: Η κατανομή του υπολογισμένου χρόνου για δύο VMM ASIC, πριν (αριστερά) και μετά (δεξιά) το calibration. Είναι φανερό πως πριν τη βαθμονόμηση, παρουσιάζονται σημαντικές διακυμάνσεις στο χρόνο εμφάνισης των παλμών, όπως αυτός δίνεται από τον TAC. Μετά το calibration, η κατανομή είναι σαφώς πιο ισορροπημένη [9].

Όλες αυτές οι λειτουργίες του calibration, μαζί με το ακριβές configuration των επιμέρους λεπτομερειών του τρόπου λειτουργίας του VMM, υλοποιούνται ουσιαστικά μέσω της επικοινωνίας με την L1DDC. Όταν η L1DDC λάβει εντολή από το κέντρο ελέγχου του ATLAS ότι ένα συγκεκριμένο chip VMM πρέπει να αλλάξει κάτι στον τρόπο λειτουργίας του (π.χ. χρήση του 6-bit ADC αντί του 10-bit, αλλαγή του επιπέδου κατωφλίου του discriminator, κ.ά.), ή πρέπει να προβεί σε calibration, στέλνει

¹¹Ωστε να μην επηρεάζει τις μετρήσεις ο εγγενής θόρυβος που υπάρχει σε κάθε κανάλι.

την εντολή αυτή στο SCA ASIC της αντίστοιχης MMFE μέσω του e-link, και το SCA διοχετεύει τα ψηφιακά δεδομένα σε ένα DAC (Digital-to-Analog Converter) μέσα στο VMM. Ο αποκωδικοποιητής DAC, μετατρέπει το ψηφιακό σήμα της εντολής σε αναλογικό παλμό, ο οποίος αποστέλλεται στο αντίστοιχο ηλεκτρονικό υποσύστημα του ASIC που πρέπει να γίνει η διαμόρφωση της λειτουργίας. Με αυτό τον τρόπο, υπάρχει μία πλήρης επικοινωνία μεταξύ του κέντρου ελέγχου και του VMM.

ART Data (Trigger Data) και ADDC

Μαζί με την ανάλυση των σημάτων, το VMM παράγει και τα δεδομένα που χρησιμοποιούνται για το *trigger*. Πρόκειται για τα *Address in Real Time (ART) data*. Καθ' όλη τη διάρκεια της λειτουργίας του, το VMM εποπτεύει τα αποτελέσματα από τους discriminator όλων των καναλιών του. Όταν κάποιος παλμός ξεπεράσει το κατώφλι του discriminator, ή όταν βρεθεί η πρώτη κορυφή ενός παλμού, τότε το VMM καταγράφει αμέσως τη διεύθυνση του καναλιού που εντοπίστηκε το γεγονός, και τη στέλνει στην ADDC μαζί με ένα flag που υποδεικνύει ότι συνέβη το γεγονός. Τα σήματα ART, αποστέλλονται στην ADDC μέσω ενός e-link σε κάθε bunch crossing (κάθε 25 ns). Στην ADDC, βρίσκονται τέσσερα ASIC: δύο ART2GBTx και δύο GBTx ASIC¹². Τα δύο πρώτα ASIC, επεξεργάζονται τα ART data που τους στέλνονται από τις 8 MMFE μέσω των e-link, και επιλέγουν τα σήματα που θα στείλουν στα GBTx. Τα GBTx, που είναι ουσιαστικά ASIC που προβαίνουν σε multiplexing πολλών σημάτων σε μία έξοδο οπτικής ίνας μεγάλης ταχύτητας, στέλνουν τα δεδομένα ART στα back-end electronics όπου γίνεται το trigger processing. Ο trigger processor, που αποτελείται κυρίως από FPGA, συνδυάζει τα δεδομένα του trigger από τους sTGC και τους MM, μαζί με τα αντίστοιχα δεδομένα από το Big Wheel. Για να συγχρονιστούν τα trigger data και των τριών ανιχνευτικών υποσυστημάτων, τα ηλεκτρονικά του κάθε ανιχνευτή πρέπει να αποστείλουν τα δεδομένα τους μέσα σε αυστηρά καθορισμένα χρονικά πλαίσια. Για τον MM για παράδειγμα, από τη στιγμή που θα γίνει μία αλληλεπίδραση στο κέντρο του ATLAS, μέχρι να φτάσουν τα δεδομένα του trigger της αλληλεπίδρασης αυτής στο κομμάτι εκείνο του trigger processor που συνδυάζει τα δεδομένα και των υπολοίπων ανιχνευτών, πρέπει να παρέλθει ένα μέγιστο χρονικό διάστημα της τάξης των $t_{max} = 1025 \text{ ns}$. Αυτή η καθυστέρηση

¹²Το GBTx ASIC θα μελετηθεί λεπτομερώς στη συνέχεια, αφού αποτελεί το κύριο ASIC της κάρτας L1DDC.

(latency) εξαρτάται από πολλούς παράγοντες, όπως για παράδειγμα: το χρόνο απόκρισης στο shaper του VMM, την ταχύτητα με την οποία το GBTx στην ADDC συμπύσσει τα δεδομένα από τα οκτώ e-links και τα αποστέλλει σειριακά στην οπτική ίνα που οδηγεί στον trigger processor, ή και την ταχύτητα μετάδοσης της πληροφορίας στις οπτικές ίνες που μόλις αναφέρθηκαν. Έχει υπολογιστεί πως το σύστημα ηλεκτρονικών του MicroMegas, προσφέρει ελάχιστη καθυστέρηση $t_{min} = 876 \text{ ns}$, και μέγιστη $t'_{max} = 1018 \text{ ns}$, που είναι ένα χρονικό πλαίσιο πλήρως αποδεκτό βάσει του σχεδιασμού ολόκληρου του πειράματος [9]. Τελικά, με τη σύγκριση όλων των δεδομένων του trigger, ο επεξεργαστής αποφαινεται για το ποια γεγονότα είναι έγκυρα (βλ. Σχ. 4.2.Ι), και αποστέλλει μέσω της L1DDC το σήμα *Level-1 Accept (L1A)*, στις κάρτες MMFE. Καθ' όλη τη διάρκεια αυτής της διαδικασίας, το VMM προέβαινε σε buffering των δεδομένων στη μνήμη FIFO. Η FIFO¹³ σε κάθε VMM, στέλνει μόνο τα αλφαριθμητικά που αντιστοιχούν σε έγκυρα γεγονότα, σύμφωνα πάντα με το σήμα L1A. Με τη λήψη του L1A δηλαδή, τα δεδομένα για το ύψος του παλμού και το χρόνο εμφάνισής του, που αντιστοιχούν σε έγκυρες τροχιές σύμφωνα με το L1A, δρομολογούνται στο e-link της L1DDC. Μέσω αυτής της διαδικασίας "κοσκινίσματος", μειώνεται δραστικά ο όγκος των δεδομένων που αποστέλλονται από τα front-end electronics στα υψηλότερα επίπεδα του CERN για να αποθηκευθούν προκειμένου να αναλυθούν λεπτομερώς σε δεύτερη φάση (off-line data analysis).

4.4 Level-1 Data Driver Card

Η πλακέτα *Level-1 Data Driver Card (L1DDC)* είναι ο ενδιάμεσος κρίκος που συνδέει τις MMFE8 με το σύστημα των back-end electronics, και συγκεκριμένα με το δίκτυο *FELIX*¹⁴ [10, 11]. Ο σχεδιασμός της έχει γίνει εξ' ολοκλήρου στο Ε.Μ.Π. από την ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών, ενώ τα πρώτα πρωτότυπα (L1DDC Prototype1) έχουν κατασκευαστεί από το Σεπτέμβριο του 2014. Την περίοδο σύνταξης της παρούσας διπλωματικής, το δεύτερο πρωτότυπο βρισκόταν υπό σχεδιασμό. Επειδή ο προγραμματισμός του FPGA αφορά το Prototype1, θα αφιερωθεί

¹³Σημειώνεται ότι η FIFO του VMM2 έχει βάθος μόνο τεσσάρων γεγονότων, άρα δεν μπορεί να προβεί σε επαρκές buffering. Στο VMM3, η FIFO θα βαθύνει αρκετά ώστε να μπορεί να αποθηκεύσει για αρκετό χρόνο τα δεδομένα μέχρι να φτάσει το L1A.

¹⁴FELIX: Front End Link eXchange. Το δίκτυο FELIX με τη σειρά του επικοινωνεί με το κέντρο δικτύου του ATLAS.

στη συνέχεια ξεχωριστή ενότητα για την πλακέτα αυτή. Στη συνέχεια θα περιγραφεί η τελική μορφή της κάρτας L1DDC, η οποία έχει μερικές διαφορές σε σχέση με το πρώτο πρωτότυπο.

4.4.1 Γενική Επισκόπηση της L1DDC

Ξεκινώντας από τα βασικά, κάθε κάρτα L1DDC συνδέεται με οκτώ κάρτες MMFE8 και με μία κάρτα ADDC (βλ. Σχ. 4.2.II) μέσω e-link, και με το δίκτυο FELIX μέσω οπτικής ίνας. Το κύριο ηλεκτρονικό σύστημα της L1DDC που τελεί τη διαδικασία αυτής της διασύνδεσης, είναι το *GBTx (GigaBit Transceiver)*, ένα ASIC που είναι ουσιαστικά ένας transceiver¹⁵ μεγάλων ταχυτήτων και υψηλής αντοχής σε ακτινοβόληση. Μέσω αυτής της επικοινωνίας, επιτυγχάνεται η εξόρυξη των δεδομένων για την ενέργεια που εναποθέτουν τα σωματίδια στα strips, και για το χρόνο εμφάνισης των παλμών (μαζί φυσικά με τη διεύθυνση του καναλιού. βλ. υποεν. 4.3.1). Επίσης, μέσω της L1DDC αποστέλλονται στις MMFE8 τα δεδομένα *TTC (Timing and Trigger Control)*. Αυτά τα δεδομένα παρέχουν για παράδειγμα τον χρονισμό (BC clock) στα ASIC της MMFE8. Εκτός από τα TTC data, η L1DDC διανέμει και τις εντολές για μικροαλλαγές στον τρόπο λειτουργίας των VMM ASIC. Αυτά τα configuration data, αποστέλλονται και στην ADDC με την οποία συνδέεται η L1DDC, καθώς η ADDC δεν διαθέτει αμφίδρομη διασύνδεση μέσω οπτικής ίνας με τα back-end electronics, προς εξοικονόμηση bandwidth. Τέλος, η L1DDC λαμβάνει πληροφορίες για τις εξωτερικές συνθήκες που επικρατούν στις MMFE8, τις οποίες και στέλνει πίσω στο κέντρο ελέγχου του ATLAS.

Συνοψίζοντας λοιπόν, η L1DDC τελεί τις εξής λειτουργίες [10]:

- Λήψη των δεδομένων για το χρόνο, το φορτίο-ενέργεια και τη διεύθυνση του strip από τις κάρτες MMFE8, και αποστολή στο δίκτυο FELIX. Αυτά τα δεδομένα καλούνται και *Level-1 Data*.
- Αποστολή εντολών για το configuration των καρτών MMFE8 και ADDC.
- Λήψη των monitoring data από τις MMFE8.
- Διανομή των δεδομένων TTC στις MMFE8.

¹⁵Transceiver: Transmitter and Receiver.

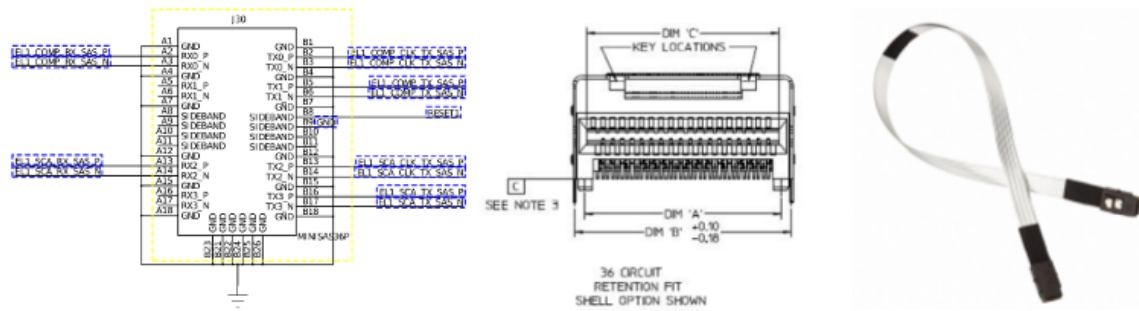
Αν και με μία πρώτη ματιά οι λειτουργίες αυτές φαίνονται απλές, στην πραγματικότητα δεν είναι ισχύει αυτό, λόγω της πολυπλοκότητας του συστήματος επικοινωνίας της L1DDC με τις υπόλοιπες πλακέτες. Κάθε L1DDC συνδέεται μέσω e-link με εννέα κάρτες (οκτώ MMFE8 και μία ADDC).

Το κάθε e-link αποτελείται από τρία διακριτά μεταξύ τους ζεύγη σειριακών διαφορικών γραμμών (*differential pairs*) [10]:

- Διαφορική γραμμή ρολογιού (*dClk+/dClk-*): Ρολόι που παρέχεται στις MMFE8 από το GBTx.
- Διαφορική γραμμή δεδομένων Downlink (*dOut+/dOut-*): Γραμμή δεδομένων από το GBTx προς τις MMFE ή προς την ADDC.
- Διαφορική γραμμή δεδομένων Uplink (*dIn+/dIn-*): Γραμμή δεδομένων από τις MMFE8 προς το GBTx.

Όπως έχει αναφερθεί και προηγουμένως στο παρόν Κεφάλαιο (βλ. εν. 4.2), το κάθε e-link, ανάλογα το ρόλο του, λειτουργεί και με διαφορετικές ταχύτητες. Όλα τα downlink για παράδειγμα, τα οποία καταλήγουν στα SCA της κάθε MMFE, έχουν ταχύτητα 80 *Mbps* (slow control), ενώ οι ταχύτητες των uplink, κυμαίνονται από 160 *Mbps* μέχρι 320 *Mbps*, ανάλογα με τη θέση της MMFE στον ανιχνευτή. Υψηλότερη ροή σωματιδίων συνεπάγεται και υψηλότερες ταχύτητες διασύνδεσης.

Το φυσικό μέσο που χρησιμοποιείται στο σύστημα ηλεκτρονικών του NSW για τη διασύνδεση των e-link, είναι τα καλώδια *miniSAS*. Το κάθε ένα από αυτά τα καλώδια μπορεί να φιλοξενήσει μέχρι και τέσσερα διαφορετικά ζεύγη διαφορικών γραμμών για την λήψη σημάτων, και αντίστοιχα τέσσερα ζεύγη για την αποστολή σημάτων. Αν αναλογιστεί κανείς ότι η κάθε MMFE8 διαθέτει δύο ASIC (ROC και SCA), και ότι η L1DDC πρέπει να επικοινωνεί με το κάθε ASIC ξεχωριστά μέσω διαφορετικών e-link, το καλώδιο *miniSAS* αποτελεί μία καλή επιλογή αφού επιτρέπει στην L1DDC να συνδέεται με μία MMFE8 μέσω ενός καλωδίου. Ένα άλλο πλεονέκτημα του *miniSAS* έγκειται στο ότι ο κοννέκτορας (*connector*) του, είναι μικρός σε μέγεθος. Έτσι περιορίζεται το μέγεθος ολόκληρης της L1DDC, γεγονός πλήρως επιθυμητό, λόγω του περιορισμού του χώρου που πρέπει να καταλαμβάνουν τα ηλεκτρονικά πάνω στον ανιχνευτή.



Σχήμα 4.4.I: Αριστερά: Το schematic του miniSAS connector μαζί με τα σήματα που στέλνει και λαμβάνει. Ο συγκεκριμένος κοννέκτορας συνδέεται με καλώδιο miniSAS που καταλήγει στις MMFE8. Υπάρχουν δύο ζεύγη διαφορικών που λαμβάνουν σήματα, ένα για το ROC και ένα για το SCA, και τέσσερα ζεύγη που στέλνουν σήματα, δύο που στέλνουν σήματα χρονοισμού στα ROC/SCA, και δύο που στέλνουν δεδομένα στα ASIC που μόλις αναφέρθηκαν. Κέντρο: Διαστάσεις του Molex 36p miniSAS connector. Δεξιά: Ένα καλώδιο miniSAS. [10].

Κάθε L1DDC λοιπόν, υλοποιεί 48 διαφορικές γραμμές για τις MMFE (δύο e-link για κάθε κάρτα, ένα για το ROC ένα για το SCA), και 6 γραμμές για την ADDC (δύο e-link, ένα για κάθε GBTx που βρίσκεται πάνω στην ADDC). Οι υψηλές ταχύτητες δεδομένων που πρέπει να υφίστανται σε κάθε e-link, προκειμένου το σύστημα να ανταπεξέρχεται στην υψηλή συχνότητα των γεγονότων (άρα και δεδομένων) που θα επικρατεί στο NSW μετά την αναβάθμιση της φωτεινότητάς του, δυσχεραίνουν το έργο της L1DDC, η οποία καλείται να διαχειριστεί έναν πολύ μεγάλο όγκο δεδομένων, με υψηλή ροή, γρήγορα και αποδοτικά. Αυτή τη λειτουργία, την τελειοποιεί η GBTx.

Το GBTx, αναλαμβάνει την ταξινόμηση και προσωρινή αποθήκευση όλων των πληροφοριών στους registers του, προκειμένου να τα προωθήσει στα αντίστοιχα υποσυστήματα του συστήματος ηλεκτρονικών του NSW. Πιο συγκεκριμένα, προβαίνει σε multiplexing των δεδομένων του uplink, τα οποία πακετάρει και αποστέλλει σειριακά στο FELIX μέσω οπτικής ίνας, η οποία δρομολογεί τα δεδομένα με ταχύτητες που αγγίζουν τα 4.8 Gb/s. Επίσης, το GBTx λαμβάνει τα σειριακά δεδομένα από την οπτική ίνα του FELIX, και προβαίνει σε demultiplexing, δηλαδή τα ταξινομεί ανάλογα τον προορισμό τους (ποιο από τα δύο ASIC ποιας MMFE8, ή ποιο από τα δύο GBTx της ADDC), και τα δρομολογεί στο αντίστοιχο e-link. Όλη αυτή η διαδικασία του serialization/deserialization πρέπει να τελειώνει και υπό συνθήκες υψηλής ακτινο-

βόλησης που μπορεί να αλλοιώσουν τις πληροφορίες που διανέμει η L1DDC. Στο επόμενο Κεφάλαιο όπου μελετάται η δομή του πρώτου πρωτότυπου της L1DDC, θα αφιερωθεί ξεχωριστή ενότητα για το πως ακριβώς λειτουργεί το GBTx ASIC.

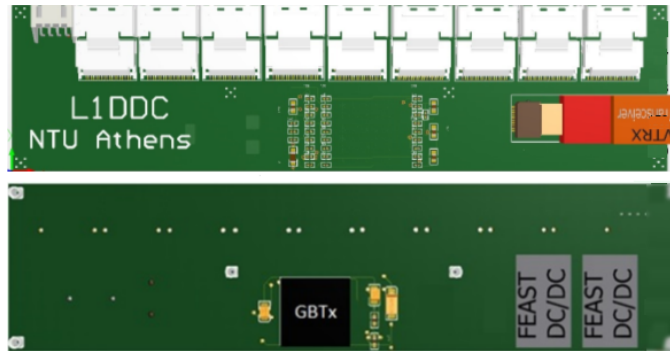
Αξίζει επίσης να αναφερθεί και ο σημαντικός ρόλος που παίζουν τρία δευτερεύοντα ASIC τα οποία βρίσκονται πάνω στην L1DDC. Κατ' αρχάς, το GBTx δεν επικοινωνεί άμεσα με την οπτική ίνα, αλλά ανάμεσά τους παρεμβάλλεται ο *VTRx Optical Transceiver*, ο οποίος αποτελείται από δύο επιμέρους ASIC: Το *GBTIA (GigaBit TransImpedance Amplifier)* και το *GBLD (GigaBit Laser Driver)*. Τα δύο αυτά ASIC, διοχετεύουν τα δεδομένα από το GBTx στην οπτική ίνα που οδηγεί στο δίκτυο FELIX. Επίσης, ο VTRx, λαμβάνει τα σειριακά δεδομένα από την οπτική ίνα, και τα διοχετεύει στο GBTx. Εκτός από αυτά τα ASIC, στην τελική L1DDC θα συμπεριληφθεί και ο *FEAST DC-DC Converter*, που είναι ένας μετατροπέας τάσης ο οποίος λαμβάνει τη γενική τάση εισόδου της L1DDC και παρέχει με συγκεκριμένες τάσεις τροφοδοσίας τα επιμέρους κυκλώματα της πλακέτας.

Τα κυριότερα ηλεκτρονικά υποσυστήματα (ASIC που όλα παρουσιάζουν υψηλές αντοχές στην ακτινοβολήση και στα ισχυρά μαγνητικά πεδία) που χρησιμοποιεί η L1DDC λοιπόν είναι τα εξής [10, 11]:

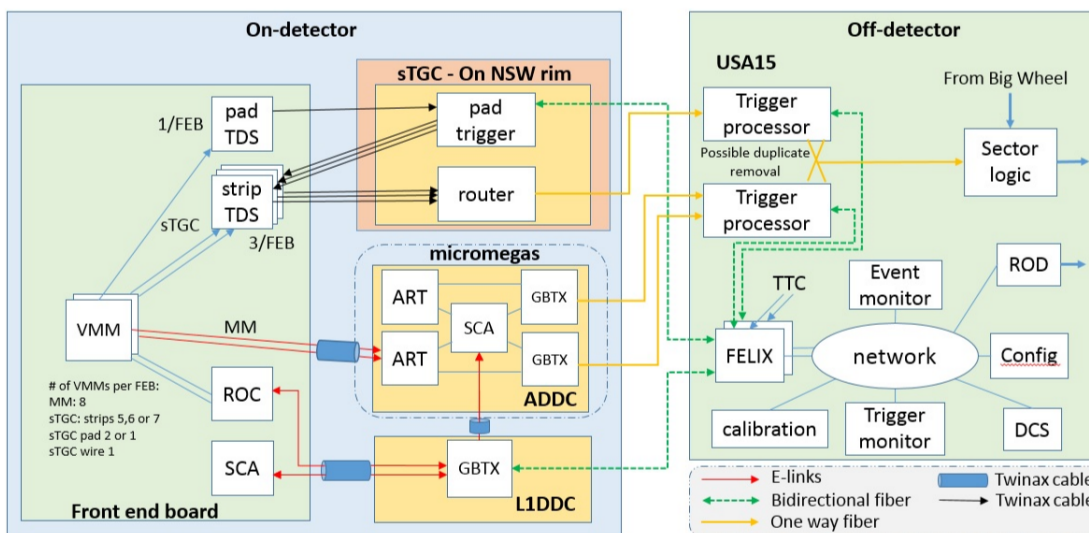
- *GigaBit Transceiver*: Το κεντρικό ASIC της L1DDC. Το GBTx προβαίνει σε multiplexing/demultiplexing των δεδομένων που λαμβάνει και στέλνει σειριακά στις κάρτες MMFE8 και ADDC μέσω των e-link, και στο δίκτυο FELIX μέσω οπτικής ίνας.
- *VTRx Optical Transceiver*: Αποτελείται από δύο ASIC (GBTIA και GBLD). Αποτελεί τον ενδιάμεσο ανάμεσα μεταξύ της γραμμής της οπτικής ίνας της πλακέτας και του GBTx.
- *FEAST DC-DC Converter*: Λαμβάνει τάσεις εισόδου στο εύρος $5 V - 12 V$, και παράγει δύο διακριτά επίπεδα τάσεων $2.5 V, 1.5 V$. Οι τάσεις αυτές, χρησιμοποιούνται από όλα τα υπόλοιπα ηλεκτρονικά συστήματα της L1DDC για την τροφοδοσία τους. Η αποδοτικότητά του αγγίζει το 76%.

Η κάρτα L1DDC, εκτός από τον ανιχνευτή MicroMegas, θα χρησιμοποιηθεί και ως ο ενδιάμεσος για τις front-end boards που θα βρίσκονται πάνω στους sTGC. Η L1DDC

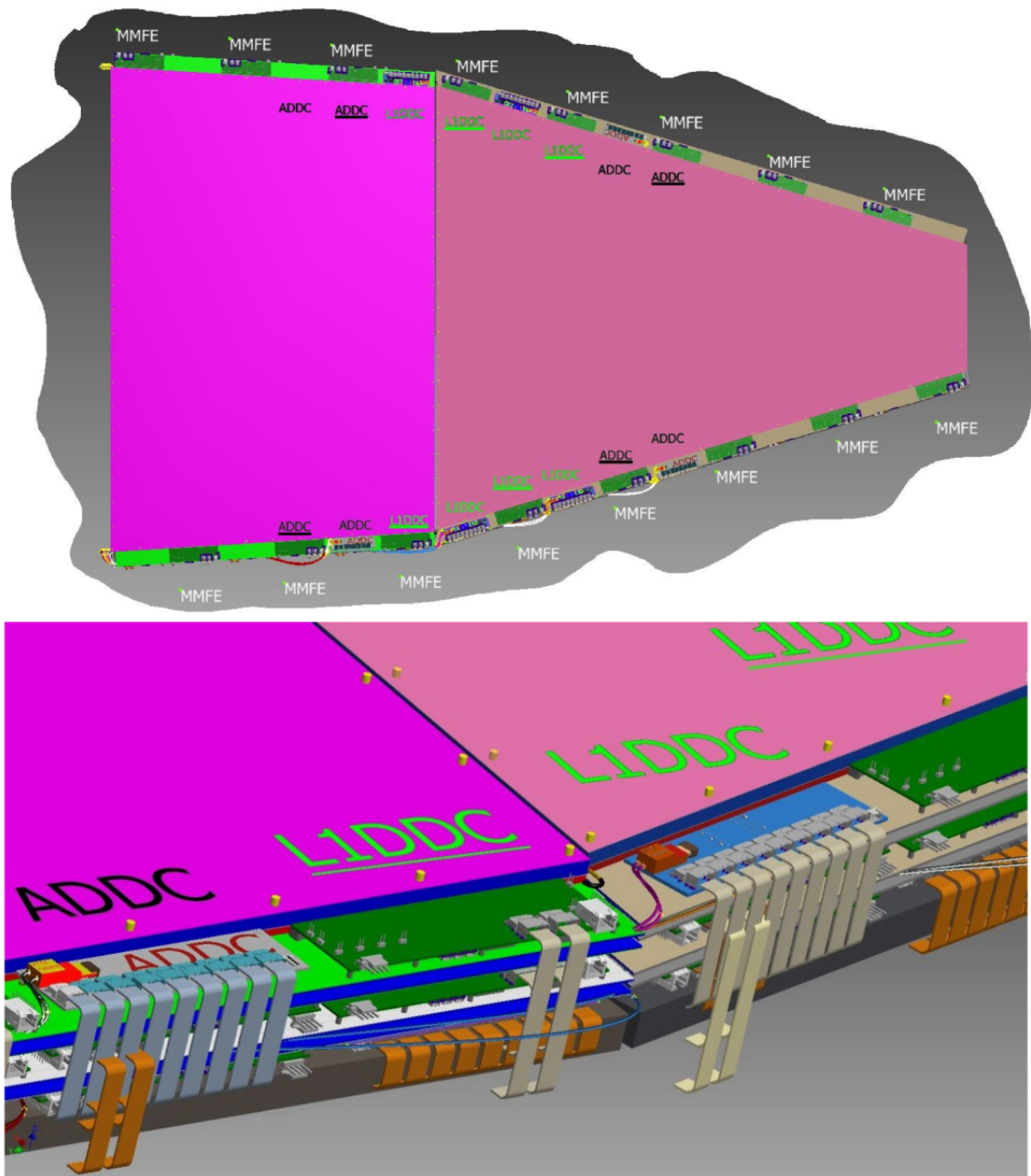
για τους sTGC όμως, θα διαθέτει μόνο τρεις κοννέκτορες για τις front-end boards, και οι διαστάσεις της θα είναι $90\text{ mm} \times 50\text{ mm}$. Από την άλλη, η L1DDC για τους MM, θα έχει διαστάσεις $200\text{ mm} \times 50\text{ mm}$ [11]. Παρακάτω, παρατίθενται εικόνες με τρισδιάστατα μοντέλα της L1DDC, και της τοποθέτησης των ηλεκτρονικών πάνω σε ένα MicroMegas wedge. Επίσης, δίνεται και ένα γενικό σχεδιάγραμμα των ηλεκτρονικών του NSW.



Σχήμα 4.4.II: Τρισδιάστατο μοντέλο της κάρτας L1DDC. Στο πάνω μέρος διακρίνονται οι εννέα κοννέκτορες των καλωδίων miniSAS (οκτώ για τις MMFE και ένας για την ADDC) και στα δεξιά ο VTRx. Στο κάτω μέρος κεντρικά βρίσκεται το GBTx ενώ στα δεξιά οι δύο FEAST DC-DC converters [11].



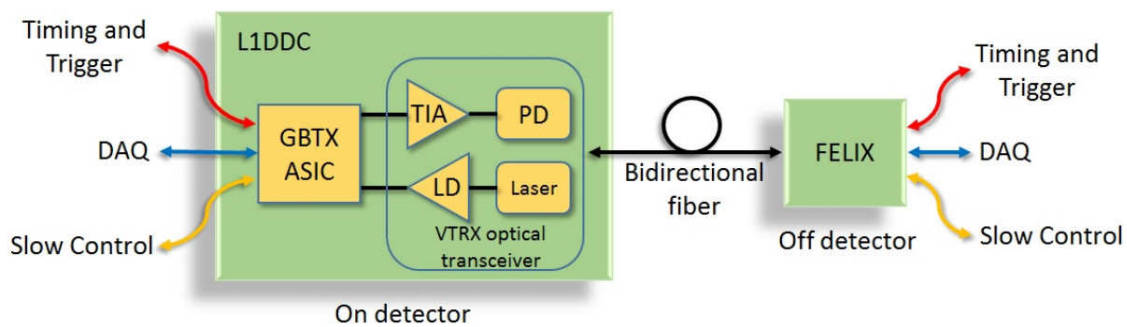
Σχήμα 4.4.III: Πρόχειρο σχεδιάγραμμα που απεικονίζει όλα τα ηλεκτρονικά υποσυστήματα του NSW, και τις μεταξύ τους διασυνδέσεις [11].



Σχήμα 4.4.IV: Τρισδιάστατο μοντέλο της τοποθέτησης των ηλεκτρονικών πάνω σε ένα MM wedge. Η πάνω φωτογραφία απεικονίζει μία γενική άποψη, όπου φαίνονται που θα τοποθετηθούν οι κάρτες MMFE, ADDC, και L1DDC. Στην κάτω φωτογραφία, αναπαρίστανται με περισσότερες λεπτομέρειες οι τοποθεσίες των καρτών μαζί με τις μεταξύ τους διασυνδέσεις [10].

Σύνοψη

Σε αυτό το Κεφάλαιο, μελετήθηκε η γενική δομή των ηλεκτρονικών υποσυστημάτων που θα υποστηρίξουν την αναβάθμιση του New Small Wheel, η οποία θα ολοκληρωθεί το 2018. Οι κάρτες MMFE8, με το εξελιγμένο ASIC ονόματι VMM, συλλέγουν τα σήματα από τον ανιχνευτή MicroMegas, και τα προωθούν στο δίκτυο των back-end electronics για περαιτέρω επεξεργασία. Ο κύριος συνδετικός κρίκος σε αυτή τη διαδικασία είναι η κάρτα L1DDC, μία κάρτα που έχει σχεδιαστεί εξ' ολοκλήρου από την ομάδα Πειραματική Φυσικής Υψηλών Ενεργειών του Ε.Μ.Π. Η κάρτα αυτή, θα χρησιμοποιήσει το GBTx ASIC, προκειμένου να διαχειριστεί τη μεγάλη ροή δεδομένων από και προς τις front-end boards. Το χρονοδιάγραμμα για τη σχεδίαση και παραγωγή των καρτών L1DDC, υποδεικνύει ότι η μαζική παραγωγή των τελικών καρτών θα ξεκινήσει το 2016 [10], και μέχρι τότε πρέπει να έχουν δοκιμαστεί όλες οι πιθανές λειτουργίες της πλακέτας, ώστε να εντοπισθούν τυχόν σφάλματα που μπορεί να εμφανιστούν. Για το λόγο αυτό, μία σειρά πρωτοτύπων έχει κατασκευαστεί, το πρώτο και πιο σημαντικό εκ των οποίων θα μελετηθεί εκτενώς στο επόμενο Κεφάλαιο.



Σχήμα 4.4.V: Σύνοψη της λειτουργίας της L1DDC [11].

5

Το Πρώτο Πρωτότυπο της Κάρτας L1DDC

Σε αυτό το Κεφάλαιο, θα μελετηθεί η δομή της πλακέτας L1DDC Prototype1, η οποία κατασκευάστηκε το Σεπτέμβριο του 2014. Συνολικά δημιουργήθηκαν πέντε κομμάτια. Υπάρχουν αρκετές διαφορές σε σχέση με την τελική κάρτα L1DDC που θα χρησιμοποιηθεί τελικά στο πείραμα του ATLAS, επειδή ο κύριος σκοπός του πρώτου πρωτότυπου είναι να ελεγχθούν οι γενικές λειτουργίες της πλακέτας, προκειμένου να διορθωθούν τυχόν προβλήματα στο μέλλον με μικρές αλλαγές στο σχεδιασμό. Εκτός από μία γενική επισκόπηση του πρώτου πρωτότυπου λοιπόν, στο παρών Κεφάλαιο θα δοθούν περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας του GigaBit Transceiver (GBTx) ASIC, αλλά και για το FPGA που θα το συνοδεύει. Πέρα από την L1DDC Prototype1 όμως, κατασκευάστηκαν και μερικές ακόμα βοηθητικές πλακέτες για τις οποίες επίσης θα γίνει σύντομη νύξη στη συνέχεια.

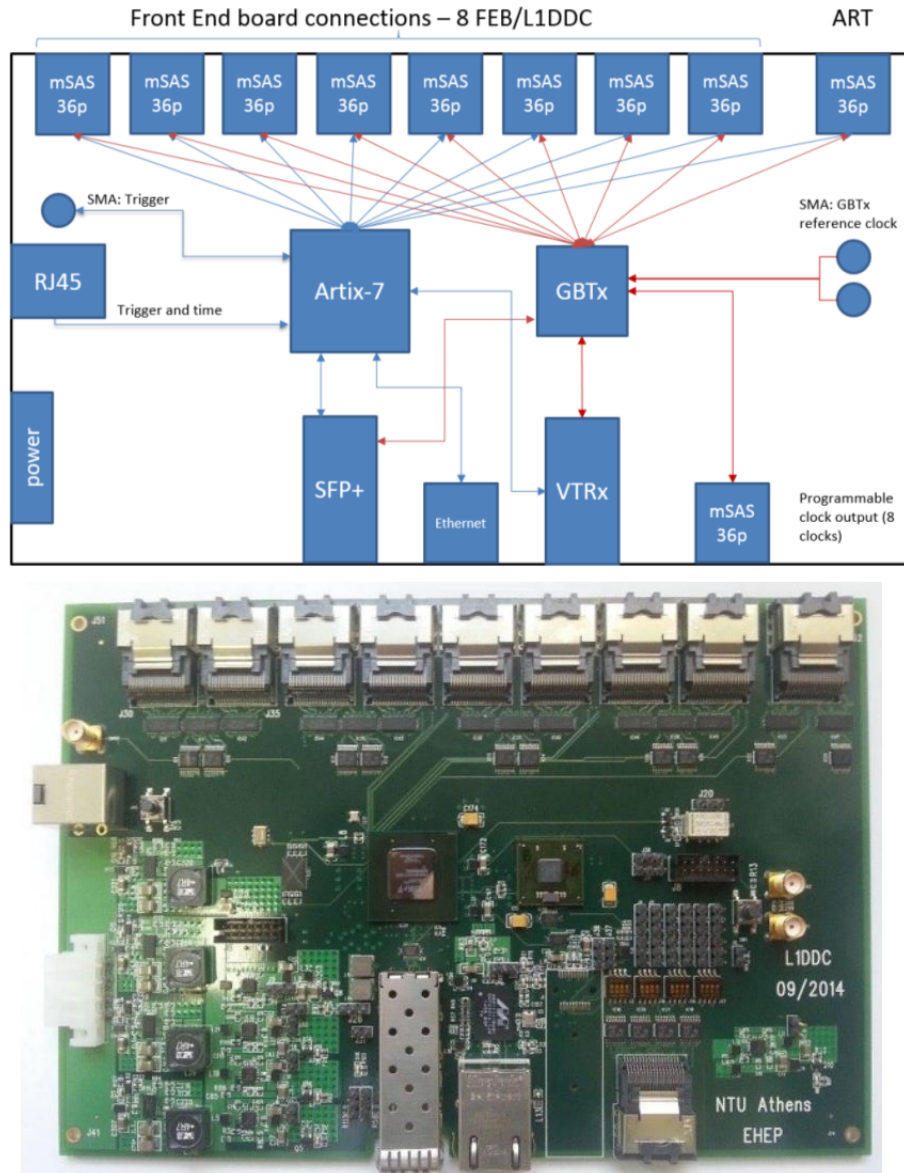
5.1 Γενικά για την L1DDC Prototype1 - Ο Ρόλος του FPGA

Το πρώτο πρωτότυπο της L1DDC (*L1DDC Prototype1*) είναι μία πλακέτα με πληθώρα λειτουργιών, οι ηλεκτρονικοί εγκέφαλοι της οποίας είναι το *GBTx* ASIC, και ένα *FPGA* της εταιρείας Xilinx® (*Artix-7 xc7a200t-3fbg484*). Τη στιγμή σύνταξης της παρούσας διπλωματικής εργασίας, βρισκόταν υπό ανάπτυξη ολόκληρο το firmware

του FPGA αυτού, το οποίο θα του επιτρέπει να μιμείται πλήρως όλες τις λειτουργίες του GBTx. Ένας λόγος ύπαρξης του FPGA στο πρώτο πρωτότυπο λοιπόν, είναι για να αντικαταστήσει το GBTx σε περίπτωση δυσλειτουργίας του τελευταίου [10]. Επίσης, λόγω της ευελιξίας του, το FPGA είναι ικανό να εκτελέσει ένα ευρύτερο σύνολο εργασιών, χωρίς κανένα ιδιαίτερο περιορισμό. Για το λόγο αυτό, ένα FPGA κρίνεται ιδανικό για διαδικασίες που είναι συνυφασμένες με πρωτότυπα καρτών, όπως το *debugging*. Επίσης, δεν πρέπει να ξεχνάει κανείς ότι τα ASIC συνήθως χρειάζονται πολύ μεγάλο χρονικό διάστημα (ακόμα και χρόνια) για να σχεδιαστούν σωστά αλλά και για να κατασκευαστούν. Στο μεσοδιάστημα αυτό, μόνο ένα FPGA μπορεί να υποκαταστήσει τη λειτουργία ενός ASIC πάνω στην πλακέτα. Στις κάρτες MMFE8 για παράδειγμα, κάτι τέτοιο συμβαίνει με τα ROC/SCA ASIC, τα οποία και υποκαθιστά ένα Artix-7 FPGA. Το γεγονός ότι χρησιμοποιούνται οι ίδιες γλώσσες προγραμματισμού (VHDL ή Verilog) για την ανάπτυξη των firmware στα FPGA, ανεξάρτητα τον κατασκευαστή ή το μοντέλο του FPGA, κάνει τη δουλειά των ερευνητικών ομάδων που συνεργάζονται μεταξύ τους για τη σχεδίαση του συστήματος ηλεκτρονικών του NSW, σαφώς ευκολότερη. Έτσι η ομάδα του E.M.Π. πλέον αναπτύσσει firmware όχι μόνο για την L1DDC, αλλά και για την MMFE8, προκειμένου να ελεγχθεί αν η μεταξύ τους επικοινωνία είναι ομαλή.

Ως προς τη συνδεσιμότητα με άλλες πλακέτες, ή με κάποιον υπολογιστή, το Prototype1 έχει όπως και η τελική πλακέτα οκτώ κοννέκτορες miniSAS για τα e-link με τις MMFE8, και έναν κοννέκτορα για τη σύνδεση με την ADDC. Επίσης, υπάρχει και ένας επιπλέον miniSAS connector, ο οποίος παρέχει οκτώ SLVS σήματα ρολογιών, που παράγονται από το GBTx. Επίσης, ένας SMA (SubMiniature version A) connector, μπορεί και παρέχει δεδομένα trigger και χρονισμού στο FPGA. Trigger επίσης μεταδίδει και ένας RJ45 (Registered Jack 45) connector στην άκρη της πλακέτας. Επίσης, υπάρχουν δύο JTAG (Joint Test Action Group) connectors, για το configuration του FPGA και του ASIC. Τέλος, η διασύνδεση με τα back-end electronics, προσομοιάζεται μέσω της σύνδεσης με έναν υπολογιστή μέσω ενός RJ45 connector για 10/100/1000 Mbps Ethernet. Εκτός από το συμβατικό Ethernet, υπάρχει και η επιλογή της οπτικής ίνας (η οποία επικοινωνεί με το FPGA) μέσω ενός SFP+ connector. Το FPGA συνδέεται και με τα δύο αυτά συστήματα επικοινωνίας. Τέλος, η πλακέτα διαθέτει και πολυάριθμα pins, στα οποία ο χρήστης μπορεί να στείλει σήματα από το FPGA προκειμένου να λάβει feedback για διάφορες εφαρμογές που επιθυμεί να εκτελεστούν

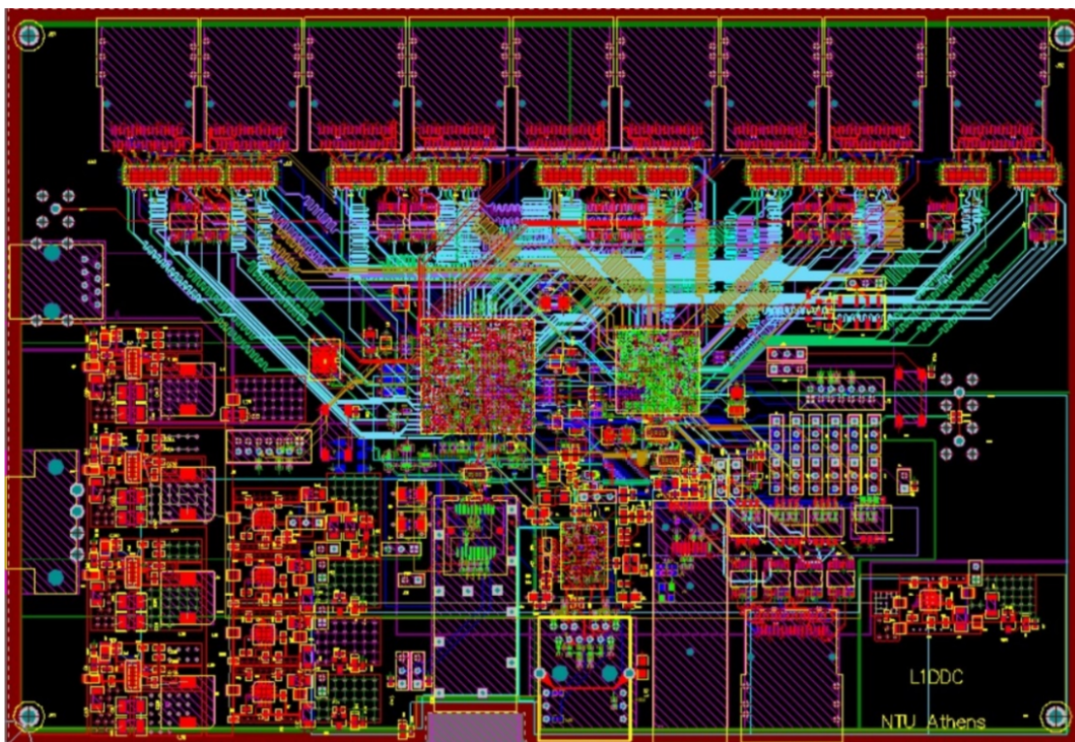
από το FPGA. Ακολουθεί σχηματική αναπαράσταση της πλακέτας με τα κυριότερα ηλεκτρονικά υποσυστήματά της, και μία φωτογραφία της.



Σχήμα 5.1.1: Σχηματική αναπαράσταση της L1DDC Prototype1, μαζί με μία φωτογραφία της πλακέτας [10].

Οι περισσότερες λειτουργίες που έχουν προαναφερθεί για την κάρτα L1DDC στο Κεφάλαιο 4, μπορούν να εκτελεστούν και από το πρώτο πρωτότυπο, με τη διαφορά

όμως, ότι μόνο ένα από τα πέντε κατασκευασμένα πρωτότυπα διαθέτει GBTx. Αυτό γιατί το GBTx βρίσκεται και αυτό σε στάδιο ανάπτυξης, και πολλές από τα υποσυστήματά του πρέπει να ελεγχθούν για την ορθή λειτουργία τους, πριν ξεκινήσει η μαζική παραγωγή του. Ένα εξωτερικό FPGA για παράδειγμα, μπορεί να διοχετεύσει σήματα στην L1DDC προκειμένου να ελεγχθεί η αποδοτικότητα του GBTx. Επίσης, το FPGA της L1DDC μπορεί να προγραμματιστεί με τέτοιο τρόπο ώστε να ελέγξει τους διαύλους επικοινωνίας μέσα στην L1DDC, οι οποίοι είναι πολυάριθμοι και περίπλοκοι, γεγονός που γίνεται φανερό στο Σχήμα 5.1.II όπου παρουσιάζεται η γενική συνδεσμολογία της πλακέτας.

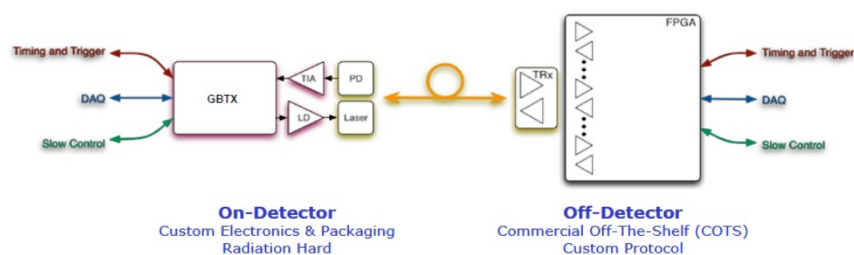


Σχήμα 5.1.II: Το layout της L1DDC. Υπενθυμίζεται ότι πρόκειται για μία κάρτα με δεκατέσσερα στρώματα [10].

Προκειμένου να αντιληφθεί ο αναγνώστης την πολυπλοκότητα και του firmware για το FPGA, το οποίο θα σχεδιαστεί έτσι ώστε να μιμείται τη λειτουργικότητα του GBTx, κρίνεται απαραίτητο σε αυτό το σημείο να παρατεθούν περισσότερες λεπτομέρειες σχετικά με το GBTx ASIC.

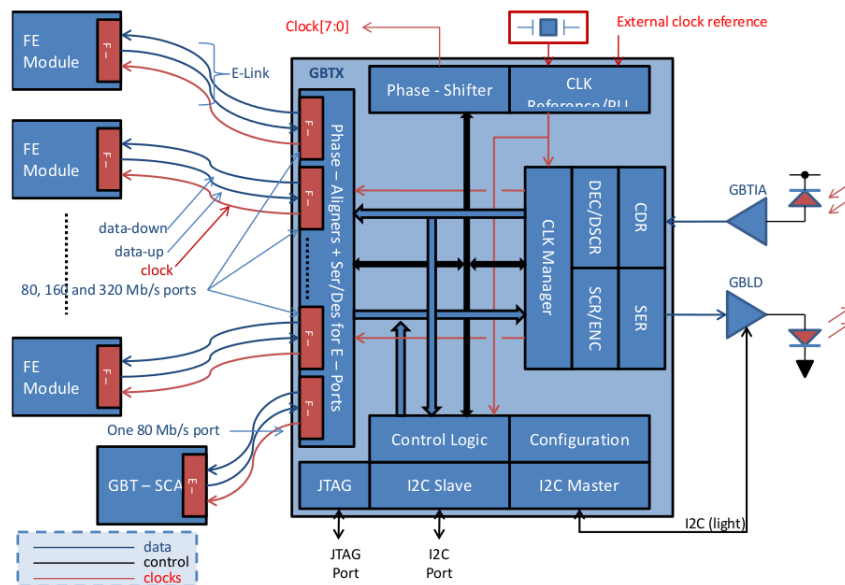
5.1.1 GBTx ASIC

Το *GigaBit Transceiver* (GBTx) ASIC είναι ένα ολοκληρωμένο κύκλωμα με υψηλές αντοχές στην ακτινοβολήση, το οποίο μπορεί να χρησιμοποιηθεί για την υλοποίηση ταχέων (3.2 – 4.8 Gb/s), αμφίδρομων διασυνδέσεων μέσω οπτικών ινών, για πειράματα Φυσικής Υψηλών Ενεργειών [17]. Έχει σχεδιαστεί με τεχνολογία CMOS στα 130 nm, διαθέτει 400 pins, και έχει διαστάσεις μερικών δεκάδων χιλιοστών. Το GBTx, μπορεί να διακρίνει τα τρία ξεχωριστά είδη δεδομένων που λαμβάνει σειριακά από το δίκτυο FELIX μέσω οπτικής ίνας, προκειμένου να τα διανεμίει στις αντίστοιχες front-end boards μέσω e-link. Αυτά τα τρία είδη data paths, είναι τα: *Data Acquisition (DAQ)*, *Trigger Timing and Control (TAC)* και *Slow Control (SC)*. Με άλλα λόγια, μέσω του GBTx, υπάρχει η δυνατότητα περισυλλογής δεδομένων από τις κάρτες front-end, είτε αυτά είναι δεδομένα για trigger, είτε ακριβείς πληροφορίες για χρόνο εμφάνισης και ύψος παλμών, ή δεδομένα για το monitoring των συνθηκών πάνω στον ανιχνευτή. Επίσης, το GBTx παρέχει σήματα χρονισμού στις front-end boards, και εντολές για αλλαγές στο configuration τους. Ένα GBTx, δύναται να υποστηρίξει μέχρι σαράντα front-end boards [11]. Το GBTx μπορεί και συμπύσσει όλα τα δεδομένα από τις κάρτες σε μία σειριακή οπτική έξοδο υψηλής ταχύτητας που οδηγεί στο δίκτυο FELIX, ενώ μπορεί και ταξινομεί τα δεδομένα που δέχεται από την οπτική ίνα του FELIX προκειμένου να τα διανεμίει στις κάρτες front-end. Η διεπαφή μεταξύ του GBTx και της οπτικής ίνας τελείται από δύο επιμέρους ASIC, τα GBLD και GBTIA. Ένα απλό σχεδιάγραμμα των λειτουργιών του GBTx παρατίθεται στο Σχήμα 5.1.III :



Σχήμα 5.1.III: Γενική λειτουργικότητα του GBTx [10].

Μια λεπτομερέστερη άποψη του τρόπου λειτουργίας του GBTx δίνεται στο Σχήμα 5.1.IV, όπου παρουσιάζεται η αρχιτεκτονική του ASIC:



Σχήμα 5.1.IV: Αρχιτεκτονική και διεπαφές του GBTx. [17].

Διασύνδεση με την Οπτική Ίνα - Λοιπά Υποσυστήματα

Όπως απεικονίζεται στο Σχήμα 5.1.IV, όσο αναφορά τη λήψη δεδομένων από την οπτική ίνα, το GBTx επικοινωνεί με το GBTIA ASIC [17]. Προκειμένου να ανακτήσει τα δεδομένα που μεταδίδονται με υψηλή ταχύτητα, το GBTx διαθέτει ένα κύκλωμα CDR (*Clock and Data Recovery*). Το CDR είναι η διαδικασία ανάκτησης ρολογιού από μία σειριακή γραμμή δεδομένων από τον δέκτη, η συχνότητα και η φάση του οποίου πρέπει να είναι τέτοια ώστε να μπορεί ο δέκτης να το χρησιμοποιήσει για να προβεί σε sampling του bitstream, προκειμένου να το αποθηκεύσει στη μνήμη του. Το κυριότερο ηλεκτρονικό υποσύστημα ενός κυκλώματος που προβαίνει σε CDR, είναι το PLL (*Phase Locked Loop*). Πιο συγκεκριμένα, το PLL, χρησιμοποιεί σαν ρολόι αναφοράς ένα ενσωματωμένο ρολόι (crystal oscillator) που βρίσκεται πάνω στο GBTx¹, και ανακτά από τη σειριακή ροή δεδομένων το CDR clock που έχει συχνότητα 2.4 GhZ. Με αυτό το ρολόι, κατασκευάζει από τα δεδομένα που δέχεται, δύο σειριακές γραμμές δεδομένων μήκους 60 bit έκαστη, τις οποίες μετά συμπύσσει στο τελικό 120-bit frame (περισσότερα για το frame παρατίθενται στη συνέχεια). Αυτή

¹Το ρολόι αυτό θα ενσωματωθεί σε μετέπειτα εκδόσεις του GBTx. Στην L1DDC Prototype1, το ρολόι αναφοράς παρέχεται απ' έξω από την πλακέτα, μέσω ενός SMA connector [10].

η διαδικασία ονομάζεται *deserialization*². Τέλος, το GBTx διορθώνει και τυχόν λάθη που εμφανίστηκαν κατά τη μετάδοση.

Προκειμένου να αποστείλει δεδομένα μέσω οπτικής ίνας, το GBTx επικοινωνεί με το GBLD ASIC [17]. Το GBTx συλλέγει τα δεδομένα που θέλει να προωθήσει στο δίκτυο FELIX, τα οποία είναι σε μορφή διανυσμάτων³, και προβαίνει σε *serialization*. Μαζί με τα σειριακά πλέον δεδομένα, το GBTx επισυνάπτει και ένα αυτοτελές πεδίο FEC (*Forward Error Correction*), το οποίο χρησιμεύει στον δέκτη στο να διορθώσει τυχόν λάθη που έγιναν σε μεμονωμένα bits κατά τη μετάδοση⁴. Τέλος, πριν αποσταλεί το πακέτο στο GBLD, το GBTx προβαίνει σε μικροαλλαγές, προκειμένου να υπάρχει DC balance (*Scrambling*). Το DC balance ή DC bias είναι η μέση τιμή του πλάτους μίας κυματομορφής τάσης. Γενικά στη μετάδοση δεδομένων είναι επιθυμητό η γραμμή να είναι ισορροπημένη ως προς την τάση, δηλαδή η μέση τιμή της τάσης να είναι μηδέν. Περισσότερα για το DC balance στη μετάδοση πληροφοριών μπορούν να βρεθούν παρακάτω στο παρών Κεφάλαιο όπου παρατίθενται μερικές λεπτομέρειες για τα πρωτόκολλα επικοινωνίας που εμφανίζονται σε διάφορα σημεία του NSW. Εκτός από την αποστολή δεδομένων που θα δρομολογηθούν στην οπτική ίνα, το GBTx προβαίνει και σε διαμόρφωση των λειτουργιών του GBLD μέσω ξεχωριστής γραμμής που υλοποιεί το πρωτόκολλο I²C, για το οποίο επίσης γίνεται νύξη μετέπειτα στο αντίστοιχο Παράρτημα.

Τέλος, αξίζει να αναφερθεί ότι το γενικότερο *monitoring/configuration* του GBTx γίνεται είτε μέσω πρωτοκόλλου I²C, του οποίου ο *slave* είναι το GBTx και ο *master* το FPGA της L1DDC, είτε μέσω της οπτικής ίνας. Τέλος, διατίθεται και διασύνδεση JTAG για εφαρμογές *Boundary Scan*. Αναφορικά, το *boundary scan* είναι μία τυποποιημένη διαδικασία όπου ελέγχονται όλες οι διασυνδέσεις μεταξύ των υποσυστημάτων ενός ολοκληρωμένου κυκλώματος για τυχόν σφάλματα.

²Η διαδικασία του SERDES (*serialization/deserialization*) θα μελετηθεί λεπτομερέστερα στο Κεφάλαιο 7 όπου παρουσιάζονται οι κώδικες που αναπτύχθηκαν για το firmware του FPGA της L1DDC.

³Διάνυσμα εν προκειμένω, καλείται μία συστοιχία-μητρώο από μεμονωμένα bits. Ένα διάνυσμα από οκτώ bits για παράδειγμα, είναι ένα byte.

⁴Συγκεκριμένα, χρησιμοποιείται το πρότυπο Reed-Solomon [17]

Διόρθωση Σφαλμάτων - Δομή του Frame

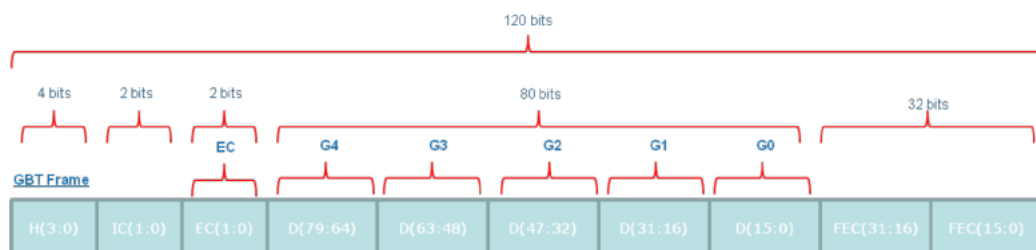
Μία σημαντική λειτουργία του GBTx είναι να προβαίνει σε διόρθωση σφαλμάτων που οφείλονται στο εχθρικό περιβάλλον που θα βρίσκεται η πλακέτα L1DDC, δηλαδή σε περιβάλλον όπου η ροή των ιοντιζουσών ακτινοβολιών (οι πιο επικίνδυνες από τις οποίες θα είναι νετρόνια) θα είναι μεγάλη. Η ακτινοβολία αυτή μπορεί να προκαλεί SEU (Single Event Upsets), που δύνανται να αλλάζουν την τιμή σε τυχαία bits στις πληροφορίες που είναι αποθηκεύονται στο GBTx ή μεταδίδονται μέσω του GBTx. Ένας καθιερωμένος τρόπος για τη διόρθωση τέτοιων σφαλμάτων είναι η μέθοδος του *TMR (Triple-Modular Redundancy)*. Εν προκειμένω, η τεχνική αυτή εφαρμόζεται ως εξής: Τα δεδομένα που φέρουν την πληροφορία για την τιμή μίας μεταβλητής, αποθηκεύεται σε τρία διαφορετικά σημεία της μνήμης (σε τρεις διαφορετικούς registers). Προκειμένου να διαβαστεί αυτή η πληροφορία, συγκρίνονται οι τιμές και των τριών registers, και η τελική τιμή καθορίζεται πλειοψηφικά. Έτσι αν έχει συμβεί ένα SEU σε κάποια περιοχή της μνήμης, οι άλλες δύο που θα έχουν μείνει ανέπαφες σε αυτό το κομμάτι (η πιθανότητα να αλλάξει στο ίδιο σημείο η τιμή ενός bit και στους άλλους registers είναι ελάχιστη), θα καθορίσουν την τελική τιμή της μεταβλητής. Η μέθοδος αυτή εφαρμόζεται από το GBTx σε στατικά σημεία της μνήμης (όπως για παράδειγμα στους registers που προγραμματίζει ο χρήστης για να καθοριστεί η συχνότητα των ρολογιών που παράγει το τσιπ), και όχι σε άλλα, όπως για παράδειγμα σε περιοχές της μνήμης όπου αποθηκεύονται προσωρινά τα δεδομένα που μεταδίδονται από τα e-link στην οπτική ίνα και αντίστροφα. Αυτό συμβαίνει επειδή η μέθοδος TMR είναι γενικά αργή, και η χρονική καθυστέρηση που θα υπεισερχόταν αν εφαρμοζόταν και στη διαδικασία του SERDES, θα έκρινε τη γενικότερη αποδοτικότητα του GBTx μη αποδεκτή [17]. Υπενθυμίζεται ότι η χρονική καθυστέρηση (latency) όλου του συστήματος ηλεκτρονικών του NSW είναι καθορισμένη σε πολύ αυστηρά πλαίσια (βλ. υποεν. 4.3.1).

Για το λόγο αυτό, στη διαδικασία όπου δρομολογούνται τα δεδομένα από και προς το δίκτυο FELIX, και από και προς τα e-link, το GBTx υλοποιεί τεχνικές διόρθωσης σφαλμάτων που συνοψίζονται γενικά στην προσάρτηση πεδίων FEC στα frames⁵ που στέλνονται στην οπτική ίνα [10, 17]. Επίσης, το δίκτυο FELIX με τη σειρά του, προσθέτει και αυτό με τη σειρά του παρόμοια κομμάτια στα πακέτα που στέλνει στο GBTx. Το GBTx αποκωδικοποιεί τα πεδία του FEC, και συγκρίνει τα αποτελέ-

⁵Άλλος τρόπος ονομασίας ενός πακέτου δεδομένων που μεταδίδεται από πομπό σε δέκτη.

σματα με τα περιεχόμενα του πακέτου που έλαβε. Στη συνέχεια προβαίνει σε διορθώσεις αναλόγως. Γενικά, η φιλοσοφία του FEC, είναι πως ο πομπός προσθέτει επιπλέον πληροφορίες μέσα στο αρχικό προς αποστολή πακέτο [19]. Μικρά πεδία FEC προστίθενται μετά από συγκεκριμένο n πλήθος bit, και εάν κατά τη μετάδοση της πληροφορίας γίνει κάποιο λάθος μέσα σε αυτό το εύρος bit, ο δέκτης αποκωδικοποιώντας τις πληροφορίες που φέρει το FEC για αυτό το εύρος bit, θα εντοπίσει το σφάλμα και θα το διορθώσει. Εάν όμως συμβούν πολλά SEU σε αυτά τα ψηφία, τότε ο δέκτης δεν θα μπορεί να διορθώσει τα σφάλματα. Για το λόγο αυτό προτιμάται τα κομμάτια που κωδικοποιεί το FEC να είναι μικρά. Όμως πολλά πεδία FEC μέσα σε ένα πακέτο, αυξάνουν κατά πολύ τις περιττές πληροφορίες που μεταδίδονται μέσα στη γραμμή, με αποτέλεσμα η "πραγματική" ροή της πληροφορίας⁶ να μειώνεται. Στο GBTx για παράδειγμα, όπου χρησιμοποιείται κωδικοποίηση FEC με τη μέθοδο Reed-Solomon, μειώνεται η ουσιαστική ταχύτητα μετάδοσης από τα 4.8 Gb/s στα 3.36 Gb/s [10]. Η διαφορά δηλαδή στο ρυθμό αποστολής και λήψης bit, έγκειται στα έξτρα bit που ανήκουν στα FEC, όπου μόλις ολοκληρωθεί η διαδικασία ελέγχου των πακέτων από το GBTx ή το FELIX, τα πεδία των FEC θα απορρίπτονται.

Το frame που θα χρησιμοποιηθεί από το GBTx για τη μετάδοση πληροφοριών από την L1DDC στο FELIX και αντίστροφα ονομάζεται *GBT Frame* [10, 17]. Στην Εικόνα 5.1.V παρατίθεται σχεδιάγραμμα της δομής του frame.



Σχήμα 5.1.V: Δομή του GBT frame [17].

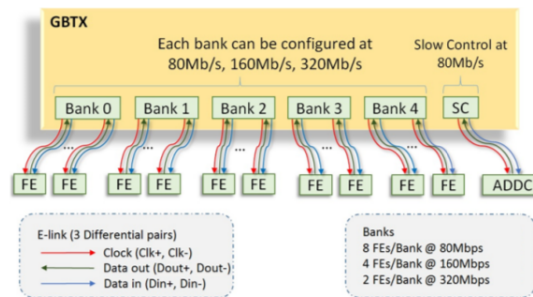
Το GBT frame αποτελείται από 120 bit. Μεταδίδεται ένα frame σε κάθε Bunch Crossing (κάθε 25 ns), που ισοδυναμεί σε ταχύτητα γραμμής 4.8 Gb/s. Ο Header (H) αποτελείται από τέσσερα bit, ενώ συνολικά το πεδίο του FEC έχει μήκος 32 bit. Έτσι μέ-

⁶Με τον όρο αυτό εννοείται η ροή των δεδομένων που φέρουν πληροφορίες που χρησιμεύουν πραγματικά στον δέκτη.

νουν 84 πραγματικά "χρήσιμα" bit τα οποία όταν μεταδίδονται μία φορά ανά tick του BC clock, δίνουν πραγματική ταχύτητα 3.36 Gb/s. Το 2-bit πεδίο IC (Internal Control) χρησιμοποιείται για διαμόρφωση της λειτουργίας του GBTx. Τα 82-bit πεδία EC (External Control) και D (Data), φέρουν την πληροφορία που προορίζονται για τα e-link (ή προέρχονται από τα e-link): Τα δεδομένα SC, DAQ και TTC (βλ. Σχ. 5.1.III). Το πεδίο FEC τώρα, χρησιμοποιεί τη μέθοδο Reed-Solomon για να προστατέψει την πληροφορία που φέρει ολόκληρο το frame. Το μέγεθος του FEC έχει οριστεί με τέτοιο τρόπο ώστε ακόμα και αν αλλοιωθούν δεκαέξι συνεχόμενα bit κατά τη μετάδοση της πληροφορίας, ο δέκτης να μπορεί να προβεί σε διορθώσεις αποκωδικοποιώντας το αντίστοιχο τμήμα του FEC. Το GBTx, μπορεί και υπολογίζει το FEC ενός προς αποστολή πακέτου μέσα σε ένα κύκλο του εξωτερικού ρολογιού αναφοράς (25 ns), ενώ μπορεί και ελέγχει ένα ληφθέν πακέτο για σφάλματα στο ίδιο χρονικό διάστημα [17]. Ως εκ τούτου, το FEC δεν επιβαρύνει καθόλου την ταχύτητα της διαδικασίας του SERDES μέσα στο ASIC, ενώ προστατεύει πλήρως το πακέτο από σφάλματα λόγω ιονισμών από ταχέα νετρόνια. Αξίζει τέλος να σημειωθεί ότι ολόκληρο το frame είναι DC balanced, γεγονός που διευκολύνει τη λήψη του σήματος και τη διαδικασία του CDR.

Διασύνδεση E-Link

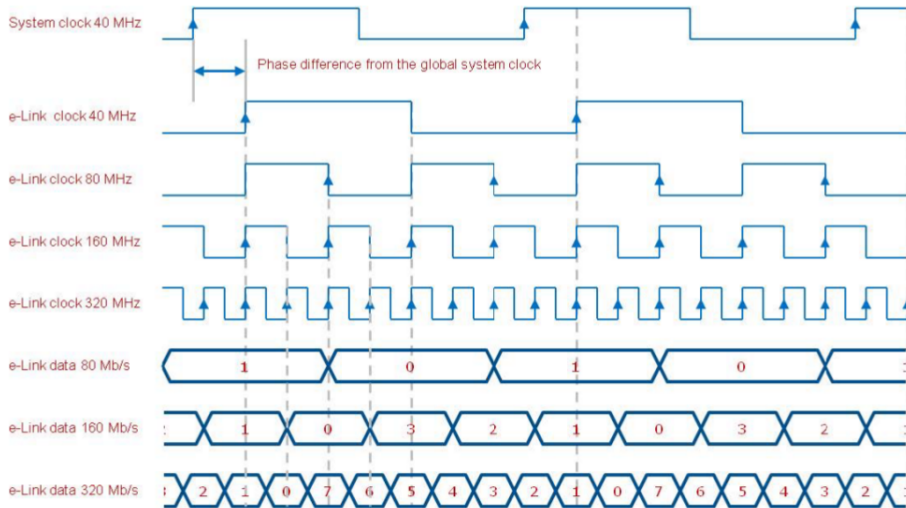
Οι διασυνδέσεις των e-link, επιτρέπουν στο GBTx να επικοινωνεί με τις κάρτες front-end, μέσω σειριακών bitstream που διαδίδονται διαφορετικά σε τρία διακριτά ζεύγη γραμμών (βλ. υποεν. 4.4.1). Το κάθε e-link, ανάλογα το ρόλο του, λειτουργεί και με διαφορετικές ταχύτητες. Όλα τα downlink για παράδειγμα, που ξεκινούν από το GBTx και καταλήγουν στα SCA της κάθε MMFE, έχουν ταχύτητα 80 Mbps (slow control), ενώ οι ταχύτητες των uplink, κυμαίνονται από 160 Mbps μέχρι 320 Mbps, ανάλογα με τη θέση της MMFE στον ανιχνευτή [10, 17]. Τα e-link υλοποιούνται στα ηλεκτρονικά υποσυστήματα του GBTx που ονομάζονται banks. Σε κάθε τσιπ, υπάρχουν οκτώ banks με προγραμματιζόμενες ταχύτητες e-link, τα οποία δύνανται να υποστηρίξουν μέχρι και σαράντα κάρτες front-end. Σε κάθε bank, μπορούν να συνδεθούν μέχρι και οκτώ front-end boards στα 80 Mbps, τέσσερις στα 160 Mbps ή δύο στα 320 Mbps. Ένα ξεχωριστό e-link, με προκαθορισμένη ταχύτητα 80 Mbps, χρησιμεύει στη διασύνδεση με την κάρτα ADDC (αποστολή δεδομένων για αλλαγή configuration). Η όλη διάταξη απεικονίζεται στο Σχήμα 5.1.VI.



Σχήμα 5.1.VI: Σχεδιάγραμμα της συνδεσμολογίας των e-link μεταξύ GBTx, FE boards και ADDC [11].

Κάθε e-link, φέρει ανάμεσα στα άλλα δύο ζεύγη, και το σήμα του ρολογιού, το οποίο μεταδίδεται από το GBTx στην κάρτα front-end. Η συχνότητα του ρολογιού αυτού μπορεί να είναι 40, 80, 160 ή 320 *Mhz* και παράγεται από ένα PLL που βρίσκεται μέσα στο GBTx με ρολόι αναφοράς το CDR clock. Η μετάδοση της πληροφορίας από και προς τις MMFE8, μπορεί να γίνεται απλά, όπου κατά το serialization, μεταδίδεται κάθε διαφορετικό bit ανά rising edge (βλ. Σχ. 3.1.I) του ρολογιού, ή με τη μέθοδο *DDR (Double Data Rate)*. Στο *DDR*, οι πληροφορίες μεταδίδονται ουσιαστικά με διπλάσιο ρυθμό, αφού η κατάσταση του serializer αλλάζει τόσο με το rising edge όσο και με το falling edge του ρολογιού. Έτσι, αν μεταδίδεται σε κάποιο e-link, ρολόι των 40 *Mhz* με *DDR*, η ροή των δεδομένων είναι στα 80 *Mbps*. Μεγάλη σημασία επίσης έχουν οι διαφορές φάσης μεταξύ των δεδομένων, των ρολογιών, και του global system clock, οι οποίες πρέπει να είναι γνωστές και σταθερές, προκειμένου να γίνεται σωστό sampling των bitstream από τους δέκτες. Στα e-link γενικά, αποφεύγεται η χρήση του CDR, και για αυτό το ρολόι που στέλνεται, πρέπει να έχει σταθερή και ίση διαφορά φάσης, με εκείνη του datastream που είναι στο ίδιο e-link. Για να επιτευχθεί αυτό, οι γραμμές που μεταδίδουν τα σήματα, και τα κυκλώματα των front-end ASIC που δέχονται αυτά τα σήματα, πρέπει να είναι σχεδιασμένα με τέτοιο τρόπο ώστε να εισάγουν ίσες χρονικές καθυστερήσεις και στα δύο είδη κυματομορφών. Στην (σπάνια) περίπτωση που δεν στέλνεται ρολόι στις front-end boards, ειδικοί τρόποι κωδικοποίησης των δεδομένων (7b/8b, 8b/10b encoding) επιτρέπουν σε κυκλώματα CDR πάνω στα front-end ASIC να εξάγουν ρολόι από το datastream για σωστό sampling. Από τη μεριά του GBTx τώρα, τα δεδομένα που δέχεται από τις MMFE8 περνούν πρώτα από ειδικά κυκλώματα ευθυγράμμισης φάσης (phase aligners), τα οποία ευθυγραμμίζουν με σταθερή διαφορά φάσης τα δεδομένα εισόδου, με το ρολόι που

μεταδίδει το GBTx στο αντίστοιχο e-link, προκειμένου να γίνεται άρτιο sampling από το GBTx [17]. Στη συνέχεια τα δεδομένα αυτά πακετάρονται, και αφού υπολογιστεί το FEC του frame, γίνεται δρομολόγηση στην οπτική ίνα που οδηγεί στο FELIX.

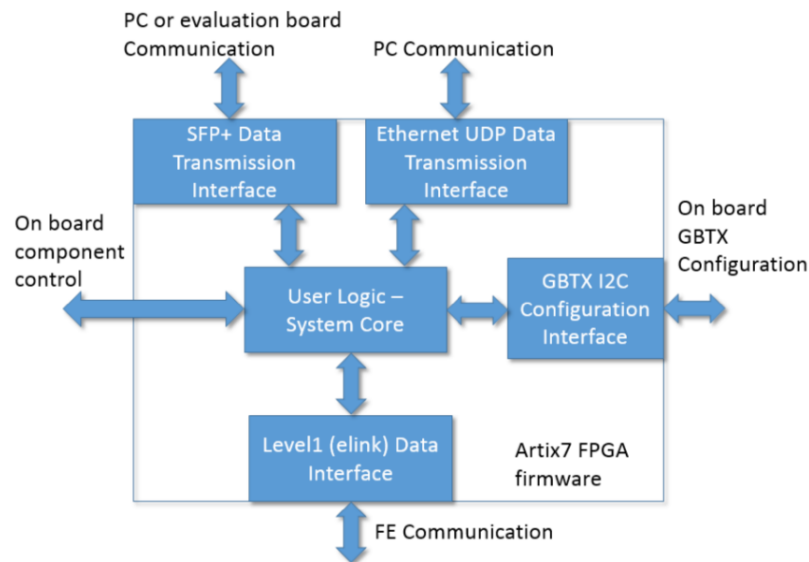


Σχήμα 5.1.VII: Τα ρολόγια των e-link, μαζί με τα δεδομένα που μεταδίδονται [10].

5.1.2 Artix-7 FPGA

Εκτός από το GBTx ASIC, η L1DDC Prototype1, διαθέτει και ένα FPGA της Xilinx® (xc7a200t-3fbg484). Ο κύριος λόγος ύπαρξης του FPGA είναι να αντικαταστήσει το GBTx σε περίπτωση δυσλειτουργίας του ASIC. Επίσης, αξίζει να σημειωθεί ότι τα πρώτα πρωτότυπα της L1DDC, δεν διέθεταν τον VTRx connector, γεγονός που δεν επέτρεπε στο GBTx να επικοινωνήσει με το back-end κομμάτι. Έτσι, ο μόνος τρόπος λήψης δεδομένων από τις MMFE8 σε πειράματα Test Beams, πριν κατασκευαστούν τα επόμενα πρωτότυπα της L1DDC, είναι μέσω του FPGA. Το FPGA, δύναται να προγραμματιστεί με τέτοιο τρόπο ώστε να μπορεί να υλοποιεί το firmware των e-link από τη μία μεριά, και ένα Ethernet interface από την άλλη, με το οποίο θα στέλνει τα δεδομένα (εν είδει διασύνδεσης με ένα εικονικό FELIX που θα προσομοιάζει ένας προσωπικός υπολογιστής), μέσω πρωτοκόλλου Ethernet UDP από τον κονήκτορα RJ45 που βρίσκεται πάνω στην L1DDC. Επίσης, την ίδια λειτουργία μπορεί να τελέσει και ο SFP+ connector με υψηλότερες ταχύτητες οπτικής ίνας. Τέλος, το FPGA μπορεί να υλοποιήσει και το πρωτόκολλο I²C στο οποίο δρα ως master. Το πρωτό-

κολλο αυτό μπορεί να χρησιμεύσει για την επικοινωνία με το GBTx προκειμένου να ελεγχθεί εάν οι registers του ASIC μπορούν να προγραμματιστούν εξωτερικά μέσω I²C με επιτυχία. Τέλος, το FPGA μπορεί να εκτελέσει μία πληθώρα άλλων εργασιών, που ανήκουν στη γενικότερη συνομοταξία διαδικασιών debugging για την κάρτα. Περισσότερες πληροφορίες για τα FPGA, τον τρόπο που προγραμματίζονται, και το firmware του FPGA της L1DDC, μπορούν να βρεθούν στα επόμενα Κεφάλαια, καθώς αποτελούν και το πρακτικό κομμάτι αυτής της διπλωματικής εργασίας. Ένα πρόχειρο σχεδιάγραμμα που απεικονίζει τις λειτουργίες που θα τελέσει το FPGA της L1DDC μετά την ολοκλήρωση σύνθεσης του firmware παρατίθεται στο Σχήμα 5.1.VIII:



Σχήμα 5.1.VIII: Διάγραμμα σε μπλοκ του firmware για το FPGA της L1DDC [10].

5.2 Λοιπές Πλακέτες

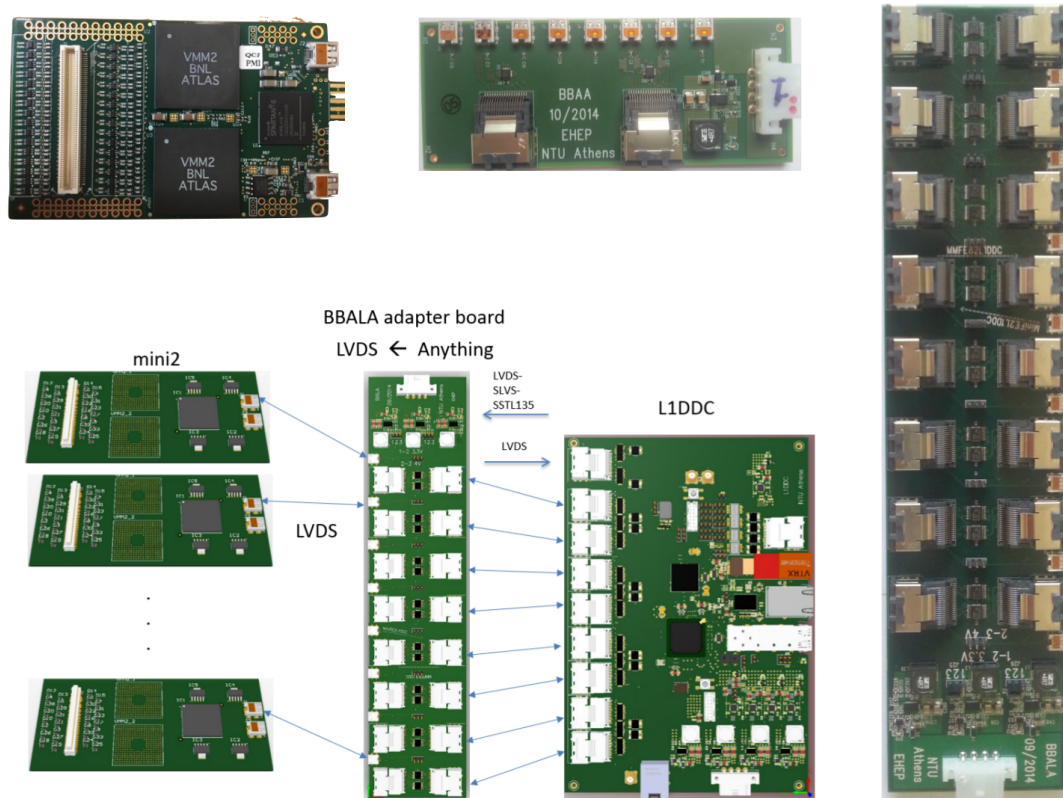
Μαζί με το πρώτο πρωτότυπο της πλακέτας L1DDC, σχεδιάστηκαν και κατασκευάστηκαν τρεις επιπλέον κάρτες. Η κάρτα *mini2* είναι μία front-end κάρτα, μικρή σε μέγεθος, η αρχιτεκτονική της οποίας δημιούργησε την ανάγκη για την κατασκευή δύο adapter boards (BBALA και BBAA), που αποτελούν τις γέφυρες ανάμεσα στις *mini2* και στις L1DDC και ADDC.

Η mini2, είναι μία μικρή πλακέτα front-end, που κατασκευάστηκε πριν την πρώτη MMFE. Φιλοξενεί δύο VMM2 ASIC για το read-out των σημάτων από έναν ανιχνευτή MicroMegas, και ένα Xilinx[®] Spartan-6 FPGA το οποίο συλλέγει τα σήματα από τα VMM και τα διαχειρίζεται, σαν συνδυασμός ROC/SCA ASIC. Η επικοινωνία με άλλες πλακέτες γίνεται μέσω δύο θυρών uHDMI. Το Spartan-6 προωθεί τα σήματα DAQ από τα VMM στον έναν uHDMI connector, που είναι δηλαδή το e-link. Η τάση τροφοδοσίας διοχετεύεται και αυτή από τη θύρα uHDMI, όπως επίσης και ένα I²C interface, όπου το Spartan-6 λειτουργεί ως Slave και δέχεται εντολές εξωτερικά από άλλο FPGA. Τα σήματα μεταδίδονται μέσω του πρότυπου LVDS.

Επειδή όμως οι L1DDC και ADDC διαθέτουν μόνο θύρες miniSAS, κατασκευάστηκε η BBALA (Bucharest Brookhaven Arizona L1DDC Adapter) Adapter Board, η οποία στη μία πλευρά της έχει θύρες miniSAS και στην άλλη θύρες miniSAS και uHDMI. Η κύρια λειτουργία της BBALA είναι να λειτουργεί ως η γέφυρα που ενώνει τις Adapter Boards με τις mini2. Πιο συγκεκριμένα [10], η BBALA λαμβάνει τάση τροφοδοσίας από ένα τροφοδοτικό, και μετατρέπει την τάση αυτή σε 3.3 V ή 4 V, η οποία διοχετεύεται στις θύρες uHDMI για την ενεργοποίηση των mini2. Τονίζεται επίσης ότι το GBTx λειτουργεί με επίπεδα τάσης στα e-link που αντιστοιχούν στο πρότυπο SLVS, και όχι στο LVDS (βλ. αντίστοιχο Παράρτημα). Για το λόγο αυτό, η BBALA διαθέτει translators, οι οποίοι δέχονται τα διαφορικά σήματα από τα miniSAS των L1DDC, και τα μετατρέπουν σε LVDS, μορφή την οποία μπορεί να διαχειριστεί η mini2. Τέλος, η BBALA είναι σχεδιασμένη με τέτοιο τρόπο που επιτρέπει την χρήση του FPGA της L1DDC για επικοινωνία με το Spartan-6 της mini2 μέσω I²C.

Η άλλη θύρα της mini2, προορίζεται για τη διασύνδεση με την ADDC. Επειδή και η ADDC έχει μόνο κοινέκτορες miniSAS και όχι uHDMI, κατασκευάστηκε η BBAA (Brookhaven Bucharest ADDC Adapter) Adapter Board [10]. Η κάρτα αυτή, εκτός από τη μετατροπή σημάτων από τα miniSAS στα uHDMI, διανείμει και τα σήματα χρονισμού από την ADDC στις mini2. Ο σχεδιασμός των ADDC, υποδεικνύει ότι πρέπει να συνδεθούν τέσσερις κάρτες mini2 (δηλαδή οκτώ VMM) σε έναν miniSAS connector της ADDC για να γίνει σωστή παροχή του σήματος του ρολογιού. Λόγω όμως του γεγονότος ότι η ADDC θα παρέχει από μία θύρα, ταυτόχρονα σε οκτώ ASIC το σήμα αυτό, προκαλείται πτώση τάσης στο σήμα του ρολογιού. Επομένως, η BBAA, προβαίνει σε *clock fan-out*, δηλαδή σε ενίσχυση της τάσης του ρολογιού, προκειμένου αυτό να φτάσει στις mini2 σε αποδεκτά επίπεδα. Οι κάρτες BBALA και

BBAA, σχεδιάστηκαν εξ' ολοκλήρου από την ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών του Ε.Μ.Π. Ακολουθούν φωτογραφίες των καρτών, και σχηματικές αναπαραστάσεις της συνδεσμολογίας τους.



Σχήμα 5.2.1: Πάνω αριστερά: mini2 front-end board [15]. Δίπλα της είναι η BBAA. Δεξιά: BBALA. Κάτω απεικονίζεται ένα πρόχειρο σχεδιάγραμμα της συνδεσμολογίας μεταξύ mini2, BBALA και L1DDC [10].

Σύνοψη

Σε αυτό το Κεφάλαιο, εξετάστηκε η αρχιτεκτονική της κάρτας L1DDC Prototype1, που είναι και το πρώτο πρωτότυπο της κάρτας L1DDC η οποία θα χρησιμοποιηθεί στην αναβάθμιση του NSW του ανιχνευτή ATLAS το 2018. Το πρωτότυπο αυτό, διαθέτει ένα FPGA μαζί με το GBTx ASIC, που έχει σαν σκοπό να παίξει βοηθητικό ρόλο στη λειτουργία του ASIC. Οι λειτουργίες του GBTx, οι οποίες ουσιαστικά συνοψίζονται στην προώθηση δεδομένων από και προς τις front-end boards, στο δίκτυο

FELIX των back-end electronics, θα υποκαθίστανται πλήρως από το FPGA. Τέλος, μελετήθηκαν επίσης οι βοηθητικές πλακέτες που έχουν αναπτυχθεί για τα πρωτότυπα του συστήματος ηλεκτρονικών του NSW.

6

Field-Programmable Gate Arrays

Στα προηγούμενα Κεφάλαια, έγιναν αλλεπάλληλες αναφορές για τα λεγόμενα *FPGA*. Τα *FPGA* (*Field-Programmable Gate Arrays*), είναι μία οικογένεια ολοκληρωμένων κυκλωμάτων, τα οποία μπορούν να υλοποιήσουν οποιαδήποτε λογική συνάρτηση, ανάλογα τα θέλω του προγραμματιστή που τα σχεδιάζουν (ή προγραμματίζουν). Σε μία οποιαδήποτε εφαρμογή, είτε αυτή περιλαμβάνει την ψηφιακή ανάλυση σήματος (*Digital Signal Processing* (*DSP*)), την υλοποίηση ενός μικροεπεξεργαστή, ή τη διαχείριση δεδομένων από πολλές πηγές (όπως γίνεται στην *L1DDC*), το μέσο για την επίτευξη των στόχων του σχεδιαστή του συστήματος, θα είναι σίγουρα κάποιο ολοκληρωμένο κύκλωμα. Τα *ASIC* είναι συνήθως η ιδανική επιλογή για τέτοια ζητήματα, όσο αναφορά τις γενικές επιδόσεις, το μικρό μέγεθος, και τη κατανάλωση ισχύος (η οποία γενικά για τα *ASIC* είναι πολύ χαμηλή). Όμως, η κατασκευή ενός *ASIC* απαιτεί την ενασχόληση μίας μεγάλης ομάδας εξειδικευμένων ερευνητών, η οποία πρέπει να αφιερώσει πολύ μεγάλο χρόνο στην σχεδίαση και ανάπτυξη του τσιπ, το οποίο μάλιστα, όταν κατασκευαστεί για πρώτη φορά, θα χρειάζεται σίγουρα εκτενείς βελτιώσεις. Το γεγονός αυτό καθιστά την παραγωγή ενός *ASIC* πολύ χρονοβόρα και ακριβή διαδικασία. Τέλος, επειδή ακριβώς τα *ASIC* είναι...*application-specific*, είναι και εξαιρετικά δυσπροσάρμοστα. Από τη στιγμή κατασκευής του, το *ASIC* είναι καταδικασμένο να εκτελεί μία και μοναδική λειτουργία. Το κενό που δημιουργείται από αυτή τη μη-προσαρμοστικότητα, έρχεται να το καλύψει το *FPGA*, το οποίο είναι ο ορισμός της ευελιξίας στον τομέα του *Reconfigurable Computing*.

Τα *FPGA*, είναι μία επαναστατική οικογένεια chip, η οποία συνδυάζει τα θετικά του hardware και του software, σε ένα πλήρως προγραμματιζόμενο ολοκληρωμένο

κύκλωμα. Υλοποιούν κυκλώματα, όπως τα ASIC, και μπορούν να επαναπρογραμματιστούν εύκολα προκειμένου να μοντελοποιήσουν μία πληθώρα λογικών συναρτήσεων. Μολαταύτα, τα FPGA υστερούν σε σχέση με τα ASIC, όσο αναφορά τη συνολική αποδοτικότητα και την κατανάλωση. Από την άλλη, ο προγραμματισμός ενός FPGA, μπορεί να πάρει από μερικές ώρες μέχρι μήνες, ανάλογα την πολυπλοκότητα της εφαρμογής, ενώ η κατασκευή του ASIC, πολύ περισσότερο. Για αυτό το λόγο, τα FPGA χρησιμοποιούνται κατά κόρον σε εφαρμογές όπου δεν απαιτείται μέγιστη αποδοτικότητα, ή για την αντικατάσταση ASIC, όσο αυτά βρίσκονται ακόμα στο στάδιο της κατασκευής. Σε κάθε περίπτωση, για κάποιον ο οποίος δεν έχει μεγάλη εμπειρία με τα συστήματα ηλεκτρονικών, η εκμάθηση προγραμματισμού ενός FPGA, είναι σαφώς ευκολότερη από την αφομοίωση των μεθόδων αρχιτεκτονικής ενός ASIC.

Όσο αναφορά τους τομείς που χρησιμοποιούν FPGA τώρα, η ικανότητά των συσκευών αυτών να μεταχειρίζονται απλά ηλεκτρικά σήματα¹, όντας επί της ουσίας ολοκληρωμένα κυκλώματα που βαίνουν σε μία πλακέτα, τα καθιστούν ιδανικά για πολλές εφαρμογές. Χρησιμοποιούνται για παράδειγμα ως ηλεκτρονικοί εγκέφαλοι δορυφόρων, αεροπλάνων, παλμογράφων, Servers, μαγνητικών και αξονικών τομογράφων, συστημάτων επεξεργασίας εικόνας, οθονών, και άλλα πολλά. Οι δυνατές εφαρμογές των FPGA, δεδομένης της ευελιξίας τους, είναι αμέτρητες. Μία εφαρμογή των FPGA λοιπόν, είναι και στο σύστημα ηλεκτρονικών του NSW [8]. Όλα τα back-end electronics, τα οποία αντιπροσωπεύουν τον ενδιάμεσο κρίκο ανάμεσα στα front-end και στους υπολογιστές του CERN, διαθέτουν FPGA για την επεξεργασία των δεδομένων, τα οποία, αν αναλογιστεί κανείς την ενέργεια και την φωτεινότητα του LHC, θα ρέουν με τεράστιους ρυθμούς. Στα front-end electronics, τα FPGA θα χρησιμοποιηθούν ως βοηθητικά κυκλώματα των ASIC, καθώς τα FPGA (με ελάχιστες εξαιρέσεις [30]), δεν είναι ανθεκτικά στο εχθρικό ραδιενεργό περιβάλλον του ATLAS. Μέχρι να κατασκευαστούν τα ASIC για την αναβάθμιση του NSW, στα Test Beams του CERN, θα χρησιμοποιούνται FPGA, τα οποία θα μιμούνται τις λειτουργίες των ASIC. Ένα τέτοιο FPGA, είναι και το FPGA της κάρτας L1DDC, τον προγραμματισμό του οποίου πραγματεύεται η παρούσα διπλωματική εργασία. Προκειμένου όμως να γίνει ξεκάθαρος ο τρόπος με τον οποίο αυτό το...μαγικό κουτί

¹Αυτά τα σήματα είναι ως επί το πλείστον ψηφιακά. Πάρα ταύτα, πλέον ένα FPGA μπορεί να υλοποιήσει και ADC module

προγραμματίζεται, στο παρόν Κεφάλαιο θα μελετηθεί η δομή του FPGA σε αρκετά χαμηλό επίπεδο, από πολλές πλευρές.

6.1 Η Αρχιτεκτονική των FPGA

Το πρώτο FPGA κατασκευάστηκε από την εταιρεία Xilinx[®], το 1985 [18], και από τότε η αρχιτεκτονική των FPGA έχει αλλάξει δραματικά. Πάντα όμως, η γενικότερη φιλοσοφία υλοποίησης λογικών συναρτήσεων από τα FPGA βασίζεται στην εκμετάλλευση δύο ειδών πόρων που υλοποιούνται στη μικροδομή του, από τρανζίστορ που σήμερα φτάνουν μέχρι και τα 20 nm [30]. Οι πόροι αυτοί, χωρίζονται σε δύο μεγάλες κατηγορίες: Τα *Λογικά Μπλοκ (Logic Blocks)*, και τις *Διασυνδέσεις (Interconnect)* [21, 23, 24, 27]. Τα λογικά μπλοκ αποτελούν την καρδιά των FPGA. Είναι αυτά που πραγματοποιούν τις λογικές πράξεις. Τα σήματα που διαχειρίζονται τα μπλοκ αυτά, μεταδίδονται μέσα στο FPGA μέσω ενός περίπλοκου δικτύου δρομολόγησης, το οποίο είναι υπεύθυνο στο να διασυνδέει τα μπλοκ μεταξύ τους. Και αυτά τα δύο είδη πόρων, είναι πλήρως προγραμματιζόμενα. Διαφορετικά προγράμματα, γραμμένα στη γλώσσα VHDL την οποία συντάσσει ο σχεδιαστής του FPGA, θα ενεργοποιήσουν και διαφορετικά λογικά μπλοκ με διαφορετικό τρόπο, και θα υλοποιήσουν και τις ανάλογες διασυνδέσεις προκειμένου τα ηλεκτρικά σήματα, τα προϊόντα της επεξεργασίας από τα μπλοκ δηλαδή, να ταξιδέψουν από κόμβο σε κόμβο, μέχρι να προκύψει το τελικό αποτέλεσμα στην έξοδο του ολοκληρωμένου κυκλώματος. Αυτή την άκρως πολύπλοκη διαδικασία, την υλοποιεί το λογισμικό της κάθε εταιρείας που παρέχει το FPGA. Ο χρήστης δηλαδή, απλά συντάσσει των κώδικα που θέλει να υλοποιήσει, και στη συνέχεια, το λογισμικό με κατάλληλη επεξεργασία ενεργοποιεί τα μέχρι πρότινος νεκρά μπλοκ, και τις σωστές διασυνδέσεις, ώστε να πραγματοποιηθούν αυτά που έχει στο μυαλό του ο χρήστης. Και φυσικά, στα σημερινά FPGA, αυτή η διαδικασία μπορεί να επαναληφθεί άπειρες φορές.

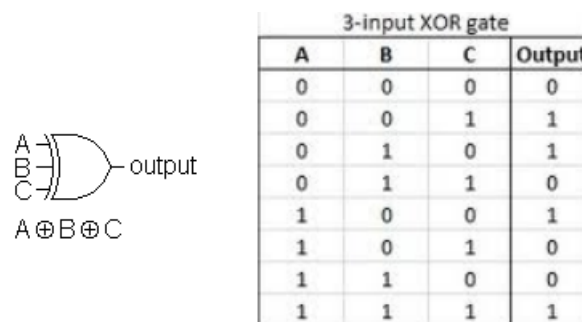
Όταν τα πρώτα FPGA βγήκαν στην αγορά, έκαναν επανάσταση στο χώρο των *Programmable Logic Devices (PLD)*, καθώς χρησιμοποιούσαν περίπλοκα λογικά μπλοκ σε σχέση με άλλα κυκλώματα της ίδιας οικογένειας, τα οποία αύξησαν εκθετικά την αποδοτικότητα των συσκευών αυτών. Τα πρώτα PLD της δεκαετίας του '70 [18], αποτελούντο από μία συστοιχία από απλές λογικές πύλες, οι οποίες ήταν μεταξύ τους ασύνδετες, καθώς σε όλους τους κόμβους υπήρχαν ασφάλειες (fuses). Ο

σχεδιαστής, έπρεπε να αφαιρέσει τις αντίστοιχες ασφάλειες προκειμένου να ενεργοποιηθούν οι αντίστοιχες διασυνδέσεις ώστε να υλοποιηθεί το περίπλοκο κύκλωμα που είχε κατά νου. Στη συνέχεια αυτή η διαδικασία μπορούσε να γίνει μέσω προγραμματισμού, με τη χρήση των *Hardware Description Languages (HDL)*, οι οποίες περιέγραφαν τα κυκλώματα μέσω μίας σύνταξης κατανοητής για τον άνθρωπο. Όμως τα FPGA πήγαν τη τεχνολογία των PLD πολλά βήματα παραπέρα, με την εισαγωγή μίας περίπλοκης συστοιχίας από εκλεπτυσμένα λογικά μπλοκ, διασυνδέσεων, και I/O μπλοκ² που επικοινωνούν με τον έξω κόσμο. Σε αυτή την ενότητα, θα γίνει μία προσέγγιση "από μέσα προς τα έξω", δηλαδή θα μελετηθεί η δομή του FPGA³ ξεκινώντας από τις πιο βασικές μονάδες υπολογισμού που φιλοξενεί, και καταλήγοντας στη γενική επισκόπηση ολόκληρου του ολοκληρωμένου κυκλώματος.

6.1.1 LUTs, Slices, CLBs

Ο βασικός μηχανισμός υλοποίησης λογικών συναρτήσεων από τα FPGA, δε γίνεται μέσω απλών λογικών πυλών, αλλά μέσω *Look-Up Tables* ή *LUT*, τα οποία βρίσκονται παντού στο FPGA, μέσα στα λογικά μπλοκ.

Προκειμένου να καταλάβει κανείς τον τρόπο λειτουργίας ενός LUT, θα δοθεί ένα απλό παράδειγμα από τη βιβλιογραφία [21]. Έστω ότι κάποιος θέλει να κατασκευάσει τον Πίνακα Αληθείας (*Truth Table*), μίας πύλης XOR (Exclusive-OR) τριών εισόδων. Ο πίνακας αυτός είναι ο εξής:

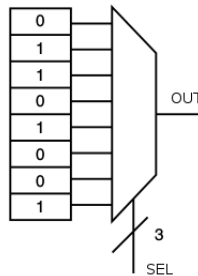


Σχήμα 6.1.1: Πύλη xor τριών εισόδων μαζί με τον πίνακα αληθείας.

²Input/Output Blocks

³Με έμφαση στην αρχιτεκτονική της οικογένειας των FPGA στην οποία ανήκει και το FPGA της L1DDC.

Το γενικό ηλεκτρονικό υποσύστημα που μπορεί να υλοποιήσει μία οποιαδήποτε λογική συνάρτηση της άλγεβρας του Boole, είναι το *LUT* [21, 27]. Ένα *LUT*, μπορεί να περιγραφεί και ως *Πολυπλέκτης (Multiplexer)*, ο οποίος αποτελείται από μία μνήμη (η είσοδος) N bit, και μία έξοδο ενός bit. Για το παράδειγμα του Σχήματος 6.1.I, το *LUT* θα έχει στη μνήμη του τις οκτώ δυνατές καταστάσεις της εξόδου, και σαν είσοδο επιλογής (*SEL*), θα δέχεται ένα διάνυσμα από τρία bit ($2^3 = 8$ δυνατές καταστάσεις). Η επιλογή της εξόδου, θα γίνεται μέσω του *SEL*, το οποίο για την πρώτη κατάσταση όπου όλοι οι εισοδοί είναι μηδέν (000) επιλέγει την πρώτη θέση της μνήμης, που είναι το λογικό μηδέν. Αν $A=1, B=0, C=1$ τότε $SEL="101"$ και επιλέγεται η έκτη θέση, με έξοδο πάλι το λογικό μηδέν.

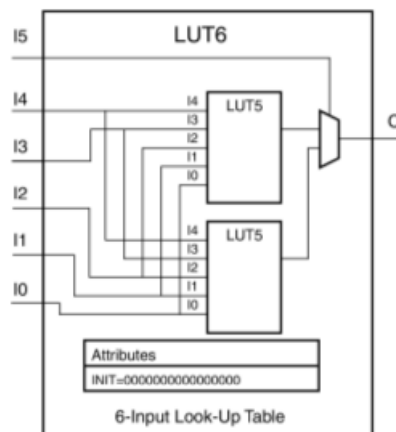


Σχήμα 6.1.II: Το *LUT* ως πολυπλέκτης που υλοποιεί μία πύλη XOR τριών εισόδων [21].

Με τον τρόπο αυτό υλοποιείται η πύλη XOR. Η είσοδος στο FPGA κωδικοποιείται ως το διάνυσμα της επιλογής του πολυπλέκτη, και διοχετεύεται στο κύκλωμα του *LUT* (που λειτουργεί ουσιαστικά ως πολυπλέκτης), το οποίο έχει στις αντίστοιχες θέσεις της μνήμης του τις πιθανές εξόδους της πύλης XOR. Προφανώς, το FPGA δεν μένει εκεί. Ο χρήστης μπορεί να κωδικοποιήσει μία πληθώρα εντολών συντάσσοντας την HDL της επιλογής του (VHDL ή Verilog), οι οποίες εντολές μπορούν όλες να μετατραπούν σε μία μήτρα από λογικές πύλες που συνδέονται μεταξύ τους με μεγάλη πολυπλοκότητα προκειμένου να δώσουν το επιθυμητό αποτέλεσμα. Έτσι, απλούς αλγορίθμους που διαθέτουν εντολές που περιλαμβάνουν, προσθέσεις, αποφάσεις (OR, NOT, WHILE), λούπες, και ο,τι μπορεί να φανταστεί ο προγραμματιστής, ο *Synthesizer* (ο αντίστοιχος compiler για τις HDL) έρχεται για να τους μετατρέψει σε μία συστοιχία λογικών πυλών. Ύστερα, ορίζει το κάθε *LUT* του FPGA να υλοποιήσει και ένα μικρό κομμάτι αυτής της μήτρας. Τελικά, θα συνδέσει τα *LUT* με τέτοιο τρόπο ώστε να παραχθεί το τελικό σήμα στην έξοδο. Πολλά *LUT* συνδεδεμένα με-

ταξύ τους δηλαδή, υλοποιούν οποιαδήποτε λογική συνάρτηση.

Αν και το LUT έχει εδραιωθεί ως το βασικό εξάρτημα υπολογισμού βασικών λογικών πράξεων στα FPGA, το μέγεθος τους είναι αντικείμενο εκτενών μελετών [21, 22]. Από τη μία, τα μεγάλα LUT με πολλές εισόδους προσφέρουν περισσότερες δυνατότητες υλοποίησης λογικών συναρτήσεων ανά λογικό μπλοκ, με αποτέλεσμα να μειώνεται η ανάγκη για χρήση πολλών LUT, το οποίο με τη σειρά του ελαττώνει την πολυπλοκότητα των διασυνδέσεων, και την καθυστέρηση στα σήματα που αυτή συνεπάγεται. Όμως τα μεγάλα LUT, είναι και αυτά με τη σειρά τους αργά, καθώς υλοποιούνται από πιο περίπλοκους πολυπλέκτες. Επίσης, όσο μεγαλύτερα τα LUT, τόσο περισσότερος χώρος στο FPGA πάει χαμένος όταν αυτά μένουν αχρησιμοποίητα. Από την άλλη, τα μικρότερα LUT έχουν λιγότερες δυνατότητες υλοποίησης λογικών συναρτήσεων, όμως είναι πιο γρήγορα, γεγονός που αντισταθμίζεται από την πολυπλοκότητα των διασυνδέσεων, η οποία αυξάνεται προκειμένου μία συστοιχία από απλά LUT να φέρει σε πέρας μία εξεζητημένη λογική συνάρτηση. Εκτός από το πλήθος των εισόδων των LUT, αντικείμενο έρευνας είναι και ο συνολικός αριθμός των LUT που περιλαμβάνει το κάθε λογικό μπλοκ. Σε κάθε περίπτωση, οι μελέτες αυτές ξεφεύγουν από τους στόχους του παρόντος κειμένου. Στη συνέχεια παρατίθεται το LUT (LUT6), που χρησιμοποιούνται σε όλα τα 7-Series FPGA της Xilinx® (Σχήμα 6.1.III). Το FPGA της L1DDC ανήκει σε αυτή την οικογένεια (Xilinx Artix-7 FPGA xc7a200t-3fbg484).

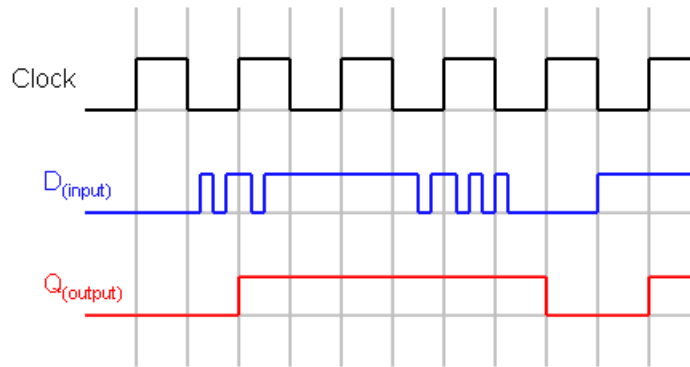


Σχήμα 6.1.III: Το LUT6. Η βασική μονάδα επεξεργασίας ενός Artix-7 FPGA [27].

Όπως μπορεί να διαπιστώσει ο αναγνώστης, το στοιχείο αυτό αποτελείται από δύο επιμέρους LUT πέντε εισόδων. Τα εξωτερικά pins εισόδου του LUT6 είναι έξι, και η έξοδος είναι μόνο μία. Προκειμένου να χρησιμοποιηθεί το LUT για τον υπολογισμό μίας λογικής συνάρτησης, πρέπει να οριστεί μία 64-bit δεκαεξαδική τιμή στη μεταβλητή INIT. Για παράδειγμα, η τιμή 8000000000000000 (πάντα σε δεκαεξαδικό), που αντιστοιχεί στη δυαδική συμβολοσειρά με τιμή 1 στη πρώτη θέση του MSB και 0 στα υπόλοιπα 63 ψηφία, κωδικοποιεί μία πύλη AND με έξι εισόδους, αφού θα δώσει στην έξοδο του LUT το λογικό ένα της πρώτης θέσης μόνο αν όλοι οι εισόδοι είναι αληθείς (που δρουν ως το SEL του πολυπλέκτη). Η προεπιλογή της τιμής του INIT είναι να είναι ίση με το μηδέν, δηλαδή η έξοδος του LUT να είναι μηδέν οποιαδήποτε και αν είναι η είσοδος (γείωση) [27].

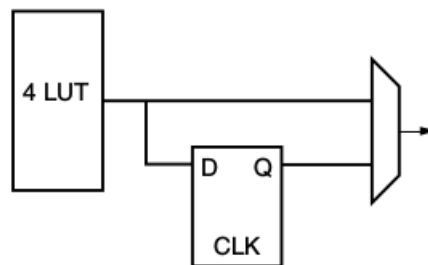
Μέχρι τώρα βέβαια, έχει γίνει αναφορά μόνο σε κυκλώματα *Συνδυαστικής Λογικής (Combinatorial Logic)* και όχι *Ακολουθιακής Λογικής (Sequential Logic)*. Στην πρώτη περίπτωση, η τιμή της εξόδου εξαρτάται μόνο από την κατάσταση των σημάτων στην είσοδο, ως έχουν σε πραγματικό χρόνο, ενώ στη δεύτερη περίπτωση, το αποτέλεσμα στην έξοδο του λογικού κυκλώματος εξαρτάται και από *προηγούμενες καταστάσεις* που έλαβαν χώρα στο κύκλωμα. Έτσι, γεννιέται η ανάγκη για αποθήκευση πληροφορίας, λειτουργία που στο επίπεδο του κυκλώματος την υλοποιούν οι *Registers*. Ο πιο απλός register, είναι η πύλη flip-flop (*Flip-Flop Gate*). Επίσης, προκειμένου να λειτουργήσει ένα κύκλωμα ακολουθιακής λογικής, είναι απαραίτητη η ύπαρξη ενός σήματος χρονισμού δηλαδή ενός *ρολογιού* (βλ. σχ. 3.1.1). Η γενική φιλοσοφία της flip-flop συνοψίζεται στο γεγονός ότι αλλάζει την κατάστασή της μόνο όταν αλλάζει κατάσταση και το ρολόι το οποίο την χρονίζει. Πιο συγκεκριμένα, οι flip-flop συνήθως αλλάζουν κατάσταση μόνο όταν εντοπίσουν ένα rising-edge στο ρολόι χρονισμού. Αυτό σημαίνει ότι σε οποιαδήποτε άλλη στιγμή, η έξοδος της flip-flop δεν αλλάζει και παραμένει η ίδια. Αν για παράδειγμα το ρολόι χρονισμού έχει συχνότητα $f = 40 \text{ Mhz}$ (άρα περίοδο $T = 25 \text{ ns}$), τότε οι flip-flop που θα το χρησιμοποιούν, θα αλλάζουν κατάσταση μία φορά κάθε 25 ns . Παρεμπιπτόντως, αξίζει να τονιστεί ότι ως αλλαγή κατάστασης στη flip-flop, νοείται η αποθήκευση της τιμής εισόδου, το οποίο συνεπάγεται αυτόματα και προώθηση της τιμής αυτής στην έξοδο της πύλης. Άρα ανάμεσα από τα rising-edges, η οποιαδήποτε αλλαγή στα bit εισόδου, αμελείται από τη flip-flop, και η έξοδος της παραμένει ίδια. Ο τρόπος λειτουργίας της flip-flop, μπορεί να συγκριθεί με μία φωτογραφική μηχανή που παίρνει

φωτογραφίες περιοδικά. Ανάμεσα από τα κλικ της μηχανής, τα γεγονότα που συμβαίνουν μπροστά από το φακό, χάνονται.



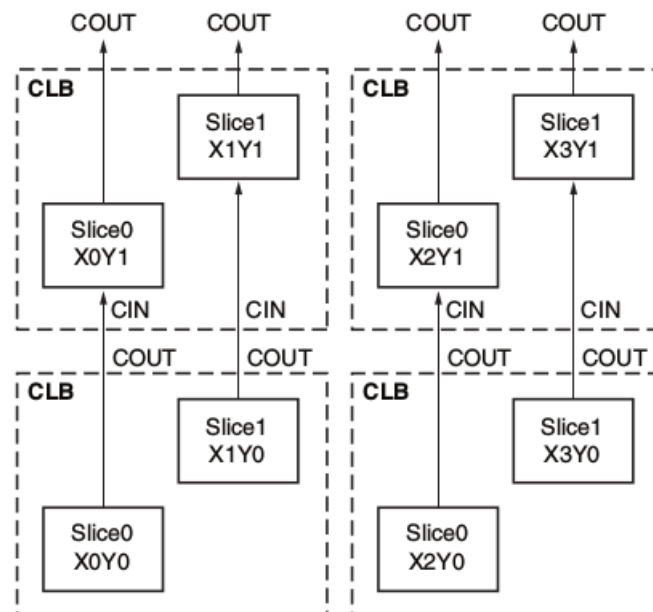
Σχήμα 6.1.IV: Χρονικό διάγραμμα μίας πύλης flip-flop. Φαίνεται πως η κατάσταση της εξόδου αλλάζει μόνο στα rising-edges του ρολογιού, και η τιμή της τότε εξισώνεται με εκείνη της εισόδου. Ανάμεσα από την άφιξη των παλμών του ρολογιού, η κατάσταση δεν αλλάζει.

Οι πύλες flip-flop, όπως και οι υπόλοιποι τύποι registers (π.χ. latches), υλοποιούνται με έναν συγκεκριμένο συνδυασμό λογικών πυλών. Πύλες flip-flop βρίσκονται παντού μέσα σε έναν ηλεκτρονικό υπολογιστή, και περισσότερο σε όσα ηλεκτρονικά υποσυστήματα έχουν να κάνουν με μνήμη. Τα FPGA τώρα, χρησιμοποιούν κατά κόρον πύλες flip-flop, προκειμένου να αποθηκεύσουν τα δεδομένα που προκύπτουν από τους υπολογισμούς των LUT [21, 23, 27]. Ο συνδυασμός των δύο αυτών βασικών ηλεκτρονικών υποσυστημάτων, μαζί με έναν multiplexer που επιλέγει είτε την έξοδο της flip-flop, είτε την έξοδο του LUT απευθείας, δομεί τα λογικά μπλοκ σε όλα τα FPGA [21].



Σχήμα 6.1.V: Το πιο απλό λογικό μπλοκ [21].

Ως προς τα 7-series FPGA της Xilinx®, τα λογικά μπλοκ ονομάζονται *Configurable Logic Blocks* ή πιο απλά *CLB*. Κάθε CLB αποτελείται από ένα ζεύγος από *Slices*. Κάθε slice, διαθέτει τέσσερα LUT6, οκτώ flip-flop (ένα για κάθε LUT πέντε εισόδων που περιέχεται στο LUT6, βλ. Σχ. 6.1.III), μαζί με πολυπλέκτες. Επίσης σε ένα slice υπάρχει και ένα εξειδικευμένο μπλοκ που κάνει προσθαφαιρέσεις δυαδικών. Το μπλοκ αυτό, δεν διασυνδέεται μόνο μέσα στο slice, αλλά έχει και δύο pins που μεταφέρουν το *Carry bit* (το αντίστοιχο του κρατούμενου για την πρόσθεση στο πεδίο των δυαδικών), έξω από το slice στο αντίστοιχο μπλοκ του γειτονικού CLB. Έτσι, τα μπλοκ πρόσθεσης (που καλούνται και *Arithmetic Logic Units (ALU)*), των γειτονικών CLB διασυνδέονται και ανταλλάσσουν μεταξύ τους τα carry bits (μέσω των pins COUT και CIN), προκειμένου να κάνουν συνεργατικά τις προσθέσεις και αφαιρέσεις των δυαδικών αριθμών κατά την επεξεργασία. Η συνδεσμολογία αυτή είναι εμφανής στο Σχήμα 6.1.VI, και καλείται *Fast Carry Chain*.



Σχήμα 6.1.VI: CLB και slice. Κάθε CLB διαθέτει δύο slice, τα οποία διασυνδέονται με γειτονικά slice για διευκόλυνση των υπολογισμών σχετικών με πρόσθεση δυαδικών (η γενικότερη διασύνδεση των CLB με το δίκτυο διασύνδεσης του FPGA μελετάται σε άλλη υποενότητα). Η αλυσιδωτή αυτή διασύνδεση διευκολύνει σημαντικά τη διαδικασία υπολογισμών από το FPGA σαν σύνολο [27].

Εκτός από την υλοποίηση λογικών συναρτήσεων και αριθμητικών πράξεων, τα slice μπορούν να διαμορφωθούν με τέτοιο τρόπο ώστε να μην πραγματοποιούν πράξεις, αλλά να αποθηκεύουν προσωρινά πληροφορίες, σαν μνήμες RAM. Μία τέτοια μνήμη καλείται *Distributed RAM*, και χρησιμοποιεί λειτουργίες εγγραφής μνήμης συγχρονισμένες με το ρολόι εισόδου του slice, και αντίστοιχες λειτουργίες ανάγνωσης (δηλαδή μετάδοση του bit σε κάποια γραμμή που είναι συνδεδεμένη με το slice), με ίδιο ή διαφορετικό χρονισμό. Ένα slice μπορεί να μετατραπεί σε 256-bit RAM, αν δεσμεύσει και τα τέσσερα LUT του για αυτό το σκοπό [27]. Επίσης, τα slice μπορούν να προγραμματιστούν με τέτοιο τρόπο ώστε να καθυστερούν τα σήματα εισόδου, για συγκεκριμένο αριθμό κύκλων ρολογιού (μέχρι 32 κύκλους). Κάθε slice δηλαδή, μπορεί να μετατρέψει τα LUT του σε flip-flops που είναι συνδεδεμένες μεταξύ τους σε σειρά και με κοινό ρολόι, και να συνδέσει και την τελική έξοδο αυτού του LUT με τις εξωτερικές flip-flop του slice. Το αποτέλεσμα είναι να δημιουργείται ένας 32-bit *Shift Register*⁴ ο οποίος δύναται να καθυστερήσει το σειριακό σήμα εισόδου για ορισμένο αριθμό κύκλων, όσο και το πλήθος των flip-flop που θα χρησιμοποιηθεί. Φυσικά, δεν πρέπει να ξεχνάει κανείς και το γενικό σκοπό ύπαρξης των LUT, που είναι ο υπολογισμός λογικών συναρτήσεων. Σε αυτή τη γενική περίπτωση που υλοποιείται στην πλειοψηφία των περιπτώσεων σε ένα FPGA, το LUT μπορεί να θεωρηθεί κατά μία έννοια και ως μνήμη ROM [27] Στο τέλος του παρόντος Κεφαλαίου, μπορεί να βρεθεί το γενικό σχεδιάγραμμα του Slice, όπως αυτό δίνεται από τη Xilinx[®].

6.1.2 Hard Blocks

Εκτός όμως από τα γενικά λογικά μπλοκ (που καλούνται CLB στα FPGA που χρησιμοποιούνται στην κάρτα L1DDC), σε ένα FPGA υπάρχουν και τα λεγόμενα *Hard Blocks* (εξειδικευμένα μπλοκ), τα οποία είναι υποσυστήματα που δεν χρησιμοποιούνται για γενική χρήση, αλλά είναι προκατασκευασμένα από την εταιρεία (σαν μικρά ASIC μέσα στο FPGA), και επιτελούν συγκεκριμένες λειτουργίες που διαφορετικά θα δέσμευαν πολλά ευέλικτα CLB για να υλοποιηθούν. Μέχρι τώρα, έχουν αναφερ-

⁴Shift Register είναι μία συστοιχία από πύλες flip-flop, οι οποίες επειδή είναι συνδεδεμένες σε σειρά (η έξοδος της μίας είναι η είσοδος της επόμενης) με κοινό ρολόι, καθυστερούν το σήμα της πρώτης εισόδου, από το να φτάσει στην τελική έξοδο, για τόσους κύκλους όσο και το πλήθος των πυλών.

θεί τα ALU μέσα στα slice των CLB, τα οποία μάλιστα διασυνδέονται μεταξύ τους με τέτοιο τρόπο ώστε να ανταλλάσσουν τα bit των κρατούμενων μεταξύ τους, προκειμένου να υπολογίζουν συλλογικά τις προσθαφαιρέσεις των δεκαδικών.

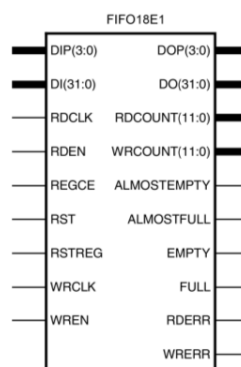
Εκτός από μπλοκ για προσθαφαιρέσεις λοιπόν, υπάρχουν και μπλοκ πολλαπλασιαστών (*Multipliers*) [21, 23]. Πρόκειται για εξειδικευμένα υποσυστήματα μέσα στο FPGA, τα οποία ενεργοποιούνται όταν ο Synthesizer αντιληφθεί ότι μέσα στον κώδικα HDL υπάρχει μία λειτουργία που περιλαμβάνει πολλαπλασιασμούς. Αντί να χρησιμοποιηθεί μία συστοιχία από πολλά LUT, τα οποία θα ανταλλάσσουν μεταξύ τους πληροφορίες για να προβούν σε πολλαπλασιασμούς, τη θέση τους θα πάρει ένα ειδικό μπλοκ που πολλαπλασιάζει δυαδικά αλφαριθμητικά μεταξύ τους, αποδεσμεύοντας έτσι τα CLB από αυτή τη δύσκολη δουλειά, εξοικονομώντας ταυτόχρονα χώρο. Ένα multiplier block άλλωστε δεσμεύει πολύ λιγότερο χώρο σε σχέση με τα CLB που θα απαιτούνταν για να γίνει η λειτουργία του πολλαπλασιασμού.

Ένα άλλο hard block που απαντάται συνεχώς στα FPGA είναι ενσωματωμένες μνήμες RAM (*Block RAM*) που βρίσκονται σε διάφορα σημεία του ολοκληρωμένου κυκλώματος [21, 27]. Εκτός από τη δυνατότητα διαμόρφωσης της λειτουργίας LUT για την υλοποίηση μίας μικρής μνήμης RAM (βλ. υποεν. 6.1.1), μέσα στο FPGA υπάρχουν και βαθιές μνήμες, όπου αν συνυπολογιστεί η χωρητικότητα όλων, μπορεί να φτάσει αρκετές τις δεκάδες Mb. Η ύπαρξη προσωρινών μνημών, κρίνεται γενικά απαραίτητη για την εύρυθμη λειτουργία του FPGA, ιδίως σε περίπλοκες εφαρμογές όπου απαιτείται προσωρινή αποθήκευση πληροφορίας. Μία τέτοια εφαρμογή είναι και η υλοποίηση μνημών *FIFO* μέσα στα FPGA.

First-In First-Out Memory

Η μνήμη *FIFO* (First-In First-Out), είναι ένα είδος μνήμης, όπου χρησιμεύει ιδιαίτερα στην προσωρινή αποθήκευση (buffering) ασύγχρονων δεδομένων [18]. Όταν μία ροή δεδομένων (είτε σειριακή, είτε παράλληλη) μεταδίδεται από ένα σημείο σε ένα άλλο, αν ο χρονισμός δεν είναι ίδιος και στα δύο σημεία, τότε αναπόφευκτα, η ροή των δεδομένων θα αλλοιωθεί. Πιο συγκεκριμένα, αν τα δεδομένα μεταδίδονται με υπερβολικά μεγάλο ρυθμό για τις δυνατότητες του δέκτη, τότε σχεδόν πάντα χρησιμοποιείται μία FIFO για λειτουργίες buffering. Η FIFO μπορεί να έχει ένα ορισμένο πλάτος (δηλαδή τα δεδομένα εισόδου να είναι διανύσματα από δυαδικά ψηφία με

συγκεκριμένο μήκος, π.χ. από ένα μέχρι 16 bit), και ορισμένο βάθος (δηλαδή πλήθος διαθέσιμων θέσεων). Η χωρητικότητα της FIFO υπολογίζεται πολλαπλασιάζοντας το βάθος της μνήμης με το πλάτος της. Συνηθισμένες χωρητικότητες για τις FIFO που υλοποιούνται στα dedicated RAM blocks των 7-Series FPGA είναι 18 kb και 32 kb. Προκειμένου να αποθηκευτούν δεδομένα εισόδου (DIN) σε μία FIFO, πρέπει να χρησιμοποιηθεί ένα ρολόι χρονισμού (WR_CLK) για το sampling των bit ή byte (ανάλογα το πλάτος), και ένα σήμα *write enable* (WR_EN), το οποίο μόνο όταν είναι ψηλά τα δεδομένα εισόδου εγγράφονται στη μνήμη της FIFO. Με τη σειρά που έγιναν οι εγγραφές, τα δεδομένα δρομολογούνται και εκτός της μνήμης (η όλη διαδικασία είναι ανάλογη με την ουρά εξυπηρέτησης σε μία τράπεζα). Ανάλογα υπάρχει ένα ρολόι χρονισμού ανάγνωσης (RD_CLK) και ένα σήμα *read enable* (RD_EN), που όταν είναι ψηλά, τα δεδομένα εξέρχονται από τη FIFO, με το ρυθμό του RD_CLK. Το πλάτος εξόδου, δεν είναι κατ' ανάγκη ίσο με το πλάτος εισόδου. Τέλος, συνήθως υπάρχουν και σήματα που ενημερώνουν το χρήστη για την κατάσταση της μνήμης (αν είναι άδεια, γεμάτη κ.ά.).



Σχήμα 6.1.VII: FIFO block για 7-Series Xilinx® FPGA [27].

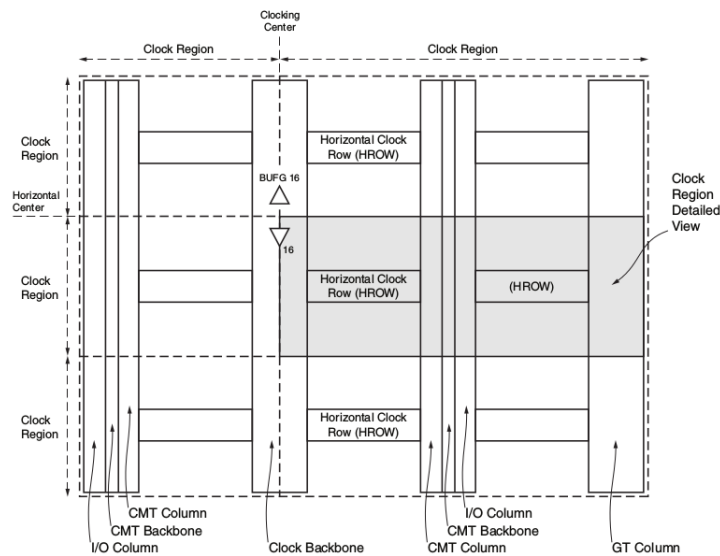
Clock Management Tiles - PLL/MMCM

Εκτός από τα μπλοκ μνήμης (dedicated RAM blocks), τα FPGA της Xilinx® διαθέτουν και εξειδικευμένα υποσυστήματα υπεύθυνα για την παραγωγή χρονισμού για ολόκληρο το κύκλωμα. Αυτά τα μπλοκ, ονομάζονται *Clock Management Tiles*, ή *CMT* [27]. Κάθε CMT, προσφέρει συγκεκριμένα κυκλώματα, υπεύθυνα για την παραγωγή ρολογιών διαφόρων συχνοτήτων, με ελάχιστο jitter, και πολύ συγκεκριμένη φάση σε

σχέση με τα εξωτερικά ρολόγια αναφοράς που τροφοδοτούν τα CMT. Τα CMT είναι οργανωμένα σε στήλες, οι οποίες στήλες είναι πολύ κοντά στα I/O blocks του FPGA. Αυτό συμβαίνει γιατί τα σήματα των ρολογιών που εισέρχονται στο FPGA από την πλακέτα, πρέπει να υφίστανται όσο το δυνατόν λιγότερη καθυστέρηση, jitter, θόρυβο, και γενικότερη παραμόρφωση πριν υποστούν επεξεργασία από το CMT. Το FPGA διαθέτει *Περιοχές Χρονισμού (Clock Regions)*, οι οποίες στα μεγάλα FPGA της σειράς 7, μπορεί να είναι και 24 τον αριθμό. Όλα τα υποσυστήματα του FPGA χρησιμοποιούν ρολόγια για να λειτουργήσουν, και ο σωστός χρονισμός κρίνεται απαραίτητος για την ομαλή λειτουργία ολόκληρου του FPGA. Κυκλώματα στην ίδια περιοχή χρονισμού, θα τροφοδοτούνται και από ρολόγια με παρόμοιες διαφορές φάσης. Μέσα στο FPGA, υπάρχει μία συστοιχία από *Global Clock Trees* που διατρέχει τη ραχοκοκκαλιά του FPGA, και διανείμει τα σήματα των ρολογιών που παράγονται από τα CMT σε όλες τις περιοχές χρονισμού με ελάχιστες καθυστερήσεις, ενώ υπάρχουν και αντίστοιχα τοπικά δίκτυα που διανείμουν τα σήματα μέχρι και σε τρία γειτονικά clock regions [27]. Ο σχεδιαστής του FPGA, μπορεί να χρησιμοποιήσει οποιοδήποτε τύπο δρομολόγησης επιθυμεί. Προφανώς, τα πολυπλοκότερα firmware, που θα απαιτούν και περισσότερα CLB, θα απλώνονται και σε πολλές clock regions, με αποτέλεσμα να ενδείκνυται στην προκείμενη περίπτωση η χρήση του global routing. Σε κάθε περίπτωση, η προσπέλαση γίνεται με τη βοήθεια ειδικών buffers (BUFG, IBUFG, BUFGCTRL, κ.ά.) οι οποίοι δέχονται σαν είσοδο τα ρολόγια που εξέρχονται από τα CMT και έχουν σαν έξοδο τη διασύνδεση με την αντίστοιχη δρομολόγηση.

Το κάθε CMT, αποτελείται από δύο κύρια ηλεκτρονικά κυκλώματα, τα οποία είναι υπεύθυνα για την σύνθεση παλμών χρονισμού. Πρόκειται για το *Phase-Locked Loop*, ή *PLL*, και για το *Mixed-Mode Clock Manager*, ή *MMCM*. Η σειρά 7 των FPGA της Xilinx®, διαθέτει μέχρι και 24 CMT (επομένως μέχρι και 24 περιοχές χρονισμού). Και τα δύο αυτά συστήματα, χρειάζονται ένα ρολόι αναφοράς για να λειτουργήσουν. Τα σήματα αυτά προέρχονται από πολυπλέκτες που έχουν σαν είσοδο τους διάφορους buffers χρονισμού (BUFH, IBUFG κ.λπ.) ο κάθε ένας εκ των οποίων είναι συνδεδεμένος με κάποια δρομολόγηση (τοπικά ή γενικά δίκτυα). Στα παρακάτω σχήματα, απεικονίζεται σχηματικά μία clock region, με το global routing που τη διαπερνά, το CMT, και τα I/O (Σχήμα 6.1.VIII). Επίσης, απεικονίζεται και μία γενική άποψη του CMT, με τους πολυπλέκτες επιλογής ρολογιού, το PLL και δύο ενδεικτι-

κούς buffers χρονισμού (Σχήμα 6.1.IX).

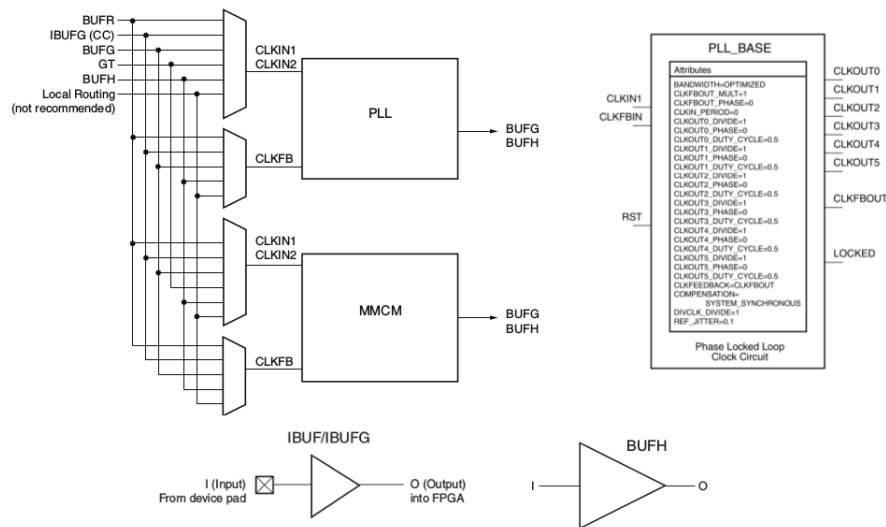


Σχήμα 6.1.VIII: Σχηματική αναπαράσταση των περιοχών χρονισμού των 7-Series FPGA μαζί με τα περιεχόμενά του. [27].

Τα PLL και MMCM, είναι κυκλώματα υπεύθυνα για τη σύνθεση ρολογιών, τα οποία μπορεί να έχουν οποιαδήποτε συχνότητα και διαφορά φάσης σε σχέση με το ρολόι αναφοράς στην είσοδό τους [18, 27]. Επίσης, λειτουργούν ως φίλτρα για το jitter, και ως βασικά υποσυστήματα σε εφαρμογές CDR. Τέλος, δύνανται να μειώνουν την καθυστέρηση των ρολογιών (clock deskewing)⁵ [27]. Αξίζει τέλος να σημειωθεί ότι οι λειτουργίες των PLL και MMCM δεν παρουσιάζουν μεγάλες διαφορές. Το PLL αποτελεί ειδικότερη περίπτωση του MMCM, ως προς τις δυνατότητές του. Εδώ, θα μελετηθεί λίγο περισσότερο ο τρόπος σύνθεσης σημάτων χρονισμού από το PLL, όπως αυτός παρατίθεται στη γενική βιβλιογραφία [18].

Τα κύρια υποσυστήματα ενός PLL είναι: ο *phase detector*, ο ενισχυτής (*amplifier*), ο *voltage-controlled oscillator (VCO)* και ο πολλαπλασιαστής (*multiplier (M)*). Το PLL, δέχεται αρχικά ένα σήμα χρονισμού ως ρολόι αναφοράς (reference clock), το οποίο

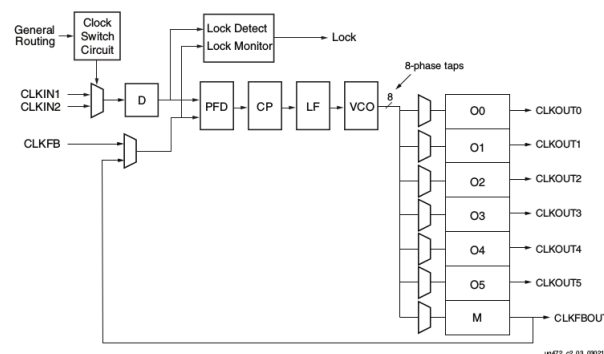
⁵Clock Skew: Έστω δύο flip-flop οι οποίες είναι συνδεδεμένες σε σειρά μεταξύ τους και λαμβάνουν τους ίδιους παλμούς χρονισμού. Όταν υπάρχει διαφορά στο χρόνο που η μία flip-flop εντοπίζει το rising edge του παλμού χρονισμού σε σχέση με την άλλη, η διαφορά αυτή ονομάζεται *clock skew*. Οφείλεται σε καθυστερήσεις διάδοσης των σημάτων μέσα στις γραμμές.



Σχήμα 6.1.IX: Αριστερά: Γενική σχηματική αναπαράσταση ενός CMT. Δεξιά: Το schematic του PLL component, όπως αυτό δίνεται για να το συμπεριλάβει ο προγραμματιστής στον κώδικά του. Κάτω παρατίθενται δύο τυπικοί clock buffers. Ο IBUFG είναι global clock buffer που λαμβάνει τα σήματα χρονισμού απ' έξω από το FPGA και τα δρομολογεί στο γενικό routing, ενώ ο BUFH λαμβάνει την έξοδο από τα CMT και διανέμει τα ρολόγια στο τοπικά δίκτυα [27].

διοχετεύει στο VCO. Το VCO είναι ένα κύκλωμα που με τη σειρά του παράγει και αυτό ένα ρολόι, η συχνότητα και η φάση του οποίου όμως, ελέγχεται από το PLL. Όταν στην είσοδο του PLL εισέλθει ένας παλμός, το PLL με τη βοήθεια του phase detector, εντοπίζει τη διαφορά φάσης μεταξύ του ρολογιού του VCO και του παλμού εισόδου. Το κύκλωμα του phase detector εντοπίζει τη διαφορά ανάμεσα στα rising edges των δύο σημάτων, και όσο μεγαλύτερη είναι αυτή η διαφορά, παράγει αναλόγως και πλατύτερο λογικό παλμό στην έξοδό του. Αυτό σημαίνει πως όταν τα δύο ρολόγια είναι συμφασικά, τότε ο phase detector παράγει έναν λογικό παλμό ένα με μηδενικό πλάτος, δηλαδή ουσιαστικά, το λογικό μηδέν. Όταν όμως ο phase detector παράγει λογικούς παλμούς με μη μηδενικό πλάτος, τα σήματα αυτά θα ενισχυθούν, και θα περάσουν στο VCO, το οποίο θα αλλάξει τη συχνότητα και τη φάση του ρολογιού του, προκειμένου να ευθυγραμμίσει το ρολόι που παράγει με τη συχνότητα του παλμού εισόδου. Έτσι σιγά-σιγά, ο phase detector θα παράγει όλο και στενότερους παλμούς, όσο τα συγκρινόμενα σήματα έρχονται σε φάση. Όταν γίνουν συμφασικά και ο phase detector δεν παράγει τίποτα πια, το PLL πλέον έχει κλειδώσει

(PLL LOCKED). Έτσι, το PLL συγχρονίζει δύο συχνότητες ρολογιών. Με αυτό τον τρόπο συνήθως γίνεται η διαδικασία του CDR (βλ. 5.1.1), και του clock deskewing. Ο άλλος τρόπος λειτουργίας του PLL, είναι η σύνθεση συχνοτήτων από ένα ρολόι αναφοράς. Το ρολόι εισέρχεται στην είσοδο του PLL και στο VCO, και ένας πολλαπλασιαστής (η τιμή του οποίου καθορίζεται από το χρήστη), πολλαπλασιάζει με έναν ακέραιο αριθμό M , τη συχνότητα εισόδου του ρολογιού. Η συχνότητα αυτή θα είναι η συχνότητα του ρολογιού του VCO. Στη συνέχεια το πολλαπλασιασμένο αυτό ρολόι διοχετεύεται πάλι στο PLL ως feedback, προκειμένου το PLL να κάνει συμφασικά τα δύο ρολόγια. Έπειτα, το πολλαπλασιασμένο ρολόι, μπορεί να δρομολογηθεί σε πολλά κυκλώματα που διαιρούν κατά N την πολλαπλασιασμένη συχνότητα. Έτσι, έχοντας ένα ρολόι αναφοράς με συχνότητα f_{ref} , μπορεί να παραχθεί ένα άλλο ρολόι οποιασδήποτε συχνότητας, ορίζοντας τον κοινό πολλαπλασιαστή και τον διαιρέτη αναλόγως. Για παράδειγμα, τα PLL που χρησιμοποιούνται στα 7-Series FPGA, έχουν έξι εξόδους (βλ. Σχ. 6.1.IX). Επομένως μπορούν να παραχθούν έξι ρολόγια με συχνότητα f_i ($i = 0, 1, \dots, 5$), σύμφωνα με τον τύπο $f_i = f_{ref} \frac{M}{N}$. Αυτή είναι και η συχνότερη χρήση του PLL στα FPGA. Κατά κανόνα, η συντριπτική πλειοψηφία των firmware που γράφονται για FPGA, περιλαμβάνουν και ένα PLL, το οποίο χάρη στην ευελιξία του, μπορεί από μία εξωτερική συχνότητα αναφοράς, να παράξει πολλά ξεχωριστά ρολόγια που δύναται να τα δρομολογήσει στα γενικά ή τοπικά δίκτυα, για να χρονίσουν flip-flop, μνήμες RAM και I/O blocks. Εκτός από τα FPGA, τα PLL χρησιμοποιούνται σε όλα τα ψηφιακά κυκλώματα σήμερα, με εφαρμογές σε κινητά τηλέφωνα, μικροεπεξεργαστές, και μητρικές πλακέτες υπολογιστών [18].



Σχήμα 6.1.X: Λεπτομερής σχηματική αναπαράσταση PLL ενός 7-Series FPGA [27].

Transceivers

Αξίζει τέλος να σημειωθεί ότι εκτός από τα προαναφερθέντα hard blocks που τελούν συγκεκριμένες εργασίες μέσα στο FPGA, υπάρχουν και οι λεγόμενοι *Transceivers*⁶. Τα κυκλώματα αυτά, είναι περίπλοκα ηλεκτρονικά υποσυστήματα τα οποία υλοποιούν σειριακές διασυνδέσεις υψηλής ταχύτητας, με άλλα ολοκληρωμένα κυκλώματα εκτός του FPGA. Ο άλλος κόμβος μπορεί να βρίσκεται πάνω στην ίδια πλακέτα, ή σε διαφορετική. Σε κάθε περίπτωση, οι ταχύτητες των 7-Series Transceiver, κυμαίνονται από τα 6.6 Gb/s μέχρι τα 28.05 Gb/s [27], και μπορούν να υλοποιήσουν μία πληθώρα πρωτοκόλλων επικοινωνίας υψηλών ταχυτήτων, όπως PCI Express, SATA, SAS, και Ethernet μέσω οπτικής ίνας ή χαλκού. Στο επόμενο Κεφάλαιο, η λειτουργία του GTP Transceiver ο οποίος υλοποιεί τη διασύνδεση τύπου 1Gb Ethernet του FPGA της L1DDC, θα μελετηθεί με λίγο περισσότερες λεπτομέρειες.

6.2 Το Δίκτυο Διασυνδέσεων του FPGA

Έχοντας πλέον εμβαθύνει στη μικροδομή και στις λειτουργικότητες των λογικών μπλοκ του FPGA, τα οποία μπορεί να είναι είτε γενικής χρήσης (CLBs, Slices, LUTs), είτε ειδικής (block RAM, CMT/PLL), μένει να αποσαφηνιστεί και η αρχιτεκτονική των διασυνδέσεων μεταξύ των μπλοκ αυτών. Έχει ήδη αναφερθεί ότι ο χρονισμός του FPGA, διαδίδεται είτε σε γενικά δίκτυα (global clocking routes), είτε σε τοπικά (local clocking routes). Επίσης, τα γειτονικά slice, ανταλλάσσουν μεταξύ τους πληροφορίες που αφορούν τις προσθαφαιρέσεις των δυαδικών αλφαριθμητικών για να αυξήσουν τη γενική τους αποδοτικότητα. Όμως δεν έχει γίνει καμία αναφορά για το πως τα αποτελέσματα των LUT περνάνε σε άλλα CLB για επεξεργασία, ούτε για το πως οι έξοδοι των block RAM διαβιβάζονται στα LUT, ούτε για το πως ένα σήμα εκτός του FPGA περνάει στα λογικά του μπλοκ. Αυτή η ενότητα πραγματεύεται τις γενικές διασυνδέσεις που υφίστανται σε ένα FPGA.

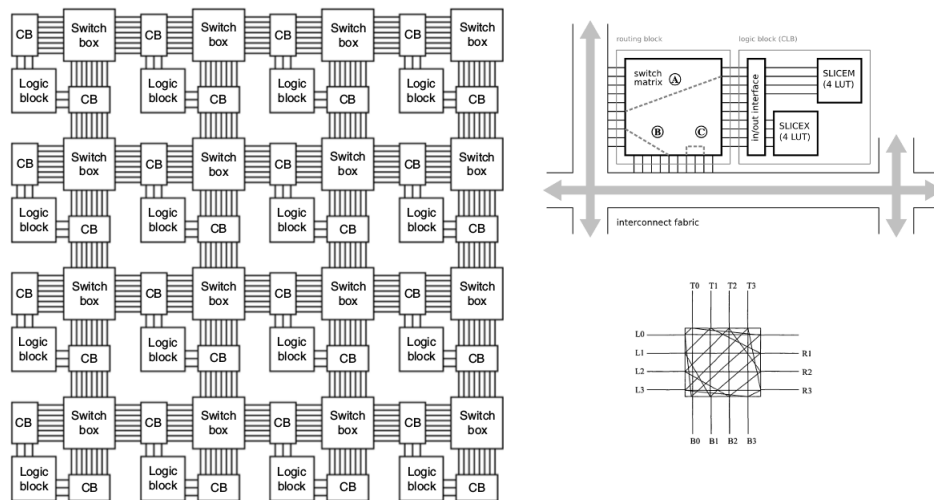
Κατ' αρχάς, αξίζει να τονιστεί ότι η συνολική επιφάνεια του FPGA, δεσμεύεται ως επί το πλείστον από το εσωτερικό δίκτυο διασυνδέσεων και όχι από τα λογικά μπλοκ, είτε αυτά είναι CLB, είτε αυτά είναι hard blocks. Πιο συγκεκριμένα, το 80 – 90% της επιφάνειας του FPGA καλύπτεται από διασυνδέσεις (οι οποίες μπορεί να

⁶Transmitter+Receiver.

θεωρηθούν και απλά ως καλώδια), και μόνο το 10 – 20% καλύπτεται από λογικά μπλοκ [24]. Η γενική μορφολογία των FPGA που σήμερα, είναι η λεγόμενη *Island-Style Architecture* [21]. Αυτό το μοτίβο αρχιτεκτονικής, υποδεικνύει ότι στα FPGA υπάρχει μία πληθώρα κάθετων και οριζόντιων καλωδίων τα οποία μεταφέρουν τα σήματα σε όλα τα μήκη και πλάτη του ολοκληρωμένου κυκλώματος. Τα καλώδια αυτά, διασταυρώνονται μεταξύ τους σε συγκεκριμένα σημεία-κόμβους, οι οποίοι μπορούν να προγραμματιστούν αναλόγως, προκειμένου ένα σήμα να μπορεί να κατευθυνθεί πλήρως μέσα στο FPGA. Πιο συγκεκριμένα, σε ένα FPGA υπάρχουν δύο είδη μπλοκ που είναι υπεύθυνα για τις δρομολογήσεις των σημάτων. Αυτά είναι τα *Connection Blocks (CB)* και τα *Switch Boxes (SB)*. Τα CB είναι στενά συνδεδεμένα με τα CLB. Όλες οι εισοδοί και έξοδοι ενός CLB, πολυπλέκονται στο εσωτερικό του CB, το οποίο με τη σειρά του συνδέεται στον κεντρικό κόμβο που υλοποιεί ένα SB. Το κάθε SB, διαπερνάται από πολλά καλώδια που προέρχονται από ακόμα περισσότερα CB⁷. Η γενική δρομολόγηση των σημάτων, πραγματοποιείται προγραμματίζοντας τα CB/SB, με τέτοιο τρόπο ώστε τα σήματα να διαδίδονται ακέραια μεταξύ των μπλοκ του FPGA, και έτσι ώστε να υλοποιούνται οι απαραίτητες λογικές πράξεις στο σύνολο του FPGA. Τέλος, αξίζει να σημειωθεί πως σε πολλά σημερινά FPGA, προκειμένου να αποφεύγονται φαινόμενα καθυστέρησης σημάτων, που εμφανίζονται όταν αυτά πρέπει να περάσουν μέσα από αρκετούς κόμβους πριν φτάσουν στον τελικό τους προορισμό, τα σήματα μπορούν να διαδοθούν και με μακρύτερα καλώδια (*Long Lines*) [21] που καταλήγουν σε απομακρυσμένα σημεία του FPGA. Έτσι υπάρχει η τοπική δικτύωση, που υλοποιείται με τα SB και CB, και η γενική δικτύωση, που υλοποιείται από τα Long Lines, που ξεκινούν από ένα SB, και καταλήγουν όχι σε γειτονικά, αλλά σε μακρινά SB. Η όλη αρχιτεκτονική, απεικονίζεται στο Σχήμα 6.2.I (τα Long Lines και τα Hard Blocks δεν περιλαμβάνονται).

Εκτός από τις τοπικές διασυνδέσεις μέσα στο FPGA, υπάρχουν και τα μπλοκ που υλοποιούν τη διεπαφή με το εξωτερικό του κυκλώματος. Ανάλογα το μέγεθός του, και ανάλογα τις δυνατότητές του, το FPGA μπορεί να μεταδώσει και να δεχθεί σήματα από πολλά διαφορετικά σημεία της πλακέτας στην οποία βαίνει, όπου τα επίπεδα τάσης και η γενικότερη μορφή αυτών των σημάτων, καλύπτουν ένα μεγάλο εύρος δυνατών πρωτοκόλλων και προτύπων. Τα *I/O Blocks*, είναι ακριβώς αυτά

⁷Στα FPGA της Xilinx®, το CB (που καλείται και Switch Matrix [21]) δέχεται και τα σήματα από τα περισσότερα hard blocks.



Σχήμα 6.2.1: Αριστερά: Σχηματική αναπαράσταση της αρχιτεκτονική τύπου island που εφαρμόζεται στα σημερινά FPGA. Το κάθε CLB συνδέεται με ένα CB στις εισόδους και εξόδους του. Το CB με τη σειρά του συνδέεται με δύο SB τα οποία υλοποιούν τη γενικότερη δικτύωση [21]. Πάνω Δεξιά: Η διασύνδεση του CLB με τα Switch Matrix (το αντίστοιχο SB της Xilinx®). Κάτω Δεξιά: Αναπαράσταση ενός Switch Box. Δέχεται τέσσερις εισόδους σε κάθε πλευρά του, και τις συνδέει μεταξύ τους ανάλογα με τον προγραμματισμό του.

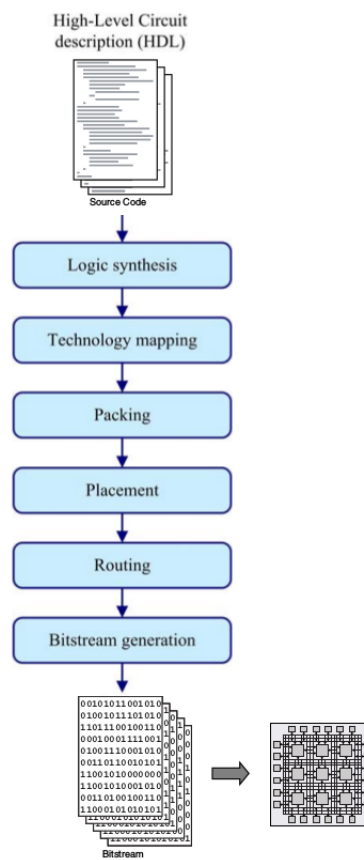
τα μπλοκ που μεταφέρουν τα σήματα από έξω, μέσα στο FPGA, και αντίστροφα. Αυτή η διαδικασία δεν είναι τόσο απλή όσο μπορεί να φανταστεί κανείς, καθώς τα διαφορετικά επίπεδα τάσεων, και η διακύμανση ως προς τη μορφολογία των παλμών που πρέπει να διαχειριστεί το FPGA στα pins του, δημιουργούν την ανάγκη για το σχεδιασμό πολλών διαφορετικών I/O Blocks, τα οποία θα πρέπει και αυτά να προγραμματίζονται, σύμφωνα με τα θέλω του χρήστη. Για παράδειγμα, τα 7-Series FPGA [27], χωρίζονται σε banks. Το κάθε bank, διαθέτει 50 I/O pins, και μία κοινή τάση τροφοδοσίας για τα pins. Στη συντριπτική τους πλειοψηφία, τα pins μπορούν να λάβουν και να μεταδώσουν τόσο διαφορετικά σήματα (LVDS, RSFS, διαφορετικό SSTL κ.ά.), όσο απλά μονοπολικά σήματα (LVCMOS, PCI, κ.ά.). Τα επίπεδα τάσεων που μπορεί να υποστηρίξει το κάθε pin (που είναι συνδεδεμένο με ένα I/O Block), εξαρτάται από το bank στο οποίο βρίσκεται. Ως προς αυτό το ζήτημα, υπάρχουν *High-Range (HR)* banks, άρα και pins, και *High-Performance (HP)* banks. Τα HR, υποστηρίζουν ένα μεγαλύτερο εύρος τάσεων (μέχρι 3.3 V) σε σχέση με τα HP που φτάνουν μέχρι τα 1.8 V. Τα HP interfaces όμως, είναι σχεδιασμένα με τέτοιο τρόπο ώστε

να συμμετέχουν στη μετάδοση πληροφοριών με υψηλές ροές, (π.χ. Gigabit Ethernet) διατηρώντας ταυτόχρονα στο έπακρο την ακεραιότητα των σημάτων που διαχειρίζεται. Σε κάθε περίπτωση, ο σχεδιαστής μπορεί να διαχειριστεί το πρότυπο που θα ακολουθήσει ένα input ή output pin που θέλει να χρησιμοποιήσει στο σχεδιασμό του, με τη βοήθεια των *I/O buffers*. Μερικοί από αυτούς έχουν ήδη αναφερθεί προηγουμένως στο παρών Κεφάλαιο (clock buffers), αλλά φυσικά, υπάρχουν πολλοί άλλοι. Για παράδειγμα, υπάρχουν buffers που δέχονται διαφορικά σήματα από την πλακέτα και τα μετατρέπουν σε μονοπολικά μέσα στο FPGA, και ο χρήστης μπορεί να ορίσει ελεύθερα το πρότυπο τάσεων που θα ακολουθήσει ο buffer αυτός. Τέλος, αξίζει να αναφερθεί το γεγονός ότι εκτός από τα pins και τα μπλοκ εισόδων/εξόδων, το σύστημα που είναι υπεύθυνο για τη μεταφορά των σημάτων εντός και εκτός του FPGA, διαθέτει και ηλεκτρονικά κυκλώματα που προβαίνουν σε διαδικασίες SERDES και sampling των δεδομένων, αλλά η μελέτη αυτών των κομματιών ξεφεύγει από τους στόχους της παρούσας εργασίας.

Έτσι λοιπόν, αρχίζει και ξεκαθαρίζει ο τρόπος με τον οποίο το FPGA τελεί τις λειτουργίες του. Ο σχεδιαστής, προγραμματίζει σε VHDL (ή Verilog) τον αλγόριθμο που θέλει να υλοποιήσει το FPGA, που μπορεί να είναι από μία απλή πύλη OR που θα δέχεται δύο σήματα από διακόπτες πάνω στην πλακέτα και θα διοχετεύει το αποτέλεσμα της πύλης σε μία δίοδο LED, ή ένα πιο πολύπλοκο σύστημα που θα λαμβάνει το σήμα ενός ρολογιού, θα συνθέτει μία πληθώρα συχνοτήτων, και θα αποστέλλει τα περιεχόμενα μίας εσωτερικής FIFO, σειριακά εκτός της πλακέτας, με διαφορετικούς ρυθμούς, ακολουθώντας το πρότυπο LVDS. Στη συνέχεια, ο Synthesizer, διαβάξει τον κώδικα αυτό, και ενεργοποιεί τα αντίστοιχα μπλοκ και τις διασυνδέσεις που θα μεταφέρουν τα σήματα από και προς τα διαφορετικά μπλοκ, προκειμένου να γίνουν οι επεξεργασίες των λογικών τιμών σωστά από τα LUT. Αυτή η διαδικασία, όπως μπορεί κανείς να φανταστεί δεν είναι καθόλου απλή. Για το λόγο αυτό, στη συνέχεια θα αφιερωθεί μία επιπλέον ενότητα που θα περιγράφει τον τρόπο που η πληροφορία "ρέει" από τον κώδικα της VHDL, μέχρι τα κυκλώματα που βρίσκονται μέσα στο FPGA.

6.3 Η Ροή της Πληροφορίας

Η διαδικασία με την οποία μεταφράζεται ένας υψηλού-επιπέδου κώδικας περιγραφής hardware (HDL language), σε ενεργοποιημένα κυκλώματα και διασυνδέσεις μέσα στο FPGA, είναι αντικείμενο εκτενούς μελέτης εδώ και δεκαετίες [21, 22, 23, 24]. Σε αυτή την ενότητα, θα αναφερθεί το γενικό μοτίβο με το οποίο γίνεται η μετάφραση των κωδίκων μίας HDL⁸, σε CLB, block RAM κ.λπ. Το ακόλουθο διάγραμμα μπορεί να χρησιμοποιηθεί σαν σημείο αναφοράς για όσα θα ειπωθούν στη συνέχεια:



Σχήμα 6.3.1: Διάγραμμα της ροής της πληροφορίας που διαμορφώνει τις λειτουργίες σε ένα FPGA [24, 21].

Αντιλαμβάνεται κανείς, ότι η γενική αποδοτικότητα του FPGA, εξαρτάται ευθέως από την ποιότητα με την οποία μεταφράζεται η πληροφορία που κρύβεται μέσα σε

⁸Η οποία συνήθως είναι ή VHDL ή Verilog

μία HDL, σε κυκλώματα του FPGA. Τα βήματα που ακολουθούνται είναι τα εξής [21, 24]:

- *Logic Synthesis*: Πρόκειται για τη διαδικασία κατά την οποία ο Synthesizer μετατρέπει την HDL σε ένα σύνολο από πύλες OR, XOR, NAND κ.λπ. και πύλες flip-flop. Προφανώς, εκτός από το να κατασκευάζει αυτές τις πύλες, ο Synthesizer κάνει και στις αντίστοιχες διασυνδέσεις μεταξύ των πυλών.
- *Technology Mapping*: Μετά τη σύνθεση των συμβατικών πυλών, ο Synthesizer συλλέγει αυτές τις πύλες, και τις μετατρέπει στη φυσική μορφή των πυλών, όπως αυτές υπάρχουν στο FPGA. Δηλαδή, στην περίπτωση των 7-Series FPGA [27], οι λογικές πύλες γίνονται LUT, και τα σήματα εισόδου/εξόδου στον απλό κώδικα HDL μετατρέπονται σε I/O blocks configuration.
- *Packing*: Σε αυτή τη φάση, το σύνολο από LUT και πύλες flip-flop που συνδέονται άμεσα μεταξύ τους, πακετάρονται στα CLB. Το τελικό αποτέλεσμα είναι πλέον CLB και I/O blocks.
- *Placement*: Σε αυτό το στάδιο, ο Synthesizer έχει στα χέρια του τα CLB και τα I/O blocks. Τοποθετεί λοιπόν τα μπλοκ τα οποία δείχνουν να κάνουν υπολογισμούς που να συσχετίζονται με τους υπολογισμούς άλλων μπλοκ, κοντά μεταξύ τους.
- *Routing*: Αφού γίνει η τοποθέτηση των CLB, προγραμματίζονται τα SB/CB προκειμένου να διασυνδεθούν σωστά μεταξύ τους τα CLB, ώστε το ένα να μεταφέρει το αποτέλεσμα της επεξεργασίας που έλαβε χώρα από τα LUT του, στο άλλο. Έτσι όλα τα CLB μαζί, κάνουν συλλογικούς υπολογισμούς περίπλοκων λογικών συναρτήσεων. Τονίζεται ότι κάθε καλώδιο του εσωτερικού δικτύου διασυνδέσεων του FPGA, θα πρέπει στο τέλος να μεταφέρει ένα και μόνο ένα σήμα.
- *Bitstream Generation*: Όταν πλέον η τελική αρχιτεκτονική του FPGA έχει καθοριστεί "στα χαρτιά", μένει να περάσει η πληροφορία μέσα στο FPGA, προκειμένου να ενεργοποιηθούν τα αντίστοιχα LUT, hard blocks, και οι διασυνδέσεις μεταξύ τους. Αυτό γίνεται κωδικοποιώντας ένα σειριακό bitstream από 0 και 1, το οποίο εισέρχεται στο FPGA, προγραμματίζοντας τις μνήμες SRAM που βρίσκονται σε αυτό. Περισσότερα παρακάτω.

Ως επί το πλείστον, τα FPGA πλέον, είναι *SRAM-Based*. Δηλαδή, το τελικό στάδιο του προγραμματισμού τους, γίνεται από μνήμες SRAM⁹ [21, 23, 24]. Κάθε προγραμματιζόμενο μέλος του FPGA, ελέγχεται από ένα κελί μνήμης SRAM. Πιο συγκεκριμένα, ένα LUT, το οποίο υπενθυμίζεται ότι έχει μία τιμή INIT με μήκος 64 bit, θα προγραμματίζεται από 64 μνήμες SRAM, η κάθε μία εκ των οποίων θα ελέγχει και μία θέση στο πεδίο INIT του LUT. Σε άλλα σημεία του CLB, υπάρχουν πολυπλέκτες, το σήμα SEL των οποίων, αντίστοιχα θα ελέγχεται από μνήμες SRAM (αν πρόκειται για έναν 2-to-1 MUX για παράδειγμα, τότε μία μνήμη SRAM αρκεί για τον καθορισμό του SEL). Βέβαια, υπάρχουν και άλλοι τρόποι διαμόρφωσης της λειτουργίας ενός CLB ή slice, που έχουν ήδη αναφερθεί προηγουμένως (μετατροπή σε μνήμη RAM). Όλα αυτά ελέγχονται από μνήμες SRAM. Αντίστοιχα, η ενεργοποίηση και οι μικροαλλαγές στις λειτουργίες των hard blocks γίνονται επίσης από τις αντίστοιχες SRAM. Τέλος, και οι διασυνδέσεις με τη σειρά τους, ελέγχονται από μνήμες SRAM. Ανατρέχοντας στο Σχήμα 6.2.I, μπορεί κανείς να δει τη δομή του Switch Box, το οποίο μπορεί να υλοποιηθεί σε εκείνη την περίπτωση δεκαέξι διαφορετικές διασυνδέσεις στους κόμβους. Υπάρχει λοιπόν μία SRAM για κάθε δυνατή διασύνδεση, για κάθε κόμβο. Όταν η διαδικασία του Routing ενεργοποιήσει κάποιο καλώδιο, τότε η αντίστοιχη SRAM στο αντίστοιχο SB, θα πάρει την τιμή 1 για να κάνει τη σύνδεση.

Η απλότητα των μνημών SRAM, τις έχει καταστήσει ως την καθιερωμένη πρακτική όσο αναφορά τον προγραμματισμό των FPGA. Ο τρόπος κατασκευής τους (τεχνολογία CMOS) δε, επιτρέπει στην εταιρεία που σχεδιάζει το FPGA να εκμεταλλευθεί τις τελευταίες εξελίξεις στον τομέα αυτό [23], ο οποίος είναι εξαιρετικά δημοφιλής. Από την άλλη βέβαια, μία μνήμη SRAM χρειάζεται πέντε ή έξι τρανζίστορ για να υλοποιηθεί, ενώ σε περίπτωση που η τροφοδοσία σταματήσει, η μνήμη της σβήνεται. Αυτό από τη μία είναι καλό, καθώς αυτό ακριβώς είναι που καθιστά τα σημερινά FPGA άπειρες φορές επαναπρογραμματιζόμενα, όμως από την άλλη, προκειμένου να αποθηκευτεί το τελικό firmware στο FPGA, πρέπει να χρησιμοποιηθούν εξωτερικές μνήμες flash.

Έτσι γίνεται πλέον σαφές το πως προγραμματίζεται ένα FPGA. Αρχικά ο προγραμματιστής συντάσσει έναν κώδικα HDL, ο οποίος μεταφράζεται πρώτα σε πύλες, μετά σε LUT και CLB, και τελικά και σε διασυνδέσεις μεταξύ αυτών των μπλοκ. Προκειμένου να υλοποιηθούν όλα αυτά, κατασκευάζεται μία ροή σειριακών ψη-

⁹Static Random Access Memory.

φιακών σημάτων (bitstream), η οποία περνάει από τον υπολογιστή στο FPGA, και προγραμματίζει όλες τις μνήμες SRAM στο FPGA, όπου το κάθε κελί μνήμης, διαχειρίζεται και ένα μικρό κομμάτι του FPGA, είτε αυτό είναι κάποιος πολυπλέκτης ή θέση μνήμης ενός LUT, είτε είναι κάποια πιθανή διασύνδεση στο δίκτυο μεταβίβασης των σημάτων μέσα στο FPGA.

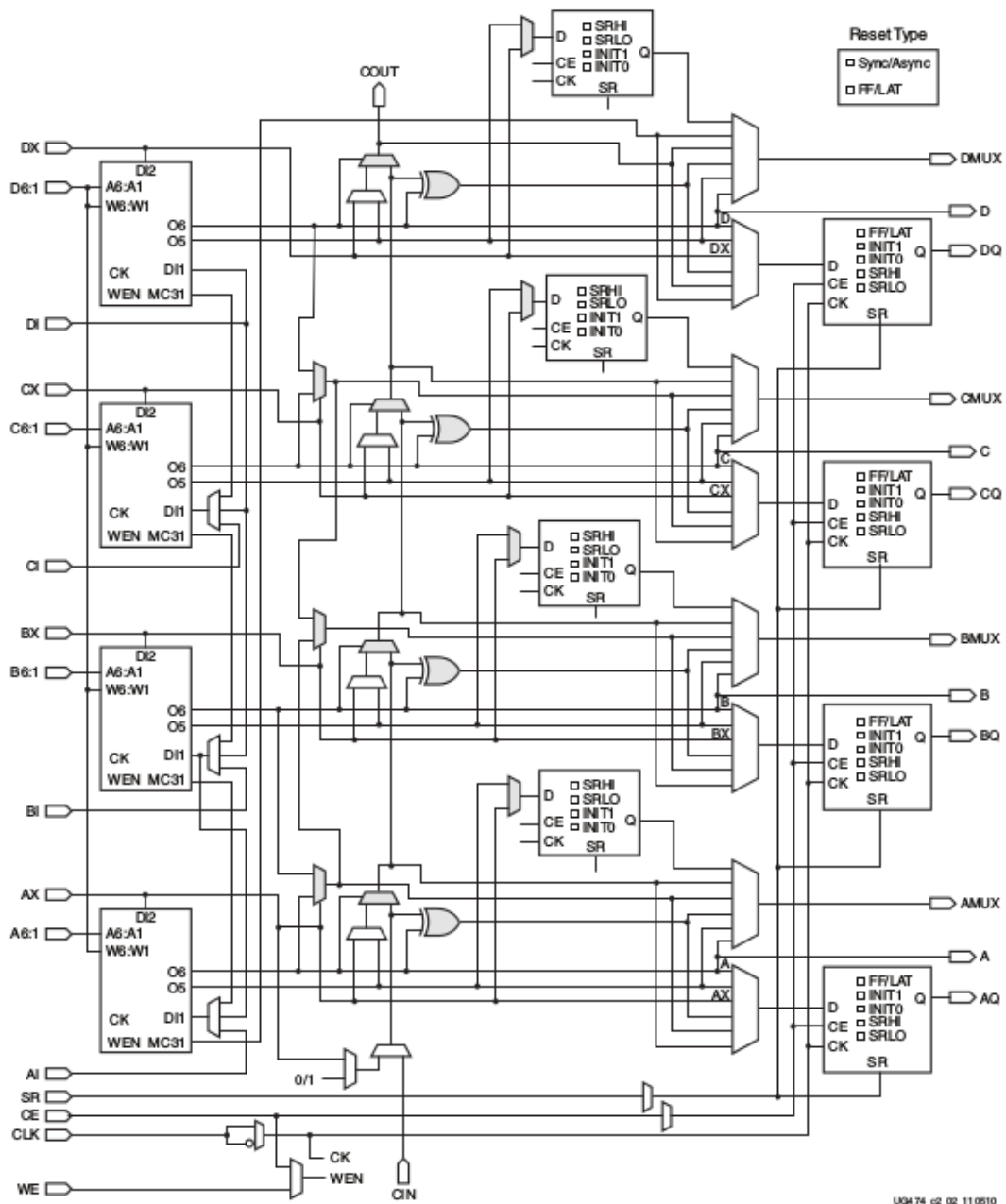
Σύνοψη

Σε αυτό το Κεφάλαιο, ξεκαθάρισε ο τρόπος λειτουργίας των FPGA. Μελετήθηκε η δομή της βασικής μονάδας επεξεργασίας του FPGA, που είναι το LUT. Τα LUT, μαζί με πολυπλέκτες και πύλες Flip-Flop, οργανώνονται σε Slice και CLB, τα οποία απαρτίζουν το κυριότερο μέρος των λογικών μπλοκ ενός FPGA. Εκτός από αυτά τα μπλοκ γενικής χρήσης όμως, υπάρχουν και εξειδικευμένα μπλοκ, τα οποία τελούν συγκεκριμένες λειτουργίες. Τέτοια μπλοκ μπορεί να εκτελούν λειτουργίες μνήμης, ή χρονοισμού. Επίσης, έγινε αναφορά και στον τρόπο με τον οποίο τα λογικά μπλοκ του FPGA συνδέονται μεταξύ τους, μέσα στο τσιπ. Τέλος, αποσαφηνίστηκε και η διαδικασία μετάφρασης του κώδικα, σε υλοποίηση του κυκλώματος μέσα στο FPGA. Πριν κλείσει λοιπόν το παρών Κεφάλαιο, καλό είναι να παρατεθούν μερικά νούμερα για το Xilinx[®] Artix-7 FPGA xc7a200t-3fbg484, το οποίο είναι και το FPGA που χρησιμοποιείται στην κάρτα L1DDC.

Device	Slices ⁽¹⁾	SLICEL	SLICEM	6-input LUTs	Distributed RAM (Kb)	Shift Register (Kb)	Flip-Flops
7A200T	33,650	22,100	11,550	134,600	2,888	1,444	269,200

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices ⁽²⁾	Block RAM Blocks ⁽³⁾			CMTs ⁽⁴⁾	PCIe ⁽⁵⁾	GTPs	XADC Blocks	Total I/O Banks ⁽⁶⁾	Max User I/O ⁽⁷⁾
		Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A200T	215,360	33,650	2,888	740	730	365	13,140	10	1	16	1	10	500

Σχήμα 6.3.Π: Πίνακας με τις προδιαγραφές του Xilinx[®] Artix-7 FPGA xc7a200t-3fbg484 [27].



Σχήμα 6.3.ΠΙ: Ένα Xilinx®7-Series FPGA slice. Στην αριστερή στήλη διακρίνονται τα τέσσερα LUT, ενώ στα δεξιά οι τέσσερις flip-flop (που μπορούν να προγραμματιστούν και ως μνήμες latch). Στο κέντρο διακρίνεται μία σειρά από πύλες και πολυπλέκτες που υλοποιούν το κύκλωμα προσθαιρέσεων. Επιπλέον πολυπλέκτες επιλέγουν τις τελικές εξόδους του slice. Επίσης φαίνονται και pins για την είσοδο σημάτων χρονισμού (CLK), σημάτων εισόδου στα LUT (A6:1) και τα pins της ALU (CIN/COUT) [27].

Firmware για το FPGA της Κάρτας L1DDC

Η παρούσα διπλωματική εργασία, καταπιάστηκε ουσιαστικά με την ανάπτυξη firmware για το FPGA της κάρτας L1DDC. Δεδομένης της πολυπλοκότητας των λειτουργιών του GBTx [17] (το οποίο καλείται το FPGA να υποκαταστήσει), η γραφή κώδικα για το FPGA δεν μπορεί σε καμία περίπτωση να θεωρηθεί εύκολη υπόθεση. Προφανώς, ένα άτομο μόνο του δεν μπορεί να πετύχει τον τελικό στόχο, αλλά χρειάζεται μία ολόκληρη ομάδα συνεργατών, όπου ο καθένας ασχολείται και με ένα μικρό κομμάτι κώδικα. Παρ' όλη την πολυπλοκότητα και τις δυσκολίες όμως, τα λίγα άτομα τα οποία έχουν συμβάλει στην ανάπτυξη του firmware, έχουν καταφέρει μέχρι τις ημέρες συγγραφής του παρόντος κειμένου, να υλοποιήσουν τα περισσότερα, και ίσως και τα πιο περίπλοκα κομμάτια του firmware. Στο Κεφάλαιο αυτό, θα παρουσιαστούν τα διάφορα ξεχωριστά τμήματα του κώδικα για το FPGA της κάρτας L1DDC. Όπως μπορεί όμως να φανταστεί ο αναγνώστης, η δουλειά αυτή είναι ακόμα εν εξελίξει (και θα συνεχίσει να είναι για αρκετό καιρό ακόμα). Αλλά οι στόχοι για το μέλλον είναι σαφείς, και όλα δείχνουν ότι δεν θα περάσει πολύς χρόνος μέχρι το ATLAS να παραλάβει ένα ολοκληρωμένο σύστημα μέσα σε ένα και μόνο FPGA (System-On-Chip).

Σε γενικές γραμμές, το firmware που αναπτύχθηκε στα πλαίσια της διπλωματικής χωρίζεται στα εξής κομμάτια:

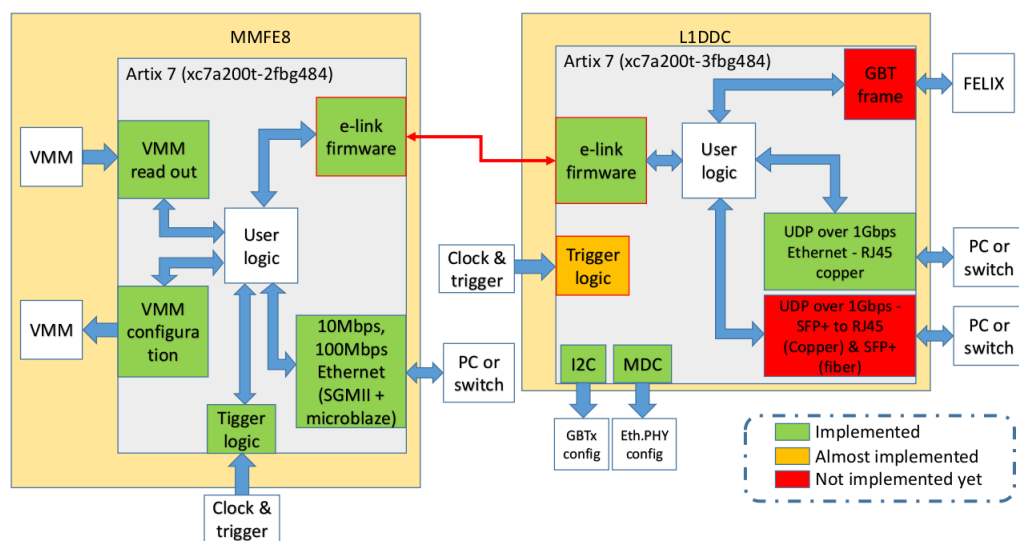
- *E-link*: Υλοποίηση των e-link για τη διασύνδεση με τις front-end, μέσω κάθε miniSAS connector της L1DDC. Ο κώδικας αυτός βρίσκεται σε προηγμένο στάδιο ανάπτυξης. Υπενθυμίζεται ότι πλέον, η ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών του Ε.Μ.Π., δε συμμετέχει αποκλειστικά στην υποστήριξη του κώδικα των e-link για το FPGA της L1DDC, αλλά λαμβάνει μέρος και στον αντίστοιχο κώδικα για το FPGA της MFFE8, το οποίο υποκαθιστά προσωρινά τα ROC/SCA ASIC.
- *Ethernet*: Υλοποίηση του πρωτοκόλλου Ethernet, για τη προσομοίωση της διασύνδεσης με τα back-end electronics. Ο κώδικας αυτός, ο οποίος θα χρησιμοποιεί UDP/IPv4 έχει ουσιαστικά ολοκληρωθεί.
- *I²C*: Υλοποίηση του πρωτοκόλλου I²C, το οποίο θα χρησιμοποιηθεί για τη διεπαφή του FPGA με το GBTx και με το FPGA της mini2. Ο κώδικας αυτός έχει ολοκληρωθεί. Επίσης, με βάση το firmware του I²C, υλοποιήθηκε και το MDIO¹ για τη διεπαφή του FPGA με το Ethernet PHY.

Σε κάθε περίπτωση, ακόμα και αν τα ξεχωριστά κομμάτια του firmware είναι λίγο-πολύ έτοιμα, οι μεταξύ τους διασυνδέσεις μέσα στη λογική του FPGA δεν πρόκειται να είναι εύκολη υπόθεση (ανάπτυξη του γενικού User Logic). Για παράδειγμα, πρέπει να βρεθεί ένας τρόπος ομαλής μετάβασης των δεδομένων από τα e-link στο UDP, ώστε να σταλούν οι πληροφορίες μέσω Ethernet, πίσω στον υπολογιστή. Επίσης υπάρχουν μερικά κομμάτια του firmware τα οποία δεν αναφέρονται στην παρούσα διπλωματική με πολλές λεπτομέρειες ή και καθόλου (π.χ. trigger logic, GBT frame), καθώς δεν αναπτύχθηκαν στα πλαίσια εκπόνησής της.

Εκτός από αυτές τις προαναφερθείσες λειτουργίες, η ευελιξία του FPGA το καθιστά ιδανικό για υλοποίηση αλγορίθμων που αφορούν το debugging της πλακέτας. Για το λόγο αυτό, στο παρών Κεφάλαιο, εκτός από τα πρωτόκολλα επικοινωνίας, θα παρουσιαστούν και τα debugging firmware που αναπτύχθηκαν για τον έλεγχο της ομαλότητας των επιμέρους λειτουργιών της κάρτας L1DDC και των συνοδευτικών πλακετών της (βλ. 5.2). Για την πλήρη κατανόηση των περιεχομένων του παρόντος Κεφαλαίου, ο αναγνώστης θα πρέπει να συμβουλευτεί συχνά τα περιεχόμενα των

¹Management Data Input/Output. Πρόκειται για ένα σειριακό πρωτόκολλο επικοινωνίας παρόμοιο με το I²C, το οποίο μεταφέρει δεδομένα από το Ethernet PHY στον Media Access Controller του firmware του FPGA και αντίστροφα.

Κεφαλαίων 4 και 5. Επίσης, στα Παρατήματα παρατίθενται όλα τα block diagrams των διαφόρων κομματιών του firmware, μαζί με μία περιγραφή των πρωτοκόλλων επικοινωνίας που υλοποιούνται από το firmware. Τέλος, αξίζει να αναφερθεί ότι η γλώσσα που έχει χρησιμοποιηθεί για την ανάπτυξη του firmware είναι η VHDL, και ένας μη εξοικειωμένος αναγνώστης ίσως πρέπει πρώτα να συμβουλευτεί το αντίστοιχο Παράρτημα που περιέχει ένα σύντομο tutorial στη VHDL.

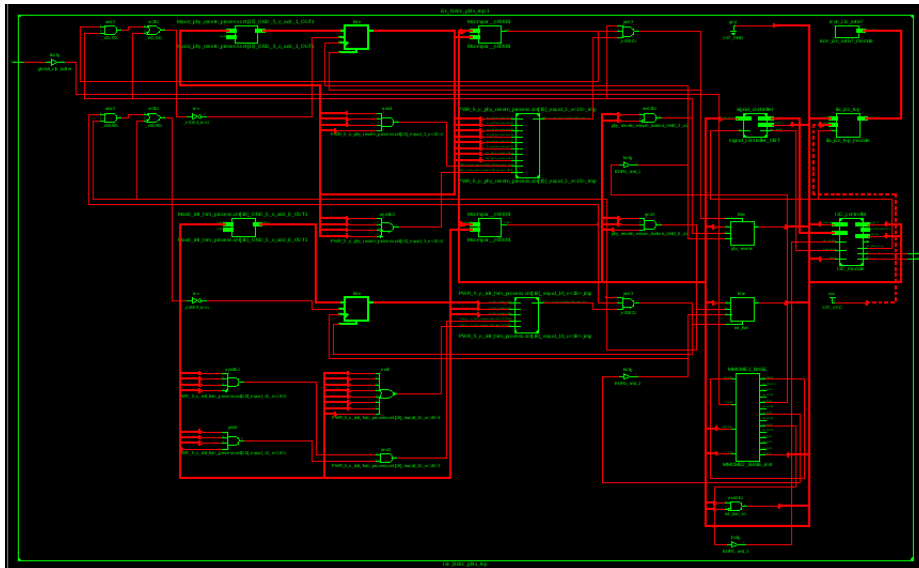


Σχήμα 7.0.I: Διάγραμμα σε μπλοκ του firmware για το FPGA της L1DDC [16].

7.1 I²C FPGA ↔ GBTx

Το πρωτόκολλο I²C, το οποίο περιγράφεται εκτενώς στο αντίστοιχο Παράρτημα, θα υλοποιηθεί κατά κύριο λόγο τη διεπαφή μεταξύ του FPGA και του GBTx στην ίδια πλακέτα, και κατά δευτερεύοντα την επικοινωνία του FPGA με το αντίστοιχο ολοκληρωμένο στη mini2 front-end board, εκτός της L1DDC. Το λογικό κομμάτι του κώδικα αυτού, υλοποιείται από δύο FSM, η μία εκ των οποίων έχει ληφθεί από το Opencores.org, ενώ η άλλη έχει σχεδιαστεί επί τούτου. Ένα RTL Schematic του top_level του κώδικα που υλοποιεί το πρωτόκολλο παρατίθεται παρακάτω (βλ. Σχ. 7.1.I). Υπενθυμίζεται ότι το FPGA θα είναι ο I²C Master, ενώ το GBTx θα είναι ο Slave.

Σε γενικές γραμμές, το top_level του κώδικα, φιλοξενεί ένα MMCM (ανάλογο του PLL αλλά με λίγο περισσότερες δυνατές λειτουργίες) το οποίο λαμβάνει ένα ρολόι αναφοράς, και διαμοιράζει τα επιμέρους σήματα χρονισμού στα άλλα υποσυστήματα του κυκλώματος. Προκειμένου να υλοποιηθεί ο I²C Master από το FPGA, χρησιμοποιείται το component I2C_controller.vhd (από opencores). Πρόκειται για μία FSM που λαμβάνει κατ' αρχάς ένα ρολόι αναφοράς (50 Mhz από το MMCM), και παράγει ένα αργό ρολόι στα 200 kHz, που κατασκευάζεται με τη χρήση ενός απλού counter. Αυτό το ρολόι χρησιμοποιείται για τον χρονισμό της FSM, η οποία όταν αλλάζει καταστάσεις τροποποιεί αναλόγως και τα σήματα εξόδου SCL/SDA προκειμένου να υλοποιηθεί το I²C στις γραμμές εκτός του FPGA. Η διεύθυνση του Slave² στον οποίο απευθύνεται ο FPGA Master, δίνεται σαν σταθερά μέσα στην FSM του I2C_controller.vhd.

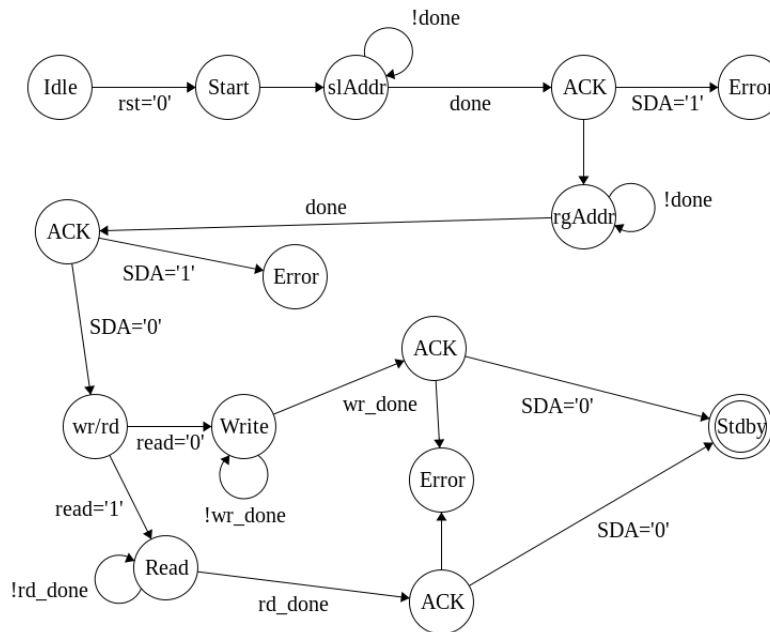


Σχήμα 7.1.1: RTL Schematic του i2c_top-level_11ddc.vhd

Με τη βοήθεια ενός index που διατρέχει τις θέσεις του διανύσματος που κωδικοποιεί τη διεύθυνση του Slave, ο FPGA Master μεταδίδει σειριακά τη διεύθυνση στην SDA. Μόλις τελειώσει, περιμένει για έναν κύκλο να λάβει ACK από τον Slave. Αν η γραμμή του SDA μείνει σε ψηλή τάση, τότε υπήρξε σφάλμα και η FSM πηγαίνει στην error state. Αν ο slave αποκριθεί με ACK κατεβάζοντας την τάση της γραμμής των

²H 7-bit διεύθυνση του GBTx slave είναι "0001111"

δεδομένων, τότε η FSM συνεχίζει μεταδίδοντας τη διεύθυνση του register που θέλει να προσπελάσει. Μετά από οκτώ κύκλους (όσο παίρνει να μεταδοθεί μία διεύθυνση 8 bit), η FSM πάλι περιμένει το ACK από τον slave. Αν δεν το λάβει, πηγαίνει σε error, αλλά αν το λάβει, μεταπίπτει σε μία μεταβατική κατάσταση όπου ελέγχει ένα σήμα εισόδου (read_WriteN), που αν είναι 1 πρέπει να διαβάσει τα περιεχόμενα του register, και αν είναι 0 πρέπει να γράψει κάτι στον register του slave. Σε κάθε περίπτωση, η FSM μεταπίπτει στην αντίστοιχη κατάσταση, όπου προβαίνει σε σειριακή μετάδοση των bits που θα γραφούν στον register (write), ή θα περιμένει τον slave να του μεταδώσει τα περιεχόμενα της register (read). Μόλις τελειώσουν οι διαδικασίες, ο FPGA Master ορίζει το STOP και η διεπαφή σταματά. Στην περίπτωση που γίνεται μετάβαση σε error state, ή μετά το Stop, έχει προστεθεί μία επιπλέον κατάσταση όπου η FSM παραμένει "σιωπηλή" (standby) για μερικούς κύκλους (το πλήθος των οποίων ορίζεται από τη μέγιστη τιμή ενός μετρητή), και μετά επιστρέφει σε IDLE state, προκειμένου να ξεκινήσει πάλι η διαδικασία.



Σχήμα 7.1.Π: Διάγραμμα καταστάσεων της FSM που υλοποιεί το I²C Master στο FPGA.

Η FSM αυτή, έχει αποθηκευμένη στο εσωτερικό της τη διεύθυνση του Slave, όμως η διεύθυνση του register η οποία θα προσπελαστεί, και τα τυχόντα data που πρέπει να γραφούν στον register έρχονται στην FSM σαν εξωτερικά σήματα (AddressIn,

DataIn). Τέλος, το αν η FSM θα γράψει δεδομένα στον register, ή αν απλά θα διαβάσει τα περιεχόμενά του, ελέγχεται επίσης από ένα εξωτερικό σήμα (Read_WriteN).

Προκειμένου να υπάρχει απόλυτος έλεγχος στην I²C MasterFSM, αναπτύχθηκε μία άλλη FSM, (signal_controllerFSM), η οποία επικοινωνεί με το υποσύστημα του I²C Master και αλλάζει τα σήματα, ανάλογα τα θέλω του χρήστη. Την τροφοδοτεί με δεδομένα για το ποια διεύθυνση register πρέπει να διαβαστεί/γραφεί, για το τι δεδομένα πρέπει να γραφούν, και για το αν θα γίνει εγγραφή/ανάγνωση. Ο κώδικάς της FSM αυτής, παρατίθεται παρακάτω:

VHDL Code 7.1: signal_controllerFSM

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity signal_controller is
5
6 port(
7   clk           : in  std_logic;
8   State_FSM_in : in  std_logic_vector(7 downto 0);
9   Read_WriteN  : out std_logic; — sends '1' to read, '0' to write
10  Reg_addr      : out std_logic_vector(7 downto 0); — regAddr to read
11  Data_wr       : out std_logic_vector(7 downto 0)); — data to write
12
13 end signal_controller;
14
15 architecture rtl of signal_controller is
16
17 SIGNAL state : STD_LOGIC_VECTOR(7 DOWNTO 0) := x"00";
18
19 — REGISTER ADDRESSES —
20
21 constant GBTreg_0       : std_logic_vector(7 downto 0) := "00000000";
22 constant GBTreg_1       : std_logic_vector(7 downto 0) := "00000001";
23 constant GBTreg_4       : std_logic_vector(7 downto 0) := "00000100";
24 constant GBTreg_5       : std_logic_vector(7 downto 0) := "00000101";
25 constant GBTreg_9       : std_logic_vector(7 downto 0) := "00001001";
26 constant GBTreg_10      : std_logic_vector(7 downto 0) := "00001010";
27 constant GBTreg_22      : std_logic_vector(7 downto 0) := "00010110";
28

```



```
29  ————— DATA TO WRITE —————
30
31  constant write_reg_0: std_logic_vector(7 downto 0):= (others => '0');
32  —write zeros to register
33  begin
34
35  fsm_process : process(clk , State_FSM_in) is
36
37      variable index : natural range 0 to 6 := 0;
38
39      begin
40
41      if(rising_edge(clk)) then —clk
42      case state is
43
44      when x"00" => — step 0, write/read
45      if(State_FSM_in = x"20") then —check input FSM
46
47      if(index = 0) then —write reg0
48      Data_wr      <= write_reg_0;
49      Reg_addr     <= GBTreg_0;
50      Read_WriteN <= '0';
51      index := index + 1;
52      state <= x"01";
53
54      elsif(index = 1) then —read reg0
55      Read_WriteN <= '1';
56      Reg_addr     <= GBTreg_0; —GBTreg_0
57      index := index + 1;
58      state <= x"01";
59
60      elsif(index = 2) then —read reg1
61      Read_WriteN <= '1';
62      Reg_addr     <= GBTreg_1;
63      index := index + 1;
64      state <= x"01";
65
66      elsif(index = 3) then —read reg4
67      Read_WriteN <= '1';
68      Reg_addr     <= GBTreg_4;
69      index := index + 1;
```

```
70     state <= x"01";
71
72     elsif (index = 4) then —read reg5
73     Read_WriteN <= '1';
74     Reg_addr    <= GBTreg_5;
75     index := index + 1;
76     state <= x"01";
77
78     elsif (index = 5) then —read reg9
79     Read_WriteN <= '1';
80     Reg_addr    <= GBTreg_9;
81     index := index + 1;
82     state <= x"01";
83
84     elsif (index = 6) then —read reg10
85     Read_WriteN <= '1';
86     Reg_addr    <= GBTreg_10;
87     index := 1;  —continuous read, falls back to read reg0
88     state <= x"01";
89
90     else null;
91     end if; —index check
92
93     else state <= x"00";
94     end if; —check input FSM
95
96     when x"01" => — wait to finish write/read of i2c_module
97     if (State_FSM_in = x"D5") then
98     state <= x"02";
99     else null;
100    end if;
101
102    when x"02" => — wait for idle of i2c_module
103    if (State_FSM_in = x"00") then — go back to read/write
104    state <= x"00";
105    else null;
106    end if;
107
108    when others => null;
109
110    end case;
```

```

111         end if ; — clk
112     end process ;
113
114 end rtl ;

```

Τα input/output ports του component υλοποιούν κατά κύριο λόγο τη διεπαφή με την FSM του Master. Στις γραμμές 21-27, διακρίνονται οι αποθηκευμένες διευθύνσεις μνήμης του GBTx που επιθυμεί ο χρήστης να προσπελαστούν (21-27). Ο κώδικας signal_controllerFSM ελέγχει επίσης την ακριβή κατάσταση στην οποία βρίσκεται η FSM του I²C Master, μέσω της εισόδου State_FSM_in (γραμμή 8). Η κύρια κατάσταση της FSM του controller, είναι η x"00" (γραμμή 45). Στη γραμμή 46, ο controller ελέγχει την κατάσταση του I²C Master, και αν αυτός έχει λάβει το ACK από το GBTx μετά το πρώτο addressing (state x"20"), σημαίνει πως είναι έτοιμος να μεταδώσει στο GBTx τη διεύθυνση του register προς προσπέλαση. Ένας index, ελέγχει ποια διεύθυνση μνήμης του GBTx θα μεταδοθεί στη γραμμή του SDA. Για παράδειγμα, αν index=3 (67-71), τότε δίνεται η εντολή διαβάσματος της διεύθυνσης του GBTx με τιμή "00000100". Μετά από αυτό, η FSM περιμένει τον I²C Master να επιστρέψει στην αρχική κατάσταση IDLE (105-106), και να ανακυκλώσει τη διαδικασία. Κατά κανόνα, όταν διαβαστεί ένας register, τότε ο index αυξάνει κατά ένα, και όταν ξεκινηθεί από την αρχή ο controller, θα δώσει εντολή για προσπέλαση άλλης διεύθυνσης, που βρίσκεται στον έλεγχο if της επόμενης τιμής του index (47-88).

Αντιλαμβάνεται κανείς ότι με την αύξηση των δυνατών τιμών στο index, υπάρχουν απεριόριστες δυνατότητες ως προς το πόσες και ποιες διευθύνσεις του GBTx δύναται να διαβαστούν/εγγραφούν από τον I²C Master. Οι δυνατότητες ελέγχου των ACK bit δε, προσθέτει μία δικλείδα ασφαλείας στην όλη διαδικασία. Επιστρέφοντας τώρα στην FSM του I²C Master, αξίζει να σημειωθεί ότι όταν γίνεται ανάγνωση της διεύθυνσης από τον slave, τότε τα δεδομένα μεταδίδονται σειριακά από τον slave στη γραμμή του SDA. Κατά τη μετάδοση των δεδομένων, η FSM του I²C Master προβαίνει σε sampling των δεδομένων, τα οποία και αποθηκεύει σε μία ένα διάνυσμα ψηφιακής λογικής οκτώ θέσεων (deserializing). Έτσι προκύπτει το σήμα DataOut, το οποίο και περιέχει την πληροφορία του register που διαβάστηκε. Προκειμένου να διαβαστούν αυτά τα δεδομένα, ο ευκολότερος τρόπος είναι να υλοποιηθούν δύο βοηθητικά IP Cores της Xilinx[®], τα *Chipscope ILA/ICON*.

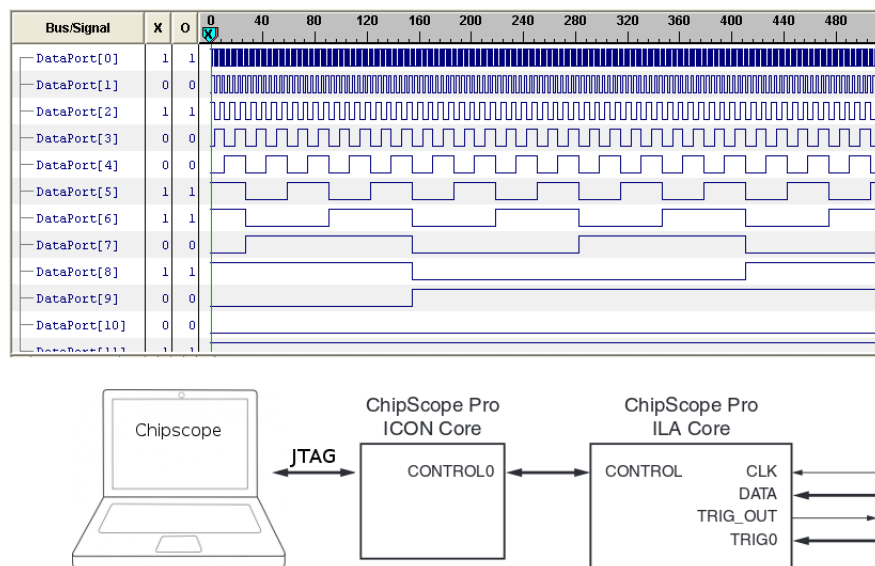
7.1.1 Chipscope ILA/ICON

Το LogiCore™ IP Chipscope™ Pro *Integrated Logic Analyzer (ILA)*, είναι ένα πλήρως προσαρμόσιμο ηλεκτρονικό υποσύστημα, το οποίο μπορεί να χρησιμοποιηθεί για την επισκόπηση οποιουδήποτε εσωτερικού σήματος στο firmware, σε πραγματικό χρόνο [27]. Κατ' αρχάς, το ILA έχει σαν είσοδο ένα ρολόι χρονισμού, και ένα trigger signal. Ένα DATA port εισόδου, του επιτρέπει να συνδεθεί με οποιοδήποτε εσωτερικό σήμα στο component στο οποίο βρίσκεται. Με την ενεργοποίηση του FPGA, και αν το trigger signal είναι αληθές, τότε το ILA ξεκινάει και κάνει sampling των δεδομένων εισόδου του, με το ρυθμό του ρολογιού αναφοράς. Είναι ουσιαστικά μία εξεζητημένη πύλη flip-flop. Στο εσωτερικό του, το ILA διαθέτει έναν buffer συγκεκριμένου βάθους που ορίζει αρχικά ο χρήστης, ο οποίος δέχεται τα δεδομένα εισόδου. Όταν ο buffer γεμίσει, τότε τα δεδομένα του δρομολογούνται σε μία block RAM του FPGA ώστε να διαβαστούν αργότερα (θα ξεκαθαριστεί αμέσως ο τρόπος με τον οποίο συμβαίνει αυτό). Όταν γεμίσει η μνήμη του ILA, τότε το sampling σταματάει μέχρι να αδειάσει ξανά ο buffer (το οποίο γίνεται κατ' εντολή του χρήστη) και η διαδικασία δειγματοληψίας επαναλαμβάνεται.

Το γεγονός ότι το ILA δρομολογεί κάποια δεδομένα του firmware σε μία block RAM, δεν το καθιστά από μόνο του και ιδιαίτερα χρήσιμο. Πρέπει με κάποιο τρόπο αυτά τα δεδομένα να διαβαστούν. Αυτό το προσφέρει το LogiCore™ IP Chipscope™ Pro *Integrated CONTroller core (ICON)*. Το ICON, είναι ένα IP Core component, το οποίο δύναται να διασυνδεθεί με πολλά διαφορετικά IP Cores που είναι υπεύθυνα για το debugging των firmware (AXI Monitor, ILA, VIO, ATC2) [27]. Αυτά τα cores, συνδέονται απ' ευθείας με το ICON, το οποίο υλοποιεί μία διεπαφή με το JTAG. Το JTAG μίας πλακέτας που φέρει FPGA δεν προβαίνει μόνο σε boundary scan του chip, αλλά είναι υπεύθυνο και για την διοχέτευση του bitstream στις SRAM. Επίσης, είναι υπεύθυνο για την επικοινωνία του Chipscope με το χρήστη. Το JTAG συνδέεται με τον υπολογιστή του χρήστη μέσω ενός συμβατικού καλωδίου USB. Έτσι, το Chipscope έχει επαφή με το χρήστη. Όταν δε το Chipscope είναι συνδεδεμένο με ένα ILA core (η σύνδεση γίνεται μέσω των CONTROL ports του), τότε μετά την ολοκλήρωση του sampling από το ILA, το Chipscope δρομολογεί τα δεδομένα που αποθηκεύτηκαν στη block RAM, στο JTAG, και άρα στο χρήστη. Ο χρήστης μπορεί να επιθεωρήσει τα αποτελέσματα του sampling του ILA μέσω του περιβάλλοντος του Chipscope, το οποίο απεικονίζει τα δεδομένα σε μορφή κυματομορφών. Το Chipscope επίσης,

δίνει τη δυνατότητα διαγραφής του buffer του ILA, προκειμένου να ξεκινήσει νέος κύκλος sampling. Το ICON μπορεί να διαχειριστεί μέχρι και 15 διαφορετικά ILA cores [27]. Επίσης αξίζει να σημειωθεί ότι κάθε ILA δεσμεύει 156 LUT, 270 flip-flop και μία block RAM. Κάθε ICON από την άλλη, δεσμεύει 90 LUT και 108 flip-flop για να υλοποιηθεί. Τα εργαλεία των ILA/ICON είναι από τα χρησιμότερα IP Cores της Xilinx. Χρησιμοποιούνται όχι μόνο για την ανάγνωση εσωτερικών σημάτων (όπως στο I²C firmware), αλλά και σε διαδικασίες debugging ενός περίπλοκο firmware με πολλά επίπεδα.

Εδώ ολοκληρώνεται η ανάλυση για το τρόπο λειτουργίας του I²C firmware. Ένα γενικό reference για την αρχιτεκτονική του, μπορεί να βρεθεί στα Παραρτήματα.



Σχήμα 7.1.III: Πάνω: Το διάγραμμα των σημάτων που λαμβάνει δείγματα ο ILA. Κάτω: Σχηματική αναπαράσταση της διασύνδεσης ILA/ICON και υπολογιστή μέσω JTAG.

7.2 Ethernet/UDP Interface

Προκειμένου να πραγματοποιηθεί η επικοινωνία του FPGA με τον ηλεκτρονικό υπολογιστή, απαιτείται η υλοποίηση ενός τμήματος των OSI Layers μέσα στο FPGA (βλ. Παράρτημα Πρωτοκόλλων Επικοινωνίας). Τα OSI (Open Systems Interconnection)

Layers, είναι διάφορα αλληλοεπικαλυπτόμενα επίπεδα που προτυποποιούν και κατηγοριοποιούν τις λειτουργίες που πρέπει να επιτελέσει ένας κόμβος, προκειμένου αυτός να μπορεί να λάβει μέρος σε ένα δίκτυο επικοινωνιών. Τα επίπεδα αυτά είναι: Application, Presentation, Session, Transport, Network, Data Link και Physical, με σειρά από το ανώτερο στο κατώτερο [19]. Στο firmware της L1DDC, τα τμήματα των επιπέδων που πρέπει να μοντελοποιηθούν μέσα στο FPGA είναι τα: Transport, Network, Data Link και Physical.

Όπως έχει ήδη γίνει σαφές, τα δεδομένα από τα e-link, θα συγκεντρώνονται από το FPGA, και θα πακετάρονται προκειμένου να σταλούν μέσω Ethernet στον υπολογιστή που θα είναι συνδεδεμένος με την πλακέτα της L1DDC. Για το σκοπό αυτό τα δεδομένα αρχικά πακετάρονται σύμφωνα με το πρότυπο που ορίζει το πρωτόκολλο UDP (Transport Layer). Στη συνέχεια αυτά τα πακέτα προωθούνται στο Network Layer, όπου υλοποιείται το πρωτόκολλο IPv4. Στο firmware της L1DDC, αυτές οι λειτουργίες τελούνται από κοινούς κώδικες VHDL που μπορούν να κατεβαστούν ελεύθερα από το OpenCores.org. Τα πράγματα γίνονται πιο περίπλοκα στην υλοποίηση των Data Link και Physical επιπέδων, όπου είναι απαραίτητη η χρήση εξειζητημένων IP Cores που διαθέτει η Xilinx[®] για τέτοιες εφαρμογές. Πρόκειται για τρία IP Cores: *Tri-mode Ethernet MAC*, *Ethernet SGMII PCS/PMA* και *7-Series GTP Transceiver*. Η εφαρμογή των τριών αυτών IP Cores, σε συνδυασμό με το firmware από το OpenCores, επιτρέπει στον χρήστη να κατασκευάσει ένα πλήρες σύστημα networking στο FPGA, το οποίο μπορεί και υλοποιεί πρωτόκολλο Gigabit Ethernet μέσω χάλκινου καλωδίου.

Ο τρόπος με τον οποίο δομείται αυτό το σύμπλεγμα διαφορετικών components μέσα στο firmware είναι ο εξής: Κατ' αρχάς, η διασύνδεση των IP cores, έχει ως εξής: TEMAC ↔ PCS/PMA-SGMII ↔ Transceiver. Ο Transceiver, είναι ένα ηλεκτρονικό υποσύστημα που μπορεί να μεταδίδει σειριακά δεδομένα, μέσω διαφορετικών ζευγών, με πολύ υψηλές ταχύτητες. Τα περισσότερα FPGA σήμερα, διαθέτουν ενσωματωμένα hard block που υλοποιούν Transceivers³, οι ταχύτητες των οποίων φτάνουν στην τάξη των *Gbps*. Το σύστημα αυτό, γενικά μπορεί να επικοινωνήσει μόνο με άλλα ανάλογα μπλοκ που βρίσκονται εκτός του FPGA. Ένας Transceiver για παράδειγμα, μπορεί να επικοινωνήσει με συστήματα που εφαρμόζουν τις διασυνδέσεις με οπτικές ίνες, να υλοποιήσει PCI Express πρωτόκολλα, και πρωτόκολλα Ethernet. Εν

³Το FPGA που χρησιμοποιείται στην L1DDC, διαθέτει 16 Transceivers, όπου ο κάθε ένας μπορεί να υλοποιήσει διεπαφές Ethernet over copper, Ethernet over fiber, PCIe κ.λπ.

προκειμένω, στην L1DDC υλοποιείται το πρωτόκολλο Ethernet, και ο Transceiver του FPGA επικοινωνεί με ένα αντίστοιχο εξωτερικό κύκλωμα της Marvell®. Πρόκειται για το *Marvell 88E1111 Integrated Ultra Gigabit Ethernet Transceiver*, το οποίο είναι ένα εξωτερικό Ethernet PHY, που επικοινωνεί αμφίδρομα με το FPGA, και συγκεκριμένα με τον Transceiver. Η διεπαφή αυτή επιτρέπει στα Ethernet frames του δικτύου να εισέρχονται στο FPGA, και στο FPGA να στέλνει frames στο δίκτυο. Στις μεγάλες ταχύτητες, όπου απαιτείται DC-balancing για να μην αλλοιώνονται τα σήματα στις γραμμές, εφαρμόζεται συνήθως 8b/10b κωδικοποίηση (βλ. αντίστοιχο Παράρτημα) η οποία προσφέρει και ευκολότερο clock recovery στα κυκλώματα που συμμετέχουν στη διεπαφή. Υπενθυμίζεται επίσης ότι η 8b/10b κωδικοποίηση, διαθέτει comma characters, τα οποία στέλνονται συνεχώς από τον κόμβο που θέλει να μεταδώσει σύντομα δεδομένα, καθώς τα comma characters έχουν το προσόν να διευκολύνουν το clock recovery. Όταν αυτά λαμβάνονται σωστά από τον δέκτη, εκείνος γνωρίζει ότι είναι συμφασικός με τη ροή δεδομένων και μπορεί να κάνει σωστό sampling.

Εν προκειμένω λοιπόν, ο χρονισμός ολόκληρου του κυκλώματος που υλοποιεί το Ethernet firmware, γίνεται από ένα ρολόι το οποίο ανακτά ο Transceiver από τη γραμμή επικοινωνίας με το Marvell ETH PHY. Σαν ρολόι αναφοράς ($f = 125\text{ Mhz}$), ο Transceiver λαμβάνει ένα διαφορετικό σήμα (MGTREFCLK), που προέρχεται από έναν κρύσταλλο υψηλής ποιότητας πάνω στην πλακέτα, και μέσω αυτού, τα εξειδικευμένα PLL του Transceiver συνθέτουν ρολόι από την εξωτερική ροή δεδομένων. Αυτό το ανακτημένο ρολόι, περνάει μέσα από ένα εξωτερικό MMCM (ανάλογο του PLL, βλ. υποεν. 6.1.2) που βρίσκεται σε ένα CMT του FPGA. Αυτό συνθέτει δύο ρολόγια (userclk, userclk2 με συχνότητες 125 Mhz και 62.5 Mhz αντίστοιχα) τα οποία χρονίζουν όλα τα components του κυκλώματος.

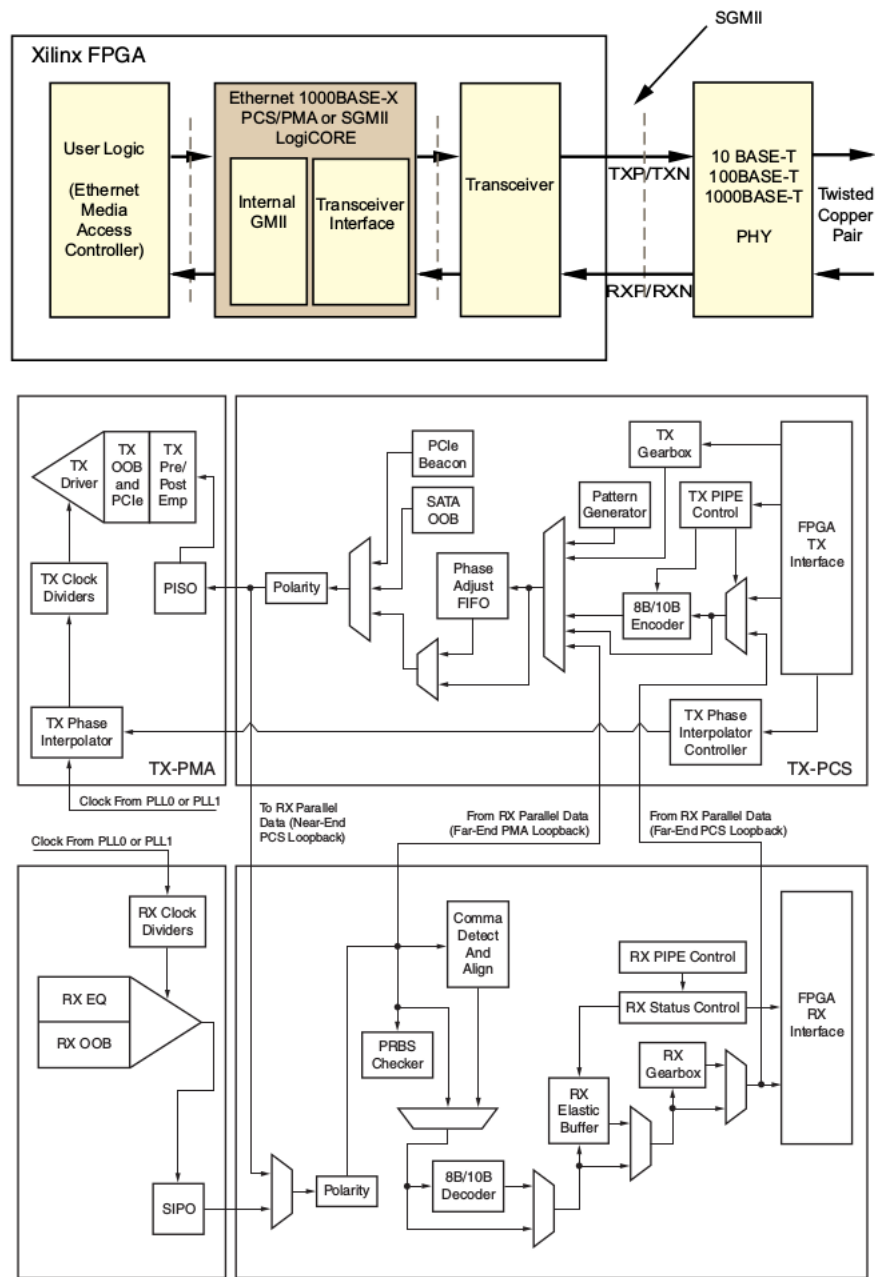
Όταν λοιπόν το UDP/IP block, που υλοποιεί το Transport και Network Layer, θέλησει να στείλει δεδομένα ένα επίπεδο κάτω, στο Link Layer, θα λάβει τον χρονισμό του από τα ρολόγια του MMCM (δηλαδή ουσιαστικά από το ανακτημένο ρολόι), και θα προωθήσει το πακέτο που έχει κατασκευάσει (βλ. Σχ. Α΄.2.2) στον EMAC. Ο EMAC, είναι ένα αυστηρά ορισμένο (βάσει των προδιαγραφών του IEEE 802.3™[29]) υποσύστημα, το οποίο τελεί τη διασύνδεση του Physical Layer με τα ανώτερα επίπεδα OSI. Πιο συγκεκριμένα, ο EMAC (ή και TEMAC), λαμβάνει τα πακέτα από το Network Layer (εν προκειμένω πακέτα IPv4 μαζί με UDP)⁴, και τα "τυλίγει"

⁴Το UDP προτιμάται στην παρούσα εφαρμογή, επειδή έχει πολύ ελαφρύ overhead

με το Ethernet Frame (βλ. Σχ. Α΄.2.2 και Α΄.2.1). Σε ευθεία αναλογία με τις λειτουργίες που τελεί το GBTx, το οποίο επισυνάπτει στα δικά του frame, πεδία FEC (για διόρθωση λαθών από το δέκτη σε περίπτωση που αλλοιώθηκαν τα δεδομένα του frame κατά τη μετάδοση), ο TEMAC προσθέτει στο Ethernet Frame ένα πεδίο *Frame Check Sequence (FCS)*. Το πεδίο αυτό έχει μήκος 4 byte, και υπολογίζεται συναρτήσεως των περιεχομένων του frame, εφαρμόζοντας έναν πολυωνυμικό τύπο 32ου βαθμού [29]. Αξίζει να σημειωθεί επίσης, ότι η διεπαφή μεταξύ IP block και EMAC block δεν είναι άμεση. Αν και τα δύο components χρονίζονται από τα ίδια ρολόγια (*userclk/2*), μεταξύ τους παρεμβάλλεται μία FIFO, (*user_side_fifo*), η οποία προβαίνει σε buffering των δεδομένων που ρέουν μεταξύ των δύο επιπέδων.

Ο EMAC προωθεί το Ethernet frame που κατασκεύασε, στο Physical Layer, που υλοποιείται από τον PCS/PMA. Ο τρόπος μετάδοσης των δεδομένων στο firmware, γίνεται σύμφωνα με το πρότυπο *SGMII (Serial Gigabit Media Independent Interface)*, το οποίο είναι μία υποπερίπτωση του *GMII* (που είναι παράλληλο) [28]. Το Marvell PHY, όντας chip που υλοποιεί διεπαφή Ethernet μέσω χάλκινου καλωδίου και όχι οπτικής ίνας, θέτει άνω όριο στην ταχύτητα στα 1 *Gbps*, και υποχρεώνει τον PCS/PMA να εφαρμόζει το πρωτόκολλο 1000BASE-T μαζί με το SGMII (το 1000 υποδεικνύει ταχύτητα 1 *Gbps*, ενώ το -T διασύνδεση μέσω χαλκού). Επίσης, ο SGMII adapter που υλοποιεί το περιβάλλον ανάπτυξης ISE, διαθέτει και εσωτερικούς counters, οι οποίοι δύνανται να καθυστερήσουν τη διασύνδεση σε 100 ή και 10 *Mbps*, κατά τη βούληση του χρήστη. Η μείωση της ροής μπορεί να κριθεί απαραίτητη αν υπάρχει θόρυβος στις γραμμές που να εμποδίζει υψηλές συχνότητες μετάδοσης.

Μεταξύ του SGMII και του Transceiver, παρεμβάλλεται ένα interface που περιλαμβάνει πύλες flip-flop για συγχρονισμό των δεδομένων. Ο Transceiver, λαμβάνει τα δεδομένα από το SGMII, και τα επεξεργάζεται ώστε να τα αποστείλει σειριακά στις διαφορετικές γραμμές που οδηγούν στο εξωτερικό PHY. Ο GTP Transceiver είναι ένα αρκετά περίπλοκο κύκλωμα, το οποίο αποτελείται από πολλά υποσυστήματα, που τελούν διάφορες λειτουργίες. Μία από αυτές είναι για παράδειγμα το 8b/10b encoding. Σύμφωνα με το documentation της Xilinx[®], ακόμα και το reset και το power-up του μπλοκ είναι περίπλοκη διαδικασία, καθώς απαιτούνται καλά χρονισμένα σήματα με τη σωστή αλληλουχία για να λειτουργήσει σωστά ο Transceiver [28]. Για το λόγο αυτό, μαζί με τον Transceiver περιλαμβάνονται και δύο ειδικές εξωτερικές FSM, οι οποίες υλοποιούν πολλά από τα σήματα που ελέγχουν τον Transceiver.



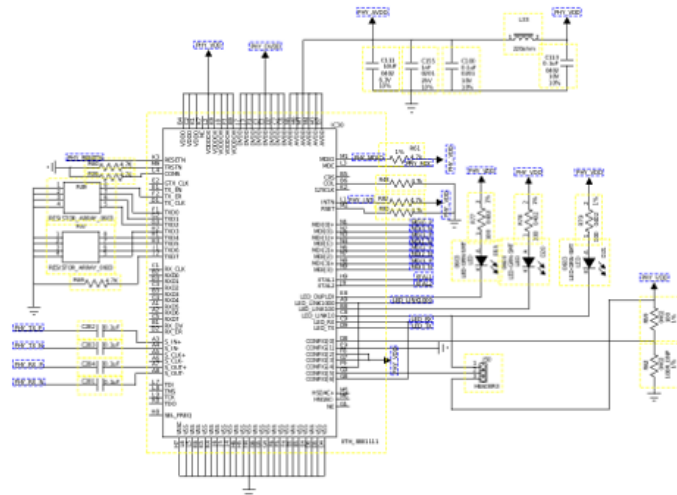
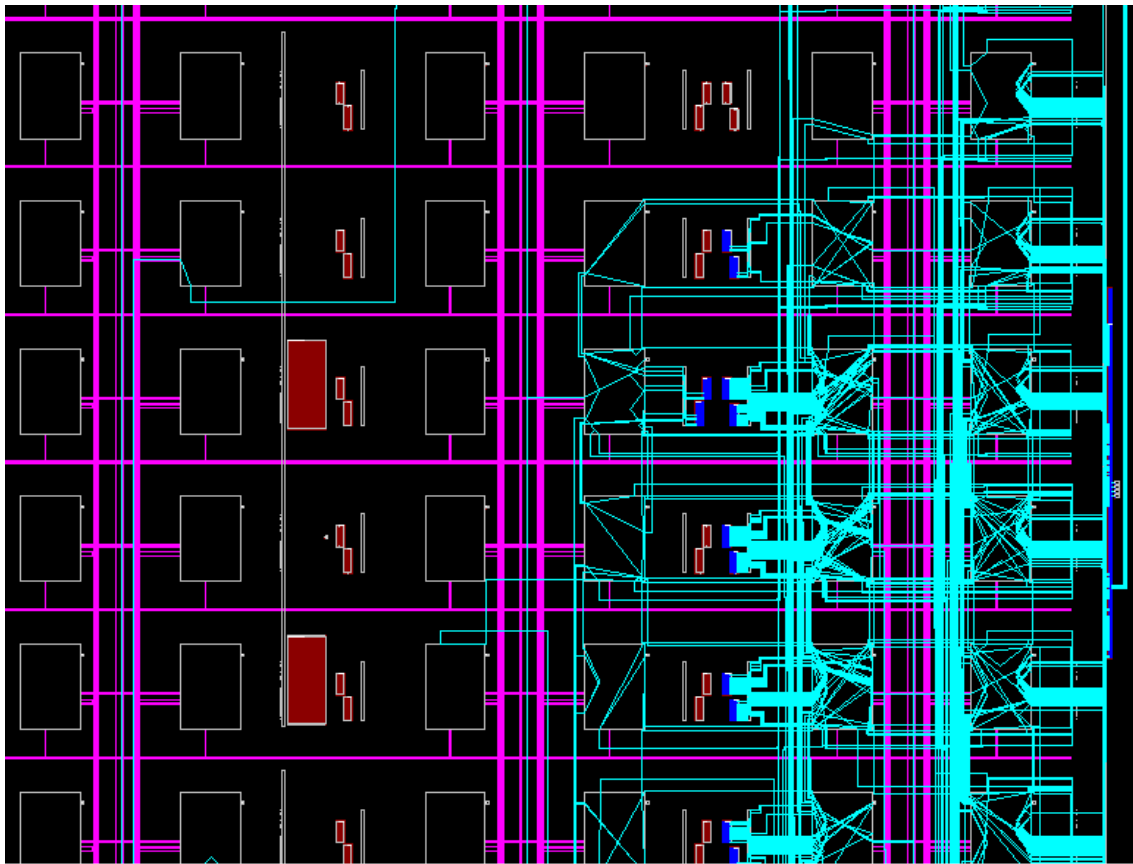
Σχήμα 7.2.1: Πάνω: Η διασύνδεση των τριών IP Cores, τόσο με τα ανώτερα στρώματα OSI, όσο και με το εξωτερικό Ethernet PHY. Κάτω: Τα διάφορα υποσυστήματα του Xilinx® GTP Transceiver που χρησιμοποιεί το FPGA της L1DDC [28].

Η διασύνδεση με το Marvell PHY, γίνεται δρομολογώντας τις σειριακές γραμμές του

εσωτερικού Transceiver που βρίσκεται στο FPGA, στα αντίστοιχα pins που υλοποιούν τη διεπαφή με το Marvell PHY. Υπάρχει επίσης και ένα MDIO Interface, το οποίο και μπορεί να διαχειριστεί το FPGA ως ο Master του, ώστε να διαβάσει τους registers του εξωτερικού PHY, οι οποίοι φέρουν σημαντικές πληροφορίες για την κατάστασή του. Επίσης μπορεί να κάνει και εγγραφές στους registers, για να διαμορφώσει τον τρόπο λειτουργίας του. Το MDIO έχει αναφερθεί ξανά στο παρών Κεφάλαιο, στην ενότητα για το I²C. Το MDIO έχει και αυτό υλοποιηθεί στο FPGA της L1DDC, καθώς είναι συγγενικό πρωτόκολλο με το I²C, και το migration αποδείχτηκε ιδιαίτερα εύκολο. Έτσι υπάρχει μία πλήρης διεπαφή με το chip που υλοποιεί το Ethernet over Copper.

Μέχρι τώρα όμως, δεν έχει γίνει καμία νύξη για τον τρόπο με τον οποίο το firmware λαμβάνει δεδομένα από το δίκτυο. Στην πραγματικότητα δεν υπάρχουν σημαντικές διαφορές. Ο Transceiver λαμβάνει τα 8b/10b σειριακά δεδομένα, τα μετατρέπει σε παράλληλα, αφού πρώτα τα αποκωδικοποιήσει σε συμβατικά, και τα δρομολογεί στο SGMII. Αυτό με τη σειρά του τα προωθεί στον EMAC, ο οποίος αποκωδικοποιεί το FCS, και προβαίνει σε τυχόν διορθώσεις του frame. Έστερα απομονώνει το ληφθέν πακέτο που βρίσκεται στο Data field, το οποίο αν είναι UDP/IPv4, το firmware μπορεί και το "τεμαχίζει" περαιτέρω, ώστε να διαβάσει τελικά την πληροφορία που του εστάλη από το δίκτυο. Έτσι, γίνεται πλέον σαφής ο τρόπος υλοποίησης του Ethernet για το FPGA της L1DDC. Κατ' αναλογία με το GBTx, υλοποιεί μία διασύνδεση με το σύστημα των υπολογιστών (ή των back-end electronics) μέσω Ethernet. Ολόκληρο το Ethernet frame, διαθέτει αρκετά πεδία που χρησιμεύουν στη διόρθωση σφαλμάτων (όπως το FEC του GBT frame). Τα τελευταία επίπεδα του OSI τώρα, μπορούν και κάνουν sampling των πραγματικών δεδομένων που βρίσκονται "βαθιά" μέσα στο frame, προκειμένου να τα ελέγξει το κομμάτι του User Logic. Προφανώς, μπορούν αναλόγως να αποστείλουν στο δίκτυο τα πακέτα που κατασκευάζουν, τα οποία είναι τα αποτελέσματα από το DAQ των e-link, και τα monitoring data. Έτσι κατασκευάζεται μία ολοκληρωμένη διεπαφή μεταξύ L1DDC και υπολογιστή.

Εδώ ολοκληρώνεται η ανάλυση για το τρόπο λειτουργίας του Ethernet firmware. Ένα γενικό reference για την αρχιτεκτονική του, μπορεί να βρεθεί στα Παραρτήματα, μαζί με το αντίστοιχο block diagram.



Σχήμα 7.2.Π: Πάνω: Απεικόνιση του Transceiver (δεξί άκρο της φωτογραφίας), των γύρω CLB και των μεταξύ τους διασυνδέσεων από το FPGAEditor. Τα γειτονικά λογικά μπλοκ υλοποιούν τις FSM που ελέγχουν τον Transceiver. Κάτω: Διάγραμμα του ολοκληρωμένου κυκλώματος του εξωτερικού Ethernet PHY.

7.3 E-Link Interface

Σε αυτή την ενότητα, θα παρατεθούν λεπτομέρειες σχετικά με τον τρόπο που υλοποιείται το e-link μεταξύ της κάρτας L1DDC και των MMFE8, από το FPGA της L1DDC, μέσω όλων των θυρών miniSAS. Προκειμένου να θεωρηθεί πλήρης όμως η ανάλυση της διεπαφής που τελείται μέσω του e-link, θα μελετηθεί ο τρόπος λειτουργίας του πρωτοκόλλου και από τη μεριά του FPGA της MMFE8.

Ανατρέχοντας σε προηγούμενα Κεφάλαια, όπου αναφέρεται ο τρόπος υλοποίησης των e-link, μπορεί κανείς να μελετήσει τον τρόπο με τον οποίο μεταφέρονται τα δεδομένα μεταξύ των front-end boards και της L1DDC. Χονδρικά, η L1DDC αποστέλλει τα σήματα χρονισμού στα συνοδευτικά ASIC της MMFE8, προκειμένου να υπάρχει συγχρονισμός ως προς τη μετάδοση και λήψη των δεδομένων μεταξύ των δύο πλακετών. Στο ίδιο e-link (αλλά σε άλλο ζεύγος διαφορικών), η L1DDC αποστέλλει στις MMFE8 και τα configuration data που προέρχονται από το FELIX (slow control). Η MMFE8, δρομολογεί στην L1DDC τα δεδομένα που συλλέγονται από τα VMM στη γραμμή για το DAQ. Στην τελική πλακέτα της L1DDC όλες αυτές οι λειτουργίες συλλογής και αποστολής δεδομένων μέσω e-link, θα πραγματοποιηθούν από το GBTx (βλ. Σχ. 5.1.III). Από τη μεριά της MMFE8, τα ROC και SCA ASIC, που τη στιγμή σύνταξης του παρόντος κειμένου βρίσκονταν σε προχωρημένο στάδιο ανάπτυξης, είναι εκείνα που θα εκτελούν τις αντίστοιχες λειτουργίες. Υπενθυμίζεται ότι όλα τα παραπάνω, έχουν αναλυθεί εκτενώς σε προηγούμενα Κεφάλαια τις εργασίας αυτής.

Τα δύο Xilinx® Artix-7 FPGA που βρίσκονται στις πλακέτες L1DDC και MMFE8, οφείλουν να υποκαταστήσουν τα ASIC και ως προς την υλοποίηση του e-link. Έτσι λοιπόν, προκύπτει η ανάγκη για σύνταξη του κατάλληλου προγράμματος σε VHDL, το οποίο να υλοποιεί τις προαναφερθείσες λειτουργίες, και για τα δύο chip. Η Ομάδα Πειραματικής Φυσικής Υψηλών Ενεργειών του Ε.Μ.Π., σε συνεργασία με άλλα μέλη του ATLAS Collaboration, έχει αναπτύξει τους κώδικες αυτούς σε ικανοποιητικό πλέον στάδιο [16]. Τα firmware για την κάθε πλακέτα δεν έχουν πολλές διαφορές μεταξύ τους, και ο κώδικας ο οποίος θα μελετηθεί στη συνέχεια, ουσιαστικά υλοποιεί και τα δύο μέλη της διασύνδεσης ταυτόχρονα, μέσα στο ίδιο FPGA, προσομοιώνοντας έτσι τις διασυνδέσεις των e-link μεταξύ διαφορετικών components στο ίδιο ολοκληρωμένο κύκλωμα. Έτσι, ελέγχεται εύκολα η ορθότητα του firmware και για

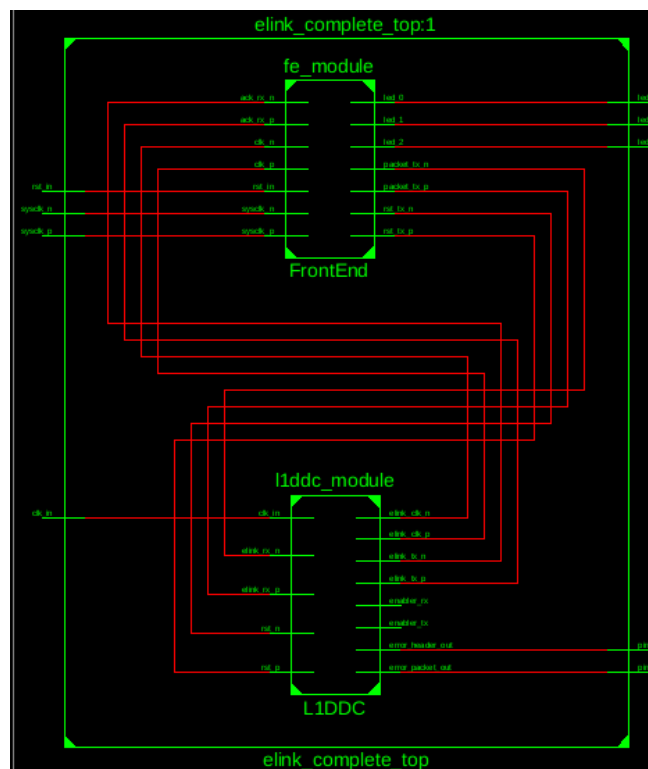
τις δύο πλακέτες συγχρόνως. Στην επόμενη ενότητα όπου θα παρατεθεί μία μεθοδολογία debugging για τα e-link, θα παρουσιαστούν τα αποτελέσματα της διεπαφής αυτής, σε πραγματικές πλέον συνθήκες.

Το firmware των e-link, έχει σχεδιαστεί με τέτοιο τρόπο ώστε να υλοποιεί τη διασύνδεση με λίγο διαφορετικούς τρόπους, ανάλογα τα θέλω του χρήστη. Κατ' αρχάς, η ροή των δεδομένων στη διεπαφή μπορεί να είναι είτε στα 80 *Mbps* είτε στα 160 *Mbps*. Η ταχύτητα των 320 *Mbps*⁵, δεν μοντελοποιείται από το παρών firmware. Ανάλογα τη ροή των δεδομένων, τότε χρησιμοποιούνται και διαφορετικά ρολόγια χρονισμού από τα components που προβαίνουν στο SERDES των δεδομένων. Επίσης, τα δεδομένα μπορεί να στέλνονται απλά, ή με 8b/10b κωδικοποίηση. Στο μέλλον, θα προστεθεί και το πρότυπο HDLC, μαζί με ένα ειδικό πρωτόκολλο που θα κωδικοποιεί τα δεδομένα TTC.

Ο κώδικας προσομοίωσης της διεπαφής των e-link (`elink_complete_sim`), περιέχει στο `top_level` του δύο components, κάθε ένα εκ των οποίων αντιπροσωπεύει και τη μία πλακέτα-κόμβο. Τα components αυτά ονομάζονται `fe_module` και `l1ddc_module`. Η ανάλυση θα ξεκινήσει από το `l1ddc_module`. Το component αυτό, λαμβάνει ένα ρολόι (`clk_in`) από την πλακέτα, και το χρησιμοποιεί αρχικά ως ρολόι αναφοράς για ένα εσωτερικό MMCM, και επιπρόσθετα το δρομολογεί και εκτός, μέσω ενός διαφορικού buffer μετάδοσης (OBUFDS). Η έξοδος αυτή, καταλήγει στο `fe_module`, το οποίο και μετατρέπει το διαφορικό σήμα σε απλό, και αντίστοιχα τροφοδοτεί ένα δικό του MMCM με το ληφθέν ρολόι. Με αυτό τον τρόπο, προσομοιάζεται η αποστολή των σημάτων χρονισμού από την L1DDC στην MMFE8.

Το component της `l1ddc` συνθέτει μέσω του MMCM του, τέσσερα διαφορετικά ρολόγια που θα χρησιμοποιηθούν από τα λογικά υποσυστήματά του. Δημιουργούνται δύο ρολόγια των 40 *Mhz*, με μικρή μεταξύ τους διαφορά φάσης, ένα ρολόι 80 *Mhz* και ένα 160 *Mhz*. Αντίστοιχα, το component της front-end, λαμβάνει αυτό το ρολόι μέσω του e-link, και το διοχετεύει (αφού πρώτα το μετατρέψει σε μονοπολικό) στο δικό του MMCM. Έτσι συντίθενται ακριβώς τα ίδια ρολόγια και από τη front-end,

⁵Η οποία θα υφίσταται για κάποια e-link που θα μεταφέρουν τα δεδομένα DAQ από τις πλακέτες front-end που θα βαίνουν στο εσωτερικό του δίσκου του ανιχνευτή όπου υπάρχει και μεγαλύτερη ροή σωματιδίων.



Σχήμα 7.3.I: Σχηματική αναπαράσταση του top_level του firmware που προσομοιάζει τα e-link.

για εσωτερική χρήση.

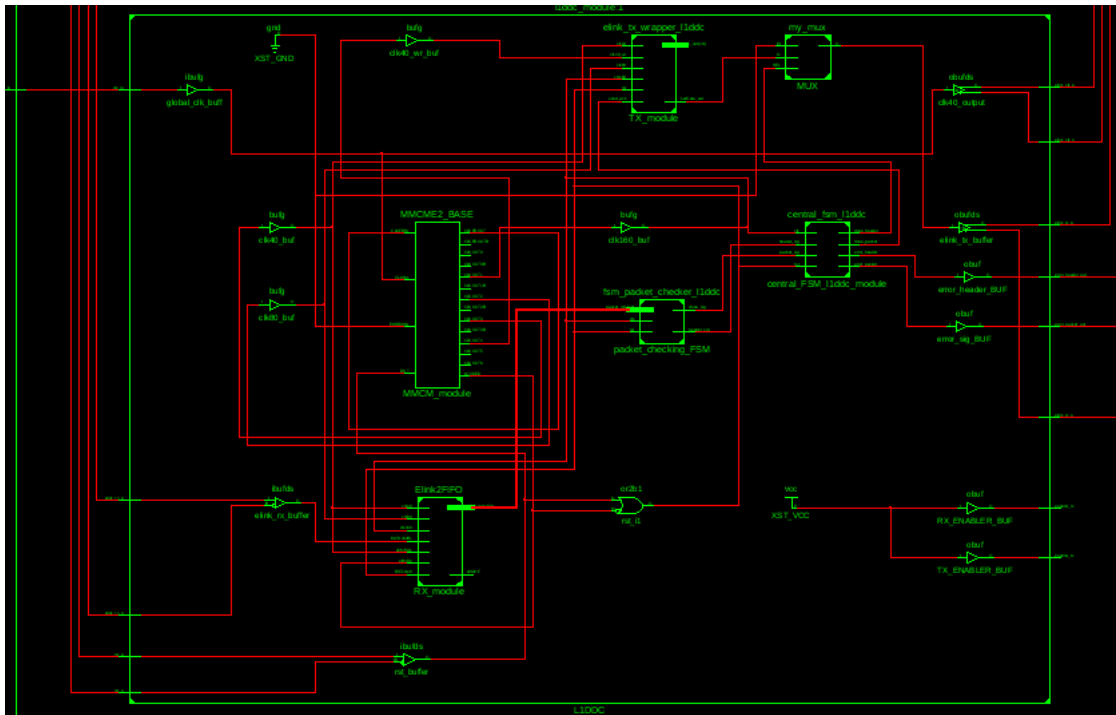
Τα δύο κύρια υποσυστήματα που υλοποιούν τη διεπαφή του e-link (και στα δύο modules) είναι τα ELINK2FIFO και FIFO2ELINK. Το ELINK2FIFO, είναι ένα component που δέχεται κατ' αρχάς μία σειριακή είσοδο (DATA1bitIN), και τρία ρολόγια χρονισμού. Αυτή η σειριακή είσοδος, προέρχεται από το e-link. Επειδή όμως τα σήματα του e-link είναι διαφορετικά, χρειάζεται και ένας buffer (IBUFDS) που να μετατρέπει το διαφορετικό σήμα εισόδου που προέρχεται από το e-link, σε μονοπολικό, το οποίο ύστερα δρομολογείται εσωτερικά και απ' ευθείας στην είσοδο DATA1bitIN. Τα σειριακά αυτά δεδομένα που μπαίνουν μέσα στο ELINK2FIFO, αποθηκεύονται προσωρινά σε ένα διάνυσμα δύο ή τεσσάρων θέσεων (ανάλογα την ταχύτητα της διεπαφής), και στη συνέχεια δρομολογούνται σε ένα εσωτερικό component που τα επεξεργάζεται (EPROC_IN). Η φύση της επεξεργασίας αυτής, εξαρτάται από το πρότυπο με το οποίο υλοποιείται το e-link. Εάν δεν γίνεται καμία κωδικοποίηση,

τότε τα δεδομένα περνάνε ως έχουν μέσα από το EPROC_IN. Αν υλοποιείται κωδικοποίηση 8b/10b, τότε το υποσύστημα αυτό προβαίνει σε αποκωδικοποίηση των 10-bit χαρακτήρων που δέχεται. Στο 8b/10b mode, το EPROC_IN επίσης μπορεί και αναγνωρίζει τα comma characters, κάνοντας έτσι τον συγχρονισμό στη γραμμή πιο εύκολη υπόθεση. Μόλις τα δεδομένα υποστούν επεξεργασία, τότε δρομολογούνται σε μία μνήμη FIFO πλάτους 16-bit και βάθους 1 kb για προσωρινή αποθήκευση. Η έξοδος της FIFO είναι ένα σήμα μήκους δύο byte. Για να διαβάσει ο χρήστης τα καταχωρημένα δεδομένα από τη FIFO, πρέπει να ορίσει το λογικό ένα στην κατάλληλη είσοδο της FIFO (efifoRE ή αλλιώς RD_ENA, βλ. 6.1.2).

Έτσι γίνεται πλέον σαφής ο ρόλος του ELINK2FIFO: είναι ουσιαστικά ένας Deserializer. Αφού προβεί σε δειγματοληψία (sampling) των σειριακών δεδομένων του e-link, τα αποθηκεύει ως διανύσματα σε μία FIFO, την οποία μπορεί να διαβάσει ο χρήστης κατά τη βούλησή του. Αντίστοιχα, το FIFO2ELINK, προβαίνει στην αντίστροφη διαδικασία, και χρησιμοποιείται από έναν κόμβο του e-link όταν αυτός θέλει να αποστείλει σειριακά δεδομένα στον άλλο κόμβο. Είναι δηλαδή ένας Serializer. Διαθέτει μία μνήμη FIFO πλάτους 18-bit και βάθους 2 kb. Ένας εξωτερικός μηχανισμός, μπορεί και διοχετεύει τη FIFO αυτή με δεδομένα με μήκος 16 bit, τα οποία είναι τα δεδομένα προς αποστολή. Τα υπόλοιπα 2 bit είναι ένας κωδικός δύο λογικών ψηφίων που υποδεικνύουν πιο κομμάτι των δεδομένων είναι αυτό που συνοδεύει⁶. Τα πρώτα 2 byte του πακέτου που πρέπει να σταλεί μέσω e-link δηλαδή θα έχουν το πρόθεμα "10", τα υπόλοιπα το πρόθεμα "00" και τα τελευταία 2 byte θα έχουν το "01", όπου τότε το FIFO2ELINK θα ξέρει ότι το πακέτο προς αποστολή τελείωσε. Όταν η διεπαφή εφαρμόζει 8b/10b encdec⁷, τότε το FIFO2ELINK, κατά τα πρότυπα του 8b/10b, πριν ξεκινήσει να στέλνει την αρχή του πακέτου δεδομένων στη διεπαφή e-link, δρομολογεί πολλά comma characters στη γραμμή, προκειμένου να επέλθει συγχρονισμός μεταξύ πομπού και δέκτη. Επίσης επισυνάπτει μετά το τέλος του πακέτου και άλλους ειδικούς χαρακτήρες, που σηματοδοτούν στον δέκτη τη λήξη του. Έτσι υλοποιείται η επικοινωνία μεταξύ του FIFO2ELINK και ELINK2FIFO, δηλαδή η διεπαφή μεταξύ των δύο κόμβων του e-link.

⁶"00"=Data, "01"=End of Packet, "10"=Start of Packet

⁷8b/10b Encoding and Decoding. Η διαδικασία αυτή τελείται από ένα υποσύστημα μέσα στα FIFO2ELINK και ELINK2FIFO, ο κώδικας του οποίου έχει πηγή το Opencores.org[31]



Σχήμα 7.3.II: RTL Schematic του I1ddc_module από το firmware που υλοποιεί τη προσομοίωση του e-link. Στο κέντρο διακρίνεται το MMCM το οποίο λαμβάνει ένα εξωτερικό ρολόι. Αυτό το ρολόι μεταδίδεται και προς τα πάνω στο fe_module(δεν απεικονίζεται στη φωτογραφία) ως διαφορικό σήμα. Επίσης διακρίνεται το ELINK2FIFO, το οποίο δέχεται σαν είσοδο το διαφορικό σήμα των e-link.

Ολόκληρο αυτό το σύστημα που υλοποιεί τη διεπαφή και των δύο κόμβων του e-link μέσα στο ίδιο FPGA, προσομοιάστηκε στο ISim με επιτυχία. Στην επόμενη ενότητα, θα μελετηθούν δύο κώδικες οι οποίοι αναπτύχθηκαν για το debugging της διασύνδεσης του e-link. Ο ένας από τους δύο, χρησιμοποιεί το firmware που αναφέρθηκε σε αυτό το κομμάτι. Περισσότερες λεπτομέρειες λοιπόν για το πως ακριβώς λειτουργεί το firmware αυτό, θα παρατεθούν στη συνέχεια, όπου παρουσιάζονται οι κώδικες του debugging.

7.4 Debugging Firmware

Γενικά ως debugging μπορεί να οριστεί οποιαδήποτε διαδικασία που περιλαμβάνει την εύρεση και επίλυση σφαλμάτων μέσα σε έναν κώδικα. Η διαδικασία του

debugging, είναι πολύ σημαντική για οποιοδήποτε project, είτε αυτό περιλαμβάνει software, είτε firmware. Στην περίπτωση του firmware, το debugging μπορεί να γίνει μία πολύ επίπονη διαδικασία, καθώς σε πολλές περιπτώσεις που ένας κώδικας για FPGA δε δείχνει να ανταποκρίνεται, το σφάλμα μπορεί να έγκειται και σε παράγοντες που αφορούν τα components της πλακέτας, και όχι στον κώδικα αυτόν καθ' αυτόν. Το πρώτο πρωτότυπο της κάρτας L1DDC, απαιτεί εκτεταμένους ελέγχους σε όλες τις δυνατές λειτουργίες της, αφού άλλωστε, τα πρωτότυπα κατασκευάζονται για αυτόν ακριβώς το σκοπό. Σε αυτό το κομμάτι, το FPGA της L1DDC έπαιξε καθοριστικό ρόλο. Για το debugging εν προκειμένω, χρησιμοποιήθηκαν συμβατικές συσκευές, όπως πολύμετρα και παλμογράφοι. Επίσης, έγινε εκτενής χρήση των Chipscope ILA/ICON, σε περιπτώσεις ελέγχου των εσωτερικών σημάτων σε κάποιον κώδικα του FPGA. Οι περισσότεροι έλεγχοι που πραγματοποιήθηκαν, αφορούσαν τα e-link.

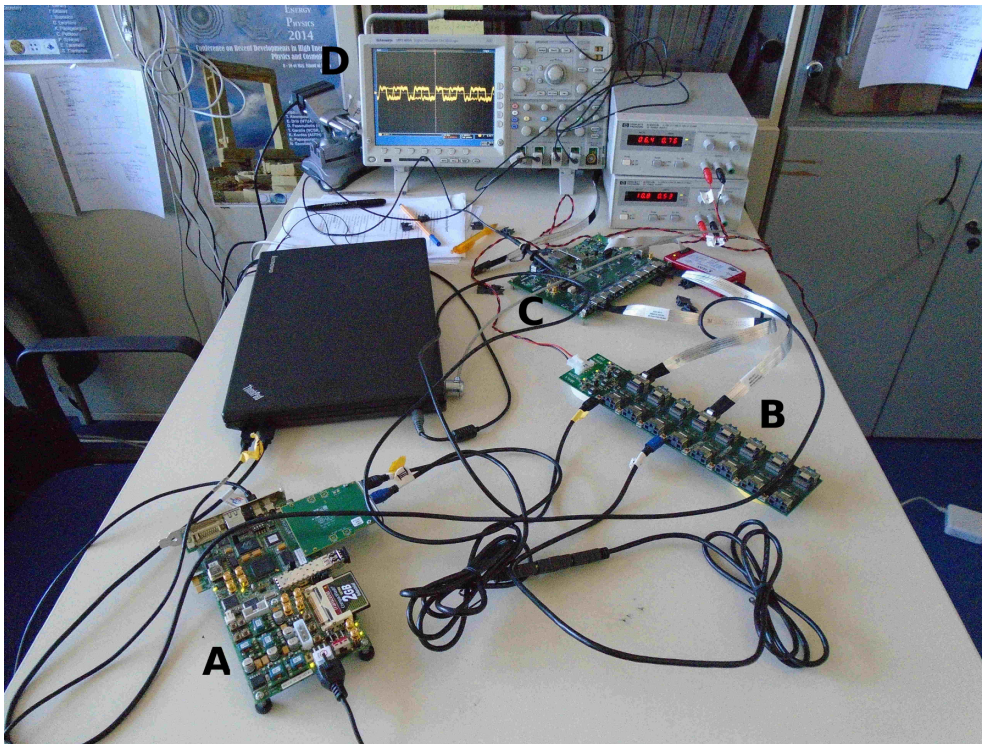
7.4.1 Απλές Διαδικασίες Ελέγχου των E-link

Πριν την ανάπτυξη του firmware για τα e-link, είχε προκύψει η ανάγκη για υλοποίηση firmware που θα ήλεγχε όλες τις πιθανές διασυνδέσεις της L1DDC με τις front-end boards. Όπως είναι ήδη γνωστό από την ενότητα 5.2, μαζί με το πρώτο πρωτότυπο της L1DDC, κατασκευάστηκαν και μερικές άλλες βοηθητικές πλακέτες. Η BBALA board, λειτουργεί σαν adapter board (κάτι σαν γέφυρα), ανάμεσα από την L1DDC και τις πρώτες front-end board που κατασκευάστηκαν από το BNL, οι οποίες είναι οι mini2. Οι mini2 όμως, δεν διαθέτουν miniSAS connectors, αλλά uHDMI. Για το λόγο αυτό η BBALA έχει στο ένα άκρο της θύρες uHDMI για να συνδέεται με τις mini2, και στο άλλο miniSAS για να συνδέεται με την αντίστοιχη θύρα της L1DDC. Έτσι υλοποιείται η διασύνδεση μεταξύ mini2 και L1DDC (βλ. και 5.2.1).

Προκειμένου να ελεγχθεί η ορθή λειτουργία της διασύνδεσης που μόλις περιγράφηκε, η οποία ουσιαστικά είναι μία διεπαφή e-link, πραγματοποιήθηκε μία σειρά από ελέγχους, κατά τους οποίους χρησιμοποιήθηκαν οι προαναφερθείσες πλακέτες, σε συνδυασμό με μία Xilinx® Spartan SP605 Evaluation Board, και έναν παλμογράφο Tektronix DPO 4504 [10]. Η SP605 διαθέτει ένα FPGA παλαιότερης τεχνολογίας (Spartan-6), το οποίο προγραμματίστηκε αναλόγως για τις ανάγκες των διαδικασιών του debugging. Ο λόγος χρήσης της SP605, έγκειται στο γεγονός ότι η πλακέτα αυτή διαθέτει μία υποδοχή για FMC adapter, ο οποίος μπορεί και λαμβάνει τα σή-

ματα από τα FPGA και τα δρομολογεί σε θύρες uHDMI. Έτσι η SP605 προσομοιάζει μία mini2.

Όλες οι διαδικασίες ελέγχου, περιελάμβαναν κατά κύριο λόγο την αποστολή σειριακών δεδομένων και σημάτων χρονισμού από την SP605 στην L1DDC και αντίστροφα, μέσω του προτύπου LVDS. Η διασύνδεση αυτή απεικονίζεται στο ακόλουθο Σχήμα (7.4.I):



Σχήμα 7.4.I: Σχηματική αναπαράσταση της συνδεσμολογίας μεταξύ SP605, BBALA και L1DDC, η οποία προσομοιάζει τα e-link. **A:** SP605 με τον FMC adapter, **B:** BBALA, **C:** L1DDC, **D:** Παλμογράφος.

Στο πρώτο βήμα ελέγχου, η SP605 προγραμματίστηκε σαν απλός Serializer, σύμφωνα με τον κώδικα που έχει παρατεθεί στο προηγούμενο Κεφάλαιο (βλ. ενότητα Β΄.3). Το σειριακό σήμα μεταδιδόταν έξω από την SP605, μέσω του FMC adapter. Το καλώδιο uHDMI που ήταν συνδεδεμένο πάνω στις υποδοχές της FMC, συνδεόταν από την άλλη του άκρη με τον αντίστοιχο κοννέκτορα στη BBALA. Ένα καλώδιο miniSAS από την άλλη μεριά της BBALA μετέφερε το σήμα από το uHDMI

στον miniSAS connector της L1DDC. Η όλη διασύνδεση έγινε ακολουθώντας το πρότυπο LVDS. Η L1DDC, λάμβανε τα διαφορικά αυτά σήματα από τον connector, τα μετέτρεπε σε μονοπολικό σήμα στο εσωτερικό του, και τα δρομολογούσε στα pins που βρίσκονταν πάνω στην πλακέτα. Τελικά, ο παλμογράφος συνδεόταν με το pin, και απεικόνιζε την κυματομορφή που προέκυπτε μετά από την όλη διαδικασία μετάδοσης/λήψης. Μεταδίδοντας ένα συγκεκριμένο σήμα από την SP605 (π.χ. serialization ενός byte "10010110"), ο χρήστης αναμένει να διαβάσει με τον παλμογράφο την ψηφιακή κυματομορφή που αντιστοιχεί στη σειριακή μετάδοση αυτού του διανύσματος. Αξίζει επίσης να σημειωθεί ότι το serialization (άρα και η μετάδοση της πληροφορίας), έγινε από την SP605 με τέσσερις διαφορετικές συχνότητες (40, 80, 160, 320 *Mhz*), κατ' αναλογία με τις διαθέσιμες ταχύτητες μετάδοσης/λήψης στα e-link. Η εφαρμογή διαφορετικών συχνοτήτων μετάδοσης, επέτρεψε την λεπτομερέστερη εξέταση του μηχανισμού μετάδοσης και λήψης δεδομένων από τα καλώδια και τις διασυνδέσεις, που λάμβαναν μέρος στην διάδοση των ψηφιακών παλμών, για μία ευρεία γκάμα συχνοτήτων [10].

Η διαδικασία που μόλις περιγράφηκε εκτελέστηκε για όλες τις κάρτες L1DDC και BBALA. Ελέγχθηκαν όλες οι δυνατές πορείες που μπορούσαν να ακολουθήσουν τα σήματα. Μέσω αυτού του ελέγχου μάλιστα, βρέθηκε ένα σφάλμα στις δρομολογήσεις των σημάτων μέσα στη BBALA, το οποίο και διορθώθηκε στις μετέπειτα εκδόσεις της πλακέτας. Κατά τα άλλα, κανένα άλλο σοβαρό πρόβλημα δεν ανέκυψε. Πέρα από την μεθοδολογία που μόλις περιγράφηκε, χρησιμοποιήθηκε και μία άλλη λιγότερο περίπλοκη, όπου το Serialization γινόταν από την ίδια την L1DDC. Το σήμα αυτό μεταδιδόταν με διαφορεική μορφή μέσω του miniSAS, το οποίο μετά κατέληγε στη BBALA. Από εκεί, ένα άλλο καλώδιο miniSAS, λάμβανε το σήμα από τη BBALA και το επέστρεφε σε έναν διαφορετικό miniSAS connector της L1DDC, το FPGA της οποίας μετέτρεπε το ληφθέν σήμα σε μονοπολικό, και το δρομολογούσε πάλι πίσω στο pin όπου το διάβαζε ο παλμογράφος.

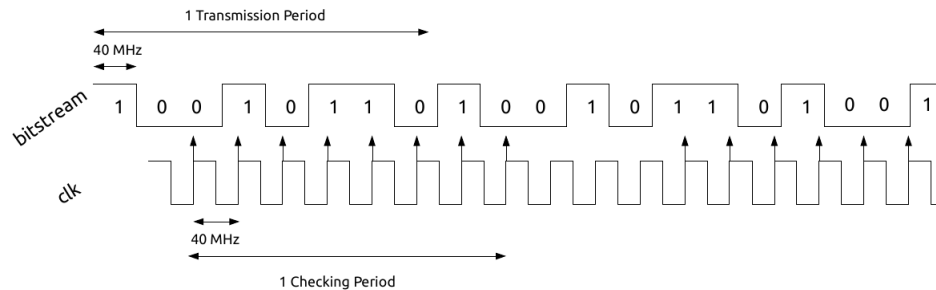
Όπως έχει γίνει σαφές, όλες οι παραπάνω διαδικασίες ελέγχου απαιτούσαν τη χρήση παλμογράφου για την επιθεώρηση των κυματομορφών. Η συμπερίληψη του παλμογράφου εισήγαγε αναγκαστικά ένα συστηματικό σφάλμα στην πειραματική διαδικασία, το οποίο έγινε εμφανές μετά από την δυσκολία στην αναγνώριση των κυματομορφών όταν η ροή των δεδομένων ήταν πολύ μεγάλη. Συνοψίζοντας, κρίθηκε απαραίτητο να ελεγχθεί το πιο σημαντικό απ' όλα: Εάν δηλαδή το FPGA της L1DDC

μπορούσε να διακρίνει το λογικό ένα από το λογικό μηδέν κατά την ανάγνωση των δεδομένων που λάμβανε απ' έξω, μέσω του miniSAS, για ένα ικανοποιητικό εύρος συχνοτήτων. Ένα τεστ με αυτή τη φιλοσοφία, θα καθόριζε επίσης με ακόμα πιο ακριβή τρόπο την ακεραιότητα των διασυνδέσεων μεταξύ των δύο πλακετών, καθώς εάν σε ένα σημείο κάποιο σφάλμα προσέδιδε ανεπιθύμητα χαρακτηριστικά στις κυματομορφές (π.χ. παραμόρφωση τάσης λόγω θορύβου), τότε η L1DDC δεν θα αναγνώριζε τα δεδομένα σωστά.

Η μέθοδος που θα περιγραφεί στη συνέχεια, επιτρέπει τον πλήρη έλεγχο της διαδικασίας λήψης σειριακών δεδομένων από την L1DDC, μέσω των θυρών miniSAS. Η διαδικασία χωρίζεται σε περιόδους ελέγχου, όπου γίνονται συνολικά 100 επαναλήψεις. Σε κάθε επανάληψη, η L1DDC ελέγχει τα δεδομένα που έλαβε, και αν αυτά συμφωνούν με τα αναμενόμενα τότε προχωράει στην επόμενη περίοδο ελέγχου. Εάν όμως υπήρξε σφάλμα, ένας εσωτερικός μετρητής αυξάνεται κατά ένα, πριν περάσει στην επόμενη περίοδο ελέγχου. Μετά από τις 100 επαναλήψεις, η L1DDC επιστρέφει στο χρήστη το πλήθος των σφαλμάτων (την τιμή του counter δηλαδή). Πιο συγκεκριμένα, αρχικά ορίζεται ένα byte προς μετάδοση στην SP605, και μία ταχύτητα μετάδοσης. Η SP605 θα μεταδίδει συνεχώς αυτό και μόνο αυτό το byte σειριακά στην L1DDC. Έστω ότι αυτό το byte είναι το "10010110"⁸. Χάρη απλότητας της μεθοδολογίας, θεωρήθηκε περιττό να τεσταριστεί η ικανότητα CDR από την L1DDC, καθώς το ζητούμενο ήταν να ελεγχθεί η τυχόν ύπαρξη σημαντικού εγγενούς θορύβου στις διασυνδέσεις μεταξύ των πλακετών, και όχι η συμφωνία ως προς τη φάση των δεδομένων μεταξύ πομπού και δέκτη. Έτσι, επινοήθηκε ένας απλός μηχανισμός ελέγχου του bitstream από την L1DDC. Με ένα εσωτερικό ρολόι, ίδιας συχνότητας με του ρολογιού μετάδοσης, η L1DDC προβαίνει σε sampling του bitstream. Σε οκτώ κύκλους, κατασκευάζει ένα διάνυσμα από οκτώ bit, το οποίο και ελέγχει. Από τη στιγμή που το bitstream είναι συνεχές και ίδιο, τότε η L1DDC μπορεί να ξεκινήσει το serialization από οποιοδήποτε σημείο του bitstream. Επομένως, μπορεί να είναι τυχερή και να αποθηκεύσει το "10010110", ή μπορεί να ξεκινήσει το deserialization μία θέση μετά και να αποθηκεύσει το "00101101", ή το "01011010", κ.ο.κ. Έτσι, για το αρχικό και γνωστό byte που μεταδίδει η SP605 ("10010110"), δημιουργούνται σύμφωνα με αυτό το μοτίβο κυκλικής εναλλαγής των bit, οκτώ byte που θα κρίνονται αποδεκτά από την L1DDC. Αν κατά τη διάρκεια της μετάδοσης αλλάξει έστω

⁸Άρα το bitstream θα είναι "10010110100101101001..."

και ένα bit, τότε αν το αποθηκεύσει η L1DDC, θα αναγνωρίσει σφάλμα αφού δεν θα ανήκει στις οκτώ πιθανές σωστές απαντήσεις. Η όλη διαδικασία περιγράφεται σχηματικά στο 7.4.Π.



Σχήμα 7.4.Π: Το sampling του bitstream μπορεί να γίνει από πολλά διαφορετικά σημεία. Παρ' όλα αυτά, αν δεν έχει αλλάξει κάποιο bit κατά τη μετάδοση, από όποιο σημείο και να ξεκινήσει το sampling, δεν θα καταχωρηθεί σφάλμα.

Μετά το πέρας 100 επαναλήψεων, η L1DDC κατασκευάζει έναν βηματικό παλμό, σαν ρολόι με μεγάλη περίοδο (περίπου ένα δευτερόλεπτο), τον οποίο και μεταδίδει πίσω στην SP605, η οποία δρομολογεί τους παλμούς σε ένα LED. Η κυματομορφή είναι φτιαγμένη με τέτοιο τρόπο ώστε το LED να αναβοσβήνει τόσες φορές όσο και το πλήθος των σφαλμάτων. Επίσης, το firmware διαθέτει και global reset μέσω ενός διακόπτη πάνω στην SP605. Το reset αυτό επιτρέπει στον χρήστη να εναλλάσσει γρήγορα και εύκολα τις διασυνδέσεις πάνω στη BBALA, ώστε να εφαρμόζει τη μέθοδο ελέγχου σε όλους τους BBALA connectors. Τελικά, εξαιρώντας τα λάθος δρομολογημένα κανάλια που είχαν ήδη βρεθεί από πριν πάνω στη BBALA, διαπιστώθηκε η αρτιότητα της BBALA και της L1DDC στο να λαμβάνουν σήματα, αφού δε σημειώθηκαν σφάλματα για όλες τις συχνότητες μετάδοσης [10]. Ακολουθεί το κομμάτι του κώδικα που προβαίνει στον έλεγχο των δεδομένων. Στις πρώτες γραμμές γίνεται ο ορισμός των οκτώ δυνατών διανυσμάτων που μπορεί να θεωρηθούν σωστά. Ο check_counter αυξάνει κατά ένα με κάθε επανάληψη ελέγχου. Στις γραμμές 20-28 γίνεται το deserialization. Στις γραμμές 34-50 γίνεται ο έλεγχος του διανύσματος με τις πιθανά σωστά διανύσματα. Αν γίνουν εκατό επαναλήψεις, ενεργοποιείται ένα σήμα που σταματάει τη διαδικασία ελέγχου (58-60). Η τιμή του error_counter καθορίζει τον παλμό που θα σταλεί στο LED της SP605 για να διαβάσει ο χρήστης.

VHDL Code 7.2: bitstream_checker

```
1 _____
2 _____
3 generic (len : natural := 7;
4           periods : natural := 100;
5           a : std_logic_vector := "10010110";
6           b : std_logic_vector := "00101101";
7           c : std_logic_vector := "01011010";
8           d : std_logic_vector := "10110100";
9           e : std_logic_vector := "01101001";
10          f : std_logic_vector := "11010010";
11          g : std_logic_vector := "10100101";
12          h : std_logic_vector := "01001011");
13 _____
14 _____
15 _____
16 if (rising_edge(clk)) then — clk
17
18     if (check_counter < periods) then — 1
19
20         if (index <= len) then — 2
21
22             case bitstream is — case check bitstream
23
24                 when '1' => test_vector(index) := '1'; index:=index+1;
25                 when '0' => test_vector(index) := '0'; index:=index+1;
26                 when others => test_vector(index) := 'U'; index:=index+1;
27
28             end case; — case check bitstream
29
30             elsif (index > len) then
31
32                 index := 0;
33
34                 if ((test_vector = a) or — compare begin
35                     (test_vector = b) or
36                     (test_vector = c) or
37                     (test_vector = d) or
38                     (test_vector = e) or
39                     (test_vector = f) or
40                     (test_vector = g) or
```

```
41         (test_vector = h)) then
42
43         test_vector := (others => 'U');
44         check_counter := check_counter + 1;
45
46         else
47
48         error_counter := error_counter + 1;
49         test_vector := (others => 'U');
50         check_counter := check_counter + 1;
51
52         end if; -- compare end
53
54         else null;
55
56         end if; -- 2
57
58     elsif (check_counter = periods) then
59
60         check_stopper := '1';
61
62         else null;
63
64         end if; -- 1
65
66     end if; -- clk
```

7.4.2 Σύνθετες Διαδικασίες Ελέγχου των E-link

Παρά την πρακτικότητα και απλότητα της προαναφερθείσας μεθόδου, δυστυχώς δεν μπορεί να θεωρηθεί πλήρως επαρκής ώστε να καθορίσει την ποιότητα των e-link μεταξύ L1DDC και front-end boards. Για το λόγο αυτό, μόλις αναπτύχθηκε το firmware που υλοποιεί τα e-link (το οποίο αναλύθηκε σε προηγούμενη ενότητα), δημιουργήθηκε μία πλήρης διαδικασία ελέγχου των e-link, κάνοντας χρήση των components του τελικού e-link firmware. Η λογική αυτής της μεθοδολογίας, ελέγχθηκε κατ' αρχάς σε συνθήκες προσομοίωσης μέσω του ISim, καθώς εφαρμόστηκε στον κώδικα elink_complete_sim ο οποίος υλοποιεί και τους δύο κόμβους των e-link στο ίδιο FPGA. Σε κάθε περίπτωση, ο αναγνώστης παραπέμπεται στην ενότητα 7.3

για να ανανεώσει τη μνήμη του περί του τρόπου λειτουργίας του firmware. Σε αυτή τη διαδικασία ελέγχου, χρησιμοποιείται η κωδικοποίηση 8b/10b αφού επιτρέπει εύκολο clock recovery από τους δύο κόμβους του e-link.

Στην περίπτωση αποστολής δεδομένων μέσω e-link, απαιτείται η χρήση της FIFO2ELINK. Ο χρήστης πρέπει να δρομολογήσει το πακέτο που θέλει να στείλει στη FIFO που βρίσκεται μέσα στο component, ώστε αυτό να το προβεί σε κωδικοποίηση 8b/10b, πριν γίνει το serialization και η αποστολή στο e-link. Η λήψη γίνεται από το ELINK2FIFO στην άλλη μεριά του e-link, όπου πραγματώνεται η δειγματοληψία των δεδομένων, μετά το deserialization, και η αποθήκευση σε μία FIFO. Έτσι λοιπόν, είναι πλέον σαφές το πως θα υλοποιηθεί η διαδικασία ελέγχου: Θα πρέπει ουσιαστικά τα περιεχόμενα και των δύο FIFO να ταυτίζονται. Δηλαδή τα δεδομένα να περνάνε ακέραια από τον αποστολέα στον παραλήπτη. Για το λόγο αυτό, δημιουργήθηκε μία FSM (packet_sending_FSM) η οποία συνθέτει πακέτα δεδομένων και τα δίνει στη FIFO του component FIFO2ELINK, ώστε μετά αυτά τα δεδομένα να μεταδοθούν σειριακά μέσω του e-link. Επίσης, δημιουργήθηκε και μία άλλη FSM (packet_checking_FSM), η οποία ελέγχει τα περιεχόμενα της FIFO του δέκτη (ELINK2FIFO), και αν αυτά ταυτίζονται με τα πακέτα που έστειλε η packet_sending_FSM, τότε το e-link θα λειτουργεί σωστά.

Σε αυτό το σημείο αξίζει να σημειωθεί ότι το ELINK2FIFO, έχει τη δυνατότητα να διακρίνει την αρχή και το τέλος των πακέτων που διαβάζει από το e-link. Υπενθυμίζεται ότι στη μεριά του αποστολέα (FIFO2ELINK), ο χρήστης πρέπει να προσθέσει σε κάθε 16-bit διάνυσμα που θέλει να στείλει μέσω e-link και τα δύο bit που υποδεικνύουν το είδος των δεδομένων (αρχή, μέση, τέλος πακέτου). Ο δέκτης (ELINK2FIFO), έχει τη δυνατότητα αναγνώρισης αυτών των κομματιών, αφού τα αποκωδικοποιήσει, και αυτό που κάνει είναι να παίρνει ένα ολόκληρο πακέτο, από την αρχή του μέχρι το τέλος του, και να κατασκευάζει "γύρω του" ένα δικό του πακέτο, με custom Header και Trailer. Ο Header αποτελείται κυρίως από ένα 16-bit χαρακτήρα ABCD σε δεκαεξαδικό. Ο Trailer αποτελείται από επίσης 16 bit, και μέσα του έχει διάφορα flags για το αν εντόπισε κάποιο λάθος, ενώ 10 bit του κωδικοποιούν το μέγεθος του πακέτου⁹. Αξίζει επίσης να σημειωθεί ότι όταν το FIFO2ELINK βρί-

⁹Αυτό το πακέτο ως έχει, αν αναφέρεται κανείς στην L1DDC, δρομολογείται στο UDP, για να καταλήξει τελικά μέσω Ethernet να διαβαστεί από τον υπολογιστή. Και έτσι υλοποιείται πλέον ολόκληρη η διασύνδεση μεταξύ MMFE8, L1DDC και back-end.

σκεται σε αναμονή και δεν δέχεται δεδομένα, συνεχίζει και στέλνει idle χαρακτήρες στη γραμμή του e-link ώστε να μη χάνεται ο συγχρονισμός.

Οι FSM που έχουν δημιουργηθεί για το σκοπό αποστολής δεδομένων και ελέγχου τους, διαχειρίζονται από μία κεντρική FSM (centralFSM). Επομένως κάθε κόμβος, διαθέτει τρεις FSM: Η μία κατασκευάζει πακέτα, η άλλη ελέγχει πακέτα, και η άλλη διαχειρίζεται τις FSM και τα υπόλοιπα components, δίνοντας τις κατάλληλες εντολές τις κατάλληλες στιγμές ώστε να είναι καλά συγχρονισμένο ολόκληρο το σύστημα. Επίσης, σε κάθε κόμβο, υπάρχει ένας πολυπλέκτης, η μία είσοδος του οποίου είναι συνδεδεμένη με την έξοδο του FIFO2ELINK (του serializer), και η άλλη είναι γειωμένη. Η έξοδος του, καταλήγει σε έναν buffer εξόδου που καταλήγει σε διαφορεική έξοδο. Αυτό είναι το e-link που μεταδίδει σειριακά δεδομένα. Το SEL, έρχεται από την κεντρική FSM, έτσι ώστε να υπάρχει πλήρης έλεγχος στη μετάδοση των πληροφοριών. Σε αυτό το τεστ, ελέγχονται δύο διαφορεικά e-link path, όπου στο ένα η L1DDC μεταδίδει δεδομένα και στο άλλο λαμβάνει, ενώ ελέγχεται και η ορθότητα μετάδοσης σημάτων χρονισμού, αφού στέλνεται ένα ρολόι από την L1DDC στην front-end. Μία σειρά idle χαρακτήρων και πακέτων θα ανταλλαχθούν μεταξύ των δύο πλακετών, προκειμένου να επιβεβαιωθεί ότι η σύνδεσή τους βαίνει καλώς. Η αλληλουχία των βημάτων λοιπόν, από την αρχή της διαδικασίας ελέγχου, μέχρι το τέλος, έχει ως εξής:

- Η L1DDC στέλνει το ρολόι στην Front-End (FE).
- Αν η FE λάβει το ρολόι, και κλειδώσει το MMCM της με αυτό το ρολόι, τότε το σήμα του κλειδώματος μπαίνει στην κεντρική FSM της FE, η οποία θα αλλάξει κατάσταση, και θα ανοίξει το δρόμο του FIFO2ELINK, ώστε να σταλούν idle πακέτα στη γραμμή του e-link. Αυτό το κάνει αλλάζοντας το SEL του πολυπλέκτη ο οποίος πριν ήταν γειωμένος.
- Η L1DDC πρέπει τώρα να αναγνωρίσει τους idle χαρακτήρες που ξεκίνησαν να στέλνονται, ώστε να επιβεβαιωθεί ότι υπάρχει συγχρονισμός. Αυτό γίνεται από την FSM που ελέγχει τα πακέτα (packet_checking_FSM), η οποία είναι συνδεδεμένη με την FIFO του component FIFO2ELINK. Αν αυτή δει ότι αποθηκεύτηκαν idle χαρακτήρες στη FIFO, τότε η κεντρική FSM της L1DDC ενημερώνεται για αυτό, και προβαίνει στο αντίστοιχο άνοιγμα του e-link, από τη δική της μεριά.

- Αναλόγως, η FSM που ελέγχει τα πακέτα στη FE, περιμένει να λάβει τους idle χαρακτήρες που ξεκίνησαν να έρχονται από την L1DDC. Αν αναγνωρίσει τα idle και αυτή, σημαίνει πως πλέον έχει εδραιωθεί πλήρης επικοινωνία μεταξύ L1DDC και FE, τόσο ως προς τα δύο ξεχωριστά e-link, όσο και ως προς το ρολόι που μεταδίδεται.
- Όταν λάβει τους idle και η FE, τότε η κεντρική FSM, ενεργοποιεί την FSM που κατασκευάζει πακέτα (packet_sending_FSM), η οποία δρομολογεί ένα πακέτο με μήκος 39 byte στη FIFO του component FIFO2ELINK. Το πακέτο στέλνεται στο e-link.
- Η FSM της L1DDC που ελέγχει τα πακέτα που λαμβάνονται, αναγνωρίζει ότι ξεκίνησε η αποκωδικοποίηση ενός πακέτου, και διαβάζει τα περιεχόμενά του και το trailer του (υπενθυμίζεται ότι το component ELINK2FIFO δεν αποκωδικοποιεί μόνο τα δεδομένα, αλλά τα βάζει και σε ένα ειδικό πακέτο που κατασκευάζει η ίδια). Αν τα περιεχόμενα ταυτίζονται με αυτά που έστειλε η FE, και αν ο trailer επίσης είναι σωστός, τότε ενημερώνεται η κεντρική FSM ότι ελήφθη το σωστό πακέτο.
- Η κεντρική FSM της L1DDC, γνωρίζοντας ότι έχει ληφθεί το κύριο πακέτο σωστά, δίνει εντολή στην FSM που στέλνει δεδομένα να αποστείλει ένα μικρό πακέτο ACK (acknowledgement), πίσω στην FE.
- Αν το ACK ληφθεί σωστά από την FE (ο έλεγχος είναι ανάλογος με αυτόν που έγινε προηγουμένως από την L1DDC), τότε η κεντρική FSM της FE σηματοδοτεί το τέλος της διαδικασίας, η οποία αν φτάσει μέχρι εδώ, κρίνεται επιτυχής.

Σε κάθε περίπτωση, μπορεί να υπάρξουν σφάλματα σε κάθε βήμα αυτής της διαδικασίας. Για το λόγο αυτό, κάθε κεντρική FSM διαθέτει εσωτερικούς μετρητές. Όταν η FSM παραμένει σε μία κατάσταση, ένας μετρητής αυξάνεται σε κάθε χτύπο του ρολογιού. Αν η FSM περάσει σε επόμενη κατάσταση, τότε ο μετρητής μηδενίζεται. Αν όμως δεν λάβει τα κατάλληλα σήματα, θα παραμείνει στην ήδη υπάρχουσα και ο μετρητής θα συνεχίσει να αυξάνει. Επομένως, αν κάποιος counter ξεπεράσει τα όριά του (τα οποία έχουν οριστεί περίπου στα δέκα δευτερόλεπτα για την πραγματική

υλοποίηση του firmware), σημαίνει ότι κάτι δεν πήγε καλά στην όλη διαδικασία¹⁰ και η FSM πέφτει σε error state. Υπάρχει και διαφορετικό error state για κάθε διαφορετικό τύπο σφάλματος. Αυτό σημαίνει ότι ο χρήστης θα δει διαφορετικό σήμα που σημαίνει σφάλμα, αν για παράδειγμα η FE δεν έλαβε ποτέ το ρολόι (δηλαδή δεν κλείδωσε ποτέ το MMCM της), ή αν η L1DDC δεν έλαβε ποτέ το κυρίως πακέτο. Τα error σήματα, δρομολογούνται σε ξεχωριστά LED ή σε pins, και αν κάνει προσεκτική αντιστοίχιση, ο χρήστης μπορεί εύκολα να διαγνώσει τυχόν προβλήματα που υπάρχουν σε όλες τις γραμμές των e-link, με τη βοήθεια αυτού του πλήρους και αξιόπιστου πακέτου ελέγχου των e-link. Αν από την άλλη η διαδικασία φτάσει μέχρι το τέλος χωρίς προβλήματα, τότε αντίστοιχα σήματα στα LED και pins θα ενημερώσουν τον χρήστη ότι η διαδικασία ολοκληρώθηκε με επιτυχία.

Το firmware αυτό, έχει ελεγχθεί τόσο σε συνθήκες προσομοίωσης μέσω ISim, τόσο και σε πραγματική διασύνδεση SP605 ↔ BBALA ↔ L1DDC. Έχει επιβεβαιωθεί ότι λειτουργεί σωστά, και μπορεί να ελέγξει όλους τους τύπους σφαλμάτων που μπορεί να εμφανιστούν στα e-link. Η εφαρμογή του κώδικα έχει επίσης καταδείξει κατασκευαστικά σφάλματα σε ορισμένους miniSAS και uHDMI connectors της BBALA. Κομμάτια του firmware και block diagrams, μπορούν να βρεθούν στο Παράρτημα (κομμάτια κώδικα για την Κεντρική FSM, FSM της L1DDC που στέλνει ACK, FSM της FE που διαβάζει τα σήματα και περιμένει το ACK).

¹⁰π.χ. η FSM που ελέγχει τα πακέτα περιμένει για πολύ ώρα να λάβει idle, αλλά αυτό δεν έρχεται ποτέ γιατί υπάρχει ίσως πρόβλημα στη γραμμή.

Πρωτόκολλα Επικοινωνίας

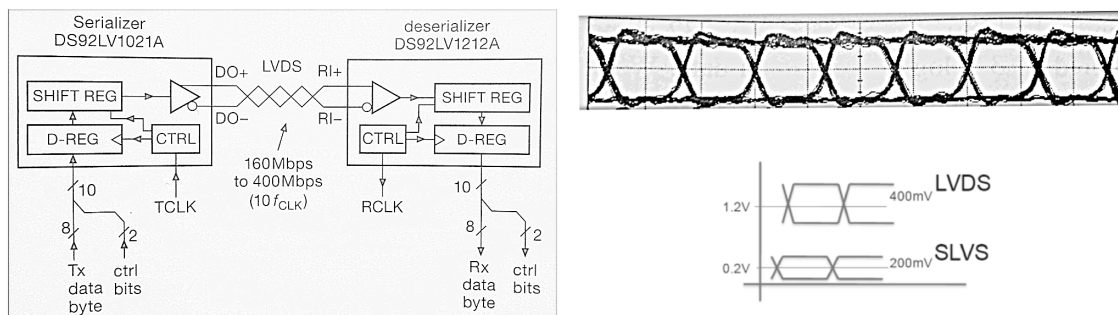
Σε αυτό το Παράρτημα, θα μελετηθούν τα πρωτόκολλα επικοινωνίας που θα εφαρμοστούν μεταξύ των καρτών του συστήματος ηλεκτρονικών του NSW. Εδώ, θα γίνουν αναφορές που έχουν περισσότερο σχέση με τη θεωρία πίσω από τα πρότυπα που θα υλοποιηθούν. Η πρακτική τους εφαρμογή, εξετάζεται στο Κεφάλαιο 7, καθώς δομούνται μέσω του firmware του FPGA της L1DDC.

Κατ' αρχάς, έχει αναφερθεί ότι τα e-link, επιστρατεύουν τα πρότυπα LVDS/SLVS για τη μετάδοση των πληροφοριών στις γραμμές τους. Επίσης, το FPGA, θα χρησιμοποιήσει τα πρωτόκολλα Ethernet και UDP για την υλοποίηση της επικοινωνίας της L1DDC με έναν υπολογιστή που θα προσομοιάζει τρόπον τινά το δίκτυο FELIX. Σημαντικό ρόλο επίσης θα παίξει και η κωδικοποίηση 8b/10b τόσο στο Ethernet, όσο στα e-link, όπως αυτά θα υλοποιηθούν από τα FPGA των L1DDC και MMFE8. Τέλος, θα αναφερθούν λεπτομέρειες για το πρωτόκολλο I²C, το οποίο θα εφαρμοστεί από το FPGA για να επικοινωνήσει με το GBTx πάνω στην L1DDC, και με τα Spartan-6 FPGA των καρτών mini2.

A'.1 Low-Voltage Differential Signaling

Το *Low-Voltage Differential Signaling (LVDS)*, είναι ένας τρόπος μετάδοσης δεδομένων από έναν πομπό σε κάποιο δέκτη, ο οποίος χρησιμοποιεί διαφορετικά σήματα. Το LVDS, κάνει χρήση διαφορικών ζευγών καλωδίων, τα οποία έχουν ένα κοινό επίπεδο αναφοράς ως προς την τάση. Η μία γραμμή είναι συμπληρωματική της άλλης, ενώ η διαφορά των μεταξύ τους τάσεων είναι σταθερή. Η τεχνική υλοποίησης του LVDS, περιλαμβάνει τη μετάδοση σταθερού ρεύματος από τον πομπό, το οποίο

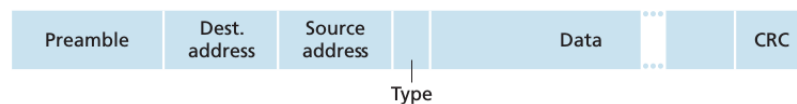
αφού περάσει από μία αντίσταση στον δέκτη (termination resistance), συνήθως $100\ \Omega$, επιστρέφει πίσω στον πομπό από την άλλη γραμμή. Έτσι δημιουργείται ένα ζεύγος κυματομορφών με αντίθετη πολικότητα και σταθερή διαφορά τάσης. Πιο συγκεκριμένα, η τάση αναφοράς μπορεί να είναι στα $1.2\ V$, και η μεταξύ τους διαφορά στα $400\ mV$, επομένως η μία γραμμή, θα αντιπροσωπεύει το λογικό ένα στα $1.4\ V$, και η άλλη στα $1.0\ V$ [18]. Επομένως ο δέκτης δεν μετράει ουσιαστικά την απόλυτη τάση, αλλά τη διαφορά τάσης που υπάρχει μεταξύ των διαφορικών ζευγών αντίθετης πολικότητας. Αυτός ο σχεδιασμός απορρίπτει τον κοινό ηλεκτρονικό θόρυβο, αφού αυτός επηρεάζει με όμοιο τρόπο και τα δύο σήματα, και δεν αλλάζει τη μεταξύ τους διαφορά ως προς την τάση. Επίσης, τα επίπεδα τάσης που υφίστανται στις γραμμές αυτές, είναι γενικά πολύ χαμηλά, γεγονός επιθυμητό, αν σκεφτεί κανείς ότι τα σύγχρονα ολοκληρωμένα κυκλώματα που υλοποιούν τέτοια πρωτόκολλα χρησιμοποιούν έτσι και αλλιώς χαμηλές τάσεις για να λειτουργήσουν. Το πρότυπο LVDS χρησιμοποιείται σε διασυνδέσεις μήκους μερικών μέτρων, όπου απαιτείται μεγάλη ροή δεδομένων. Έτσι εμφανίζεται σε διασυνδέσεις μεταξύ ολοκληρωμένων κυκλωμάτων στην ίδια πλακέτα (PCIe, SATA), και στην επικοινωνία μεταξύ καρτών χωρίς μεγάλη απόσταση μεταξύ τους (e-link εν προκειμένω). Τα e-link, υλοποιούν το πρότυπο LVDS, που έχει τάση αναφοράς $1.2\ V$ και διαφορά τάσης $\pm 400\ mV$, και το SLVS, με τάση αναφοράς στα $0.2\ V$, και διαφορά τάσης $\pm 200\ mV$ [10].



Σχήμα Α'.1.1: Αριστερά: Διάταξη ενός πομπού και δέκτη που επικοινωνούν μέσω LVDS. Ο πομπός (serializer), δέχεται bits σε παράλληλη μορφή, και τα μεταδίδει σειριακά μέσω ενός ρολογιού TCLK. Ο δέκτης, προβαίνει σε deserialization και αποθηκεύει τα bits από τη γραμμή σε έναν register σε παράλληλη μορφή. Παρόμοια διαδικασία SERDES λαμβάνει χώρα και στα e-link. Δεξιά-Πάνω: Τυπική κυματομορφή ενός ζεύγους LVDS. Δεξιά-Κάτω: Τα επίπεδα τάσης LVDS και SLVS [18, 10].

Α'.2 Ethernet - UDP Protocol

Το Ethernet είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται σε τοπικό (Local Area Networks (LAN)) ή ευρύτερο επίπεδο (Metropolitan Area Networks (MAN)). Εμφανίστηκε πρώτη φορά στα μέσα της δεκαετίας του '70 [19], και εκτόπισε γρήγορα τις άλλες τεχνολογίες που είχαν αναπτυχθεί για τους ίδιους σκοπούς, λόγω της απλότητάς του και των υψηλών ταχυτήτων του. Το Ethernet εμπίπτει στο δεύτερο επίπεδο του OSI model¹ το οποίο ονομάζεται *Data Link Layer*. Το Ethernet κατασκευάζει το αντίστοιχο frame, και το προωθεί στο τελευταίο και πιο βασικό επίπεδο, το *Physical Layer* ή *PHY*. Πιο πάνω από το Link Layer, υπάρχουν τα Transport και Network επίπεδα, τα οποία υλοποιούν τα πρωτόκολλα μεταφοράς και δικτύου αντίστοιχα². Το πρωτόκολλο μεταφοράς που έχει υλοποιηθεί στο FPGA της L1DDC είναι το *User Datagram Protocol (UDP)*. Περισσότερα για τον τρόπο εφαρμογής του Ethernet και UDP μπορούν να βρεθούν στο Κεφάλαιο 7. Ενδεικτικά, παρατίθεται η γενική δομή ενός Ethernet frame στο Σχήμα Α'.2.1:



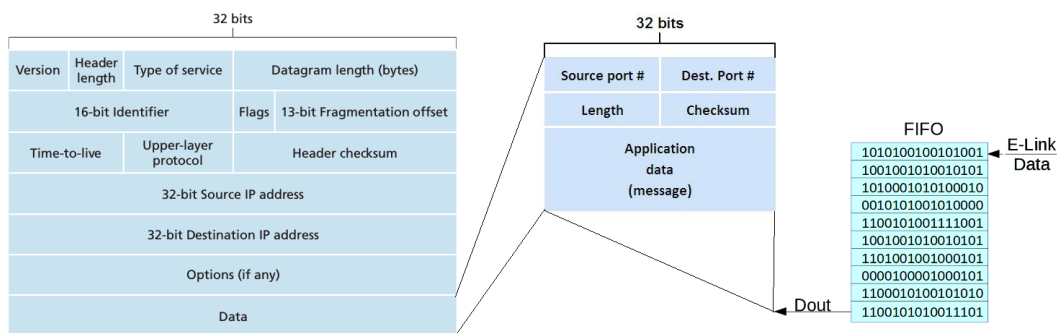
Σχήμα Α'.2.1: Δομή ενός Ethernet frame [19].

Η διεύθυνση του παραλήπτη (Dest. Address), είναι η διεύθυνση MAC (*MAC Address*) του κόμβου εκείνου στο LAN, για τον οποίο προορίζεται το πακέτο. Έχει μήκος έξι byte. Αντίστοιχα, η διεύθυνση αποστολέα (Source Address) είναι η διεύθυνση του κόμβου από τον οποίο προέρχεται το frame. Το πεδίο του Type, αναφέρει τον τύπο του πρωτοκόλλου δικτύου που χρησιμοποιεί το frame (π.χ. IPv4). Τα δεδομένα (Data), έχουν μέγεθος μέχρι και 1500 byte, και φέρουν το πακέτο που κατασκευάστηκε ένα επίπεδο πιο πριν, στο Network Layer. Το πακέτο αυτό καλείται και *Datagram*. Τέλος, το πεδίο Cyclic Redundancy Check (CRC), παίζει το ρόλο του FEC, και ο δέκτης, αποκωδικοποιώντας το, μπορεί να εντοπίσει και να διορθώσει τυχόν

¹OSI model: Ένα γενικό μοντέλο που περιγράφει τις διάφορες διεργασίες που εφαρμόζει ένα υπολογιστικό σύστημα για την επικοινωνία του με άλλα συστήματα. Το OSI model κατηγοριοποιεί τις λειτουργίες σε επίπεδα (layers).

²Η σειρά των επιπέδων (Layers) την οποία πρέπει να απομνημονεύσει ο αναγνώστης δηλαδή είναι: Transport → Network → Link → Physical.

σφάλματα στο frame που εμφανίστηκαν κατά τη μετάδοση. Παρακάτω, παρατίθεται το Datagram του IPv4, του πιο συνηθισμένου πρωτοκόλλου του Network Layer, μαζί με το UDP Datagram. Υπενθυμίζεται ότι αυτό το πακέτο αυτούσιο είναι το Data field του Ethernet frame. Ο αναγνώστης θα προσέξει ότι και το IPv4 datagram, περιέχει και αυτό με τη σειρά του ένα πεδίο Data. Το πεδίο αυτό, έχει κατασκευαστεί επίσης ένα επίπεδο πιο ψηλά, στο Transport Layer, και ονομάζεται πακέτο UDP ή UDP Datagram [19]. Τα δεδομένα που περιέχονται στο πακέτο του UDP, είναι τα δεδομένα που έχουν ληφθεί από το FPGA μέσω των e-link από τις front-end boards. Έτσι είναι πλέον φανερό το τι γίνεται στο FPGA κατά τη λήψη πληροφοριών από τα e-link: Αρχικά κατασκευάζεται το πακέτο του UDP, όπου το πεδίο των Data του πακέτου γεμίζει με τα δεδομένα από τα e-link (Transport Layer), ύστερα το πρωτόκολλο IPv4 λαμβάνει το UDP Datagram και το επισυνάπτει αυτούσιο στο πεδίο δεδομένων του δικού του πακέτου (Network Layer), και τελικά στο Data field του Ethernet frame, εφαρμόζεται το IPv4 Datagram (Link Layer). Το frame αυτό, δρομολογείται τελικά σε ένα ολοκληρωμένο κύκλωμα εξωτερικό του FPGA, που μεταδίδει το frame σε μορφή ψηφιακού διαφορικού σήματος στη γραμμή του Ethernet. Στην ίδια γραμμή είναι συνδεδεμένοι οι υπόλοιποι κόμβοι, οι οποίοι αναμένουν να λάβουν κάποιο frame, με διεύθυνση παραλήπτη ίδια με τη MAC Address τους, ώστε να το διαβάσουν αφού το ξεπακετάρουν. Στο Κεφάλαιο 7 μελετάται η εφαρμογή του αλγορίθμου αυτού στο FPGA της L1DDC.



Σχήμα Α'.2.2: Δεξιά: Τα δεδομένα του e-link αποθηκεύονται προσωρινά σε μία μνήμη FIFO. Αυτή προωθεί τα δεδομένα στο Transport Layer. Κέντρο: Το UDP Datagram. Τα δεδομένα από το e-link πακετάρονται στο πεδίο Data του πακέτου UDP. Δεξιά: Το IPv4 Datagram στο Network Layer. Το UDP Datagram αποθηκεύεται στο πεδίο Data του πακέτου IPv4. Στη συνέχεια το τελικό πακέτο μπαίνει στο Data field του Ethernet frame.

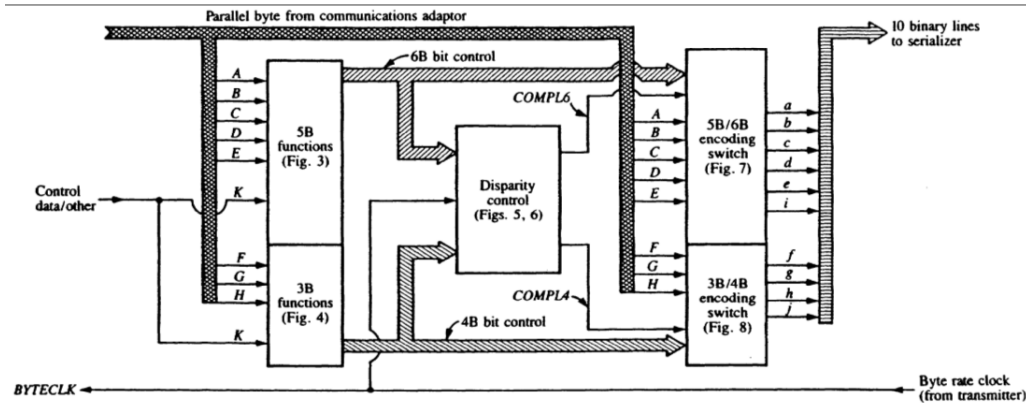
Α.3 8b/10b Encoding

Η κωδικοποίηση 8b/10b, χρησιμοποιείται πλέον ευρέως στα περισσότερα σειριακά μοτίβα επικοινωνίας (FireWire, SATA/SAS, Ethernet, HDMI, κ.ά.). Η πρώτη περιγραφή της μεθόδου έγινε το 1983 από δύο ερευνητές της IBM (βλ. [20]). Η βασική φιλοσοφία της κωδικοποίησης αυτής, συνοψίζεται στην κατάτμηση των προς αποστολή δεδομένων σε 8-bit κομμάτια, τα οποία στη συνέχεια κωδικοποιούνται σε χαρακτηριστές μήκους 10 bit. Το αποτέλεσμα που προκύπτει είναι μία σειρά δεδομένων όπου το πλήθος των bit 1 και bit 0 είναι ισορροπημένο [18]. Πιο συγκεκριμένα, ο αλγόριθμος κωδικοποίησης 8b/10b, εξασφαλίζει ότι δεν πρόκειται να υπάρχουν περισσότερα από πέντε συνεχόμενα 1 ή 0 στο σειριακό datastream, και ότι ο συνολικός αριθμός των 1 και 0 σε μία συμβολοσειρά αποτελούμενη από είκοσι ή περισσότερα bit, θα διαφέρει το πολύ κατά δύο.

Η σημασία της κωδικοποίησης 8b/10b είναι διττή [20]: Κατ' αρχάς η ισορροπία των λογικών 1 και 0, εξασφαλίζει και ισορροπία στα επίπεδα τάσης στις γραμμές όπου γίνονται οι σειριακές μεταδόσεις των δεδομένων. Αυτή η ισορροπία ονομάζεται *DC Balancing*. Το DC balancing, είναι απολύτως επιθυμητό, καθώς διευκολύνει την επικοινωνία μεταξύ διαφορετικών κυκλωμάτων, αφού έτσι αποφεύγεται εύκολα η συσσώρευση φορτίων στους πυκνωτές τους. Επίσης, η κωδικοποίηση 8b/10b, επειδή αυξάνει τις μεταπτώσεις μεταξύ των επιπέδων τάσεων, κάνει και τη διαδικασία του Clock Recovery σαφώς ευκολότερη. Για το λόγο αυτό ακριβώς, οι περισσότερες επικοινωνίες υψηλής ταχύτητας πλέον, χρησιμοποιούν αυτό τον τρόπο κωδικοποίησης. Τέλος, προς περαιτέρω διευκόλυνση του συγχρονισμού μεταξύ πομπού και δέκτη, το 8b/10b encoding, κωδικοποιεί και ειδικούς χαρακτηριστές (*K characters, Comma characters*) οι οποίοι χρησιμοποιούνται για λειτουργίες ελέγχου (π.χ. συμβολίζουν εντολές ABORT, RESET, IDLE) και για συγχρονισμό. Οι χαρακτηριστές συγχρονισμού καλούνται comma characters, είναι τρεις τον αριθμό (K.28.1, K.28.5, K.28.7)³ και τοποθετούνται στα άκρα των πακέτων. Εάν ο δέκτης αποκωδικοποιήσει έναν comma character, τότε γνωρίζει ότι η ευθυγράμμιση του χρονισμού του σε σχέση με το πακέτο που λαμβάνει είναι σωστή [20]. Έτσι ξεκινάει τη διαδικασία του deserialization. Η κωδικοποίηση 8b/10b, χρησιμοποιείται από το FPGA της L1DDC στην υλοποίηση

³0011111001, 0011111010, 0011111000 σε 10-bit μορφή, και 001111100, 001111101, 001111111 σε 8-bit, αντίστοιχα.

του Ethernet για την επικοινωνία με τον υπολογιστή. Επίσης, μαζί με το FPGA της MMFE8, εφαρμόζουν 8b/10b encoding/decoding στις γραμμές των e-link.

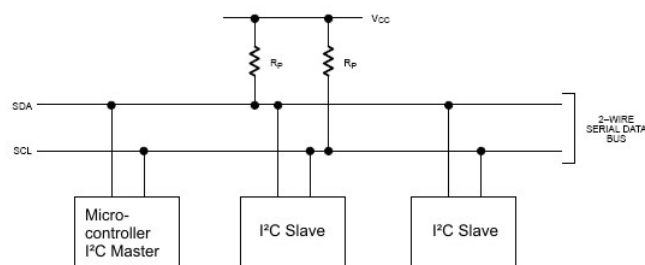


Σχήμα Α.3.1: Διάγραμμα ενός 8b/10b encoder. Ο κωδικοποιητής δέχεται ένα 8-bit input (ABCDEFGH), και η έξοδος του αποκρίνεται με ένα 10-bit παράλληλο σήμα (abcdeifghj). Το παράλληλο αυτό σήμα προωθείται στην είσοδο ενός serializer, ο οποίος μεταδίδει το σήμα σειριακά στη γραμμή επικοινωνίας [20].

Α.4 I²C Protocol

Το *Inter-Integrated-Circuit* (IIC, I2C, ή I²C) είναι ένα σειριακό πρωτόκολλο επικοινωνίας, που χρησιμοποιείται κυρίως για την διασύνδεση μεταξύ ολοκληρωμένων κυκλωμάτων. Υλοποιήθηκε πρώτη φορά το 1982 από την εταιρεία Philips. Χρησιμοποιεί δύο γραμμές, η μία εκ των οποίων φέρει το κοινό ρολόι (SCL), και η άλλη τα δεδομένα (SDA) [18]. Στο πρωτόκολλο αυτό, μπορούν να επικοινωνήσουν πολλά ολοκληρωμένα κυκλώματα μεταξύ τους, δηλαδή να είναι συνδεδεμένα όλα στο ίδιο ζεύγος γραμμών. Το ρολόι και οι εντολές, μεταδίδονται από έναν κόμβο, ονόματι *Master*. Ο *Master* μεταδίδει ένα περιοδικό τετραγωνικό παλμό, που είναι το ρολόι. Το πρότυπο ορίζει την συχνότητα αυτή στα 100 kHz (αν και πλέον υπάρχουν και γρηγορότερα πρότυπα για το I²C). Με έναν συγκεκριμένο τρόπο, ο *Master* ορίζει το Start signal μέσω του SDA, και όλα τα υπόλοιπα κυκλώματα που είναι συνδεδεμένα με τις γραμμές του SCL/SDA, παίρνουν αυτόματα το ρόλο του *Slave*. Μέσω του I²C, ο *Master*, μπορεί να διαβάσει τα περιεχόμενα μίας συγκεκριμένης διεύθυνσης της μνήμης του *Slave*, ή να κάνει μία νέα εγγραφή σε κάποια μνήμη του. Το κάθε κύ-

κλωμα, έχει μία 7-bit διεύθυνση, την *Slave Address*, ενώ διαθέτει και πολλές θέσεις στη μνήμη του, η κάθε μία εκ των οποίων έχει επίσης μία διεύθυνση (συνήθως με μήκος 8 bit, όπου ονομάζεται *Register Address*)⁴. Μετά το σήμα START, ο Master μεταδίδει την διεύθυνση του κυκλώματος με το οποίο θέλει να επικοινωνήσει, μαζί με ένα bit που υποδεικνύει αν θέλει να διαβάσει ή να γράψει στις μνήμες του κυκλώματος αυτού (R/W bit). Ο αντίστοιχος Slave, μετά τη μετάδοση της διεύθυνσής του, ανταποκρίνεται με ένα *Acknowledge bit* (ACK), όπου απλά κρατάει χαμηλά την τάση της γραμμής SDA για έναν κύκλο ρολογιού. Με τη λήψη του ACK, ο Master αποκρίνεται με τη διεύθυνση μνήμης που θέλει να προσπελάσει.



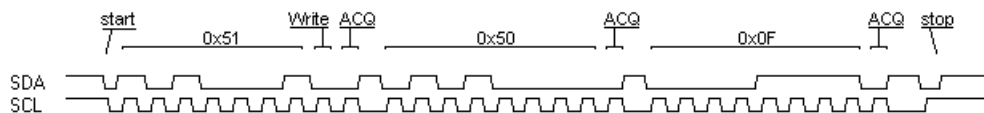
Σχήμα Α'.4.1: Διάγραμμα ενός μικροεπεξεργαστή που είναι ο I²C Master, και δύο άλλων ολοκληρωμένων κυκλωμάτων που είναι οι Slaves.

Εάν ο Master θελήσει για παράδειγμα να γράψει στη μνήμη ενός Slave, κρατάει αρχικά την τάση και των δύο γραμμών του πρωτοκόλλου ψηλά, και μετά κατεβάζει στο λογικό μηδέν την τάση του SDA. Αυτό είναι το σήμα START, και αμέσως μετά στο SCL μεταδίδεται το ρολόι των 100 kHz. Όλες οι υπόλοιπες μεταβάσεις κατάστασης του SDA, γίνονται καθώς το SCL είναι χαμηλά. Με το επόμενο tick του ρολογιού, ξεκινάει η μετάδοση της 7-bit Slave Address μαζί με το WRITE bit (λογικό μηδέν). Ο αντίστοιχος κόμβος/IC, αποκρίνεται κρατώντας χαμηλά την τάση στο SDA (ACK bit). Με τη λήψη του ACK, ο Master μεταδίδει την 8-bit διεύθυνση μνήμης (Register Address), και ο Slave αποκρίνεται πάλι με ACK, αν όντως διαθέτει αυτή τη διεύθυνση στη μνήμη του. Με τη λήψη και αυτού του ACK, ο Master μεταδίδει την τιμή που θέλει να εγγράψει στην διεύθυνση του Slave, και μετά το τελευταίο ACK, σηματοδοτείται το STOP από τον Master, το οποίο και κάνει κρατώντας ψηλά την τάση

⁴Το GBTx για παράδειγμα, έχει 366 θέσεις στη μνήμη του, με τιμές μήκους 8-bit. Οι μεταβλητές που αποθηκεύονται στις Register Addresses, ελέγχουν διάφορες λειτουργίες του ASIC [17].

του SCL, καθώς ανεβάζει από το λογικό μηδέν στο λογικό ένα την τάση του SDA. Εάν ο Master θέλει τώρα να διαβάσει τα περιεχόμενα μίας διεύθυνσης μνήμης ενός Slave, ξεκινάει ομοίως, αλλά μετά τη διεύθυνση του κυκλώματος επισυνάπτει το λογικό ένα (READ bit). Ο Slave αποκρίνεται με ACK, ο Master μεταδίδει τη διεύθυνση μνήμης, και ο Slave αποκρίνεται διαβιβάζοντας σειριακά τα περιεχόμενα της μνήμης στη γραμμή SDA. Το STOP μεταδίδεται τελικά από τον Master όπως πριν⁵.

Σε γενικές γραμμές, όλα τα ολοκληρωμένα κυκλώματα διαθέτουν τη δυνατότητα να πάρουν το ρόλο του Master ή του Slave. Ως προς τα FPGA, χάρη στην ευελιξία τους, το τι ρόλο μπορούν να πάρουν στις γραμμές του I²C, είναι καθαρά ζήτημα προγραμματισμού. Στο Κεφάλαιο 7, παρουσιάζεται το κομμάτι εκείνο του firmware του FPGA της L1DDC, όπου το FPGA παίζει το ρόλο του Master στην επικοινωνία του με το GBTx, το οποίο διαθέτει ένα ενσωματωμένο I²C Slave Interface [17]. Έτσι, ο χρήστης μπορεί να προγραμματίσει τους Registers του GBTx, και να διαβάσει τα περιεχόμενά τους, χρησιμοποιώντας ως Master το FPGA.



Σχήμα Α.4.2: Διάγραμμα του datastream στις γραμμές SCL/SDA μίας υλοποίησης πρωτοκόλλου I²C. Ο Master αρχικά "καλεί" το IC με διεύθυνση 0x51 (1010001) για να γράψει σε έναν register του. Ο Slave αποκρίνεται με ACK, και μετά διαβάζει τη διεύθυνση του register που θέλει να αλλάξει ο Master. Αυτή είναι η 0x50 (01010000). Τέλος, ο Master διαβιβάζει σειριακά το αλφαριθμητικό που θέλει να εγγράψει στη θέση 0x50 της μνήμης του Slave. Αυτό είναι το 0x0F (00001111). Μετά το ACK (που συμβολίζεται με ACQ στο διάγραμμα) από το Slave, ο Master ολοκληρώνει τη διαδικασία με το STOP.

⁵Συνήθως, και στις δύο περιπτώσεις, μετά τη λήψη του πρώτου ACK, ο Master προβαίνει σε *Repeated Start*, όπου μεταδίδει ξανά το σήμα Start στη γραμμή, πριν το serialization της διεύθυνσης μνήμης προς προσπέλαση.

VHDL Tutorial

Τα αρχικά *VHDL*, σημαίνουν *VHSIC Hardware Description Language*¹. Η γλώσσα VHDL, είναι μία γλώσσα περιγραφής αρχιτεκτονικής κυκλωμάτων σε υψηλό επίπεδο, η οποία χρησιμοποιείται κυρίως για τον προγραμματισμό FPGA, και για τον καθορισμό της γενικής δομής ενός ASIC, πριν αυτό μπει στη γραμμή παραγωγής [25]. Η VHDL υποστηρίζει τόσο τη μετάφραση της γλώσσας αυτής σε πραγματικά κυκλώματα μέσα στο FPGA, όσο και την προσομοίωση της απόκρισης του κυκλώματος, μέσω *simulators*. Στο παρόν Παράρτημα, θα μελετηθούν μερικοί απλοί κώδικες VHDL, και θα γίνει αναφορά στα εργαλεία που χρησιμοποιούνται για την σύνθεση των κωδίκων αυτών, τα οποία παρέχονται από την εταιρεία της οποίας τα FPGA ο χρήστης προγραμματίζει.

B.1 Xilinx® ISE Design Suite 14.7

Το περιβάλλον ανάπτυξης της Xilinx® που χρησιμοποιήθηκε για την εκπόνηση της παρούσας διπλωματικής, ήταν το ISE Design Suite 14.7. Σήμερα, η εταιρεία έχει σταματήσει την υποστήριξη του λογισμικού αυτού, καθώς έχει μεταβεί σε πιο εξελιγμένα περιβάλλοντα (Vivado Design Suite). Μολαταύτα, το ISE είναι ένα πολύ χρήσιμο εργαλείο, καθώς ήδη από μόνο του διαθέτει αρκετές επιπρόσθετες λειτουργίες που βοηθούν την ανάπτυξη firmware για ένα FPGA.

Κατ' αρχάς, το λογισμικό υποστηρίζει όλα τα FPGA της Xilinx®, και μπορεί να προσαρμόσει οποιονδήποτε κώδικα VHDL στα κυκλώματά της. Επίσης, μετά τη σύν-

¹Very High Speed Integrated Circuit

θεση κάποιου κώδικα, το πρόγραμμα μπορεί και κατασκευάζει μία σχηματική αναπαράσταση του προκύπτοντος κυκλώματος σε απλές πύλες, την οποία μπορεί και συμβουλεύεται ο χρήστης για εύκολο εντοπισμό λαθών, και για να έχει μία γενική επισκόπηση του κώδικά του (RTL Schematic).

Μία ακόμα δυνατότητα του ISE, είναι να παρέχει και το βοηθητικό πρόγραμμα ISim, το οποίο υλοποιεί σε ένα εικονικό FPGA, το κύκλωμα που συντέθηκε, μέσα σε ένα περιβάλλον προσομοίωσης. Ο χρήστης μπορεί και τροφοδοτεί τα διάφορα σημεία του κώδικα με σήματα (στις εισόδους και στις εξόδους των components), που μπορεί να είναι είτε λογικές τιμές, ή παλμοί ρολογιών, και να επιθεωρήσει με διάφορα διαγράμματα την απόκριση του firmware. Το πρόγραμμα προσομοίωσης ISim, είναι απολύτως απαραίτητο για τον σχεδιασμό οποιουδήποτε προγράμματος, αφού πριν γίνει η πραγματική υλοποίηση στο chip, η μελέτη της λειτουργίας του γίνεται μόνο με αυτόν τον τρόπο.

Συνήθως, όταν ένας κώδικας περάσει τον έλεγχο της προσομοίωσης, είναι έτοιμος για υλοποίηση στο κύκλωμα. Για να γίνει αυτό όμως, πρέπει να οριστούν από τον προγραμματιστή τα λεγόμενα *constraints*. Όταν γράφεται ένας κώδικας σε VHDL, θα υπάρχει το λεγόμενο *Top-Level port*, οι εισοδοί και έξοδοι του οποίου θα είναι ουσιαστικά τα pins του FPGA. Ο χρήστης, έχοντας σαν αναφορά το σχεδιάγραμμα της πλακέτας στην οποία δουλεύει, πρέπει να αντιστοιχήσει το κάθε I/O pin του κώδικα, σε κάθε I/O pin του FPGA, προκειμένου να οδηγήσει τα σήματα όπως πρέπει. Δεν μπορεί για παράδειγμα να θέλει να τροφοδοτήσει ένα PLL στον κώδικά του με ένα ρολόι, και η είσοδος του ρολογιού στο top-level να είναι οδηγημένη στην δρομολόγηση που καταλήγει σε έναν διακόπτη της πλακέτας. Τα constraints ορίζονται στο .ucf file, το οποίο μπορεί να συνταχθεί με έναν κειμενογράφο, ή γραφικά, με τη χρήση του FloorPlanner, ο οποίος απεικονίζει όλα τα I/O pins του FPGA.

Εκτός από τον FloorPlanner, σημαντικός είναι και ο ρόλος που παίζει ο FPGA Editor στην μελέτη ενός firmware. Αφού οριστούν και τα constraints, και αν ο Synthesizer κρίνει ότι το FPGA μπορεί να υλοποιήσει το firmware, ο FPGA Editor προσφέρει τη δυνατότητα στο χρήστη να επιθεωρήσει το firmware, όπως θα είναι στο FPGA σε επίπεδο κυκλώματος. Ο FPGA Editor απεικονίζει όλα τα LUT, FF, CLB που χρησιμοποιούνται σε έναν κώδικα, και όλες τις μεταξύ τους διασυνδέσεις. Δείχνει τα PLL και τις μνήμες που ενεργοποιεί ο Synthesizer προκειμένου να κατασκευάσει

το κύκλωμα που περιγράφεται στην HDL του προγραμματιστή. Ο χρήστης, μπορεί έτσι να επιθεωρήσει σε πραγματικό πλέον επίπεδο τον κώδικά του, και φυσικά να προβεί σε αλλαγές στην αρχιτεκτονική του FPGA. Τα περισσότερα μεγάλα project πλέον [21, 24], δεν απαιτούν μόνο τη σωστή σύνταξη ενός κώδικα VHDL, αλλά και τη λεπτομερή αναδιάταξη των LUT και των δρομολογήσεων των σημάτων, προκειμένου το κύκλωμα να λειτουργεί με τον βέλτιστο δυνατό τρόπο. Βέβαια, αξίζει να σημειωθεί ότι για να ολοκληρωθεί η σύνθεση και να δημιουργηθεί το bitstream που θα ενεργοποιήσει τις αντίστοιχες SRAM, το λογισμικό προβαίνει σε πολλούς ελέγχους προκειμένου να αποφανθεί αν είναι δυνατή η υλοποίηση του κυκλώματος που περιγράφει η HDL σε συνδυασμό με τα constraints του χρήστη. Πραγματοποιεί για παράδειγμα, αναλύσεις στα σήματα χρονισμού που τροφοδοτούν τις flip-flop (και όχι μόνο) σε όλο το design, ώστε να κρίνει αν οι καθυστερήσεις στους παλμούς καθώς αυτοί περνάνε από τα καλώδια του chip, είναι αρκετά μικρές ώστε να μην επηρεάσουν τη λειτουργικότητα του firmware. Αυτό φυσικά, δε σημαίνει ότι κανένα design δεν επιδέχεται βελτιώσεων. Κάθε άλλο. Μπορεί να αφιερωθεί πολύς χρόνος στην αλλαγή των slice, ή του routing μέσα σε ένα firmware. Αυτό το έργο, το επιτελεί ο FPGA Editor.

Ένα τελευταίο χρήσιμο και άξιο αναφοράς εργαλείο, είναι το ChipScope. Αναφέρθηκε παραπάνω η λειτουργικότητα του iSim, ότι δηλαδή προβαίνει σε προσομοίωση του firmware. Το μειονέκτημα του iSim, είναι ότι ουσιαστικά δεν είναι πραγματικό. Με τη χρήση όμως του ChipScope, είναι δυνατός ο έλεγχος όλου του firmware, κατά την υλοποίησή του από το FPGA. Αυτό επιτυγχάνεται με ειδικά components της Xilinx[®], τα οποία αν τοποθετηθούν σε συγκεκριμένα σημεία του κώδικα, συλλέγουν πληροφορίες για τα σήματα που επιθυμεί ο χρήστης. Ο χρήστης μπορεί να επιθεωρήσει αυτά τα αποτελέσματα μέσω του ChipScope.

Περισσότερα για όλα αυτά όμως θα ξεκαθαριστούν στη συνέχεια, στην πράξη. Στις επόμενες ενότητες του παρόντος Κεφαλαίου, θα παρουσιαστούν ορισμένοι απλοί κώδικες σε VHDL, ώστε ο αναγνώστης να αντιληφθεί πως λειτουργεί η γλώσσα, και πως το λογισμικό του ISE υποβοηθάει την ανάπτυξη του firmware γενικώς.

B'.2 Hello World!

Όταν κανείς καταπιαστεί για πρώτη φορά με την εκμάθηση μίας γλώσσας, το πρώτο πράγμα που διαβάζει συνήθως σε ένα βιβλίο που τον καθοδηγεί, είναι η σύνταξη ενός προγράμματος που τυπώνει στην οθόνη του υπολογιστή του, τη φράση "Hello World!". Κάτι παρόμοιο για τη VHDL δεν υπάρχει, όμως μπορεί να επινοηθεί ένας κώδικας παρόμοιος απλότητας. Το...Hello World της VHDL λοιπόν εδώ, θα είναι ένας κώδικας, ο οποίος υλοποιεί μία πύλη OR τριών εισόδων.

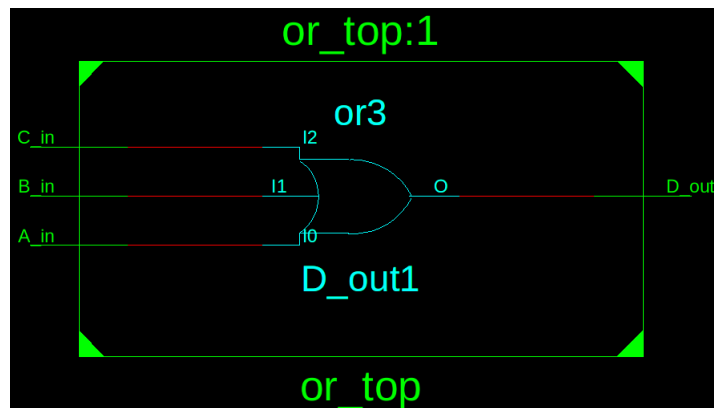
VHDL Code B'.1: or_top

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity or_top is
5
6     port (
7         A_in  : in  std_logic;
8         B_in  : in  std_logic;
9         C_in  : in  std_logic;
10        D_out : out std_logic);
11 end or_top;
12
13 architecture RTL of or_top is
14
15 begin
16
17     D_out <= A_in OR B_in OR C_in;
18
19 end RTL;
```

Στις γραμμές 1-2 του κώδικα B'.1, ορίζονται οι βιβλιοθήκες που χρησιμοποιούνται για την υλοποίηση. Στις γραμμές 4-11, ορίζεται το entity του κυκλώματος. Αυτός ο κώδικας, όπως κάθε αυτούσιος κώδικας VHDL, περιγράφει ένα κύκλωμα, το οποίο μπορεί να θεωρηθεί και ως μαύρο κουτί. Τα σήματα που αναφέρονται στις γραμμές 7-10, είναι τα pins του μαύρου κουτιού, τα σήματα δηλαδή που επικοινωνούν με τον έξω κόσμο. Αυτά εδώ είναι τα A_in, B_in, και C_in που είναι σήματα εισόδου, και το D_out που είναι σήμα (ή pin) εξόδου. Το πρόθεμα std_logic, υποδεικνύει ότι τα pins ανήκουν στο σύνολο του "standard logic", δηλαδή είναι απλά ψηφιακά σήματα

λογικού ένα ή μηδέν. Το επόμενο ενδιαφέρον σημείο του κώδικα, είναι η γραμμή 17, όπου φανερώνεται και ο τρόπος λειτουργίας του κυκλώματος (εξ ου και η λέξη *architecture* στη γραμμή 13). Η έξοδος, είναι ουσιαστικά μία πύλη OR τριών εισόδων.

Η υλοποίηση του κώδικα Β'.1 είναι εφικτή. Το ISE δίνει τη δυνατότητα οπτικοποίησης του κώδικα σε κύκλωμα απλών πυλών:



Σχήμα Β'.2.1: Αναπαράσταση του κώδικα Β'.1 σε κύκλωμα απλών πυλών. RTL Schematic.

Ως...εκ θαύματος, η εικόνα έχει μία πύλη OR. Ενδιαφέρον επίσης παρουσιάζει και η αντιστοιχία των ονομασιών που έδωσε ο χρήστης στα σήματα και στο entity του κώδικα, με όσα απεικονίζονται στο RTL Schematic. Προκειμένου τώρα ο αναγνώστης να αντιληφθεί ότι μέσα σε έναν κώδικα VHDL μπορεί να συμπεριληφθούν άφθονα επίπεδα με άλλους κώδικες που περιγράφουν άλλα κυκλώματα, ο κώδικας *or_top* θα υποστεί μερικές επεκτάσεις. Για την ώρα, θα παρατεθεί ο κώδικας που υλοποιεί έναν απλό multiplexer.

VHDL Code Β'.2: my_mux

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity my_mux is
5
6     port (
7         I0 : in std_logic;
```

```

8         I1  : in std_logic ;
9         SEL : in std_logic ;
10        O   : out std_logic );
11 end my_mux ;
12
13 architecture Behavioral of my_mux is
14
15 begin
16
17     process (I0 , I1 , SEL)
18     begin
19
20         case SEL is
21         when '0' =>
22             O <= I0 ;
23         when '1' =>
24             O <= I1 ;
25         when others =>
26             O <= I0 ;
27         end case ;
28
29     end process ;
30
31
32 end Behavioral ;

```

Ο κώδικας αυτός είναι ένας τυπικός πολυπλέκτης 2-σε-1. Δέχεται δύο σήματα εισόδου (I0,I1), και η τιμή της εξόδου του (O), θα ταυτίζεται με το I0 αν το SEL είναι χαμηλά, ή με το I1 αν το SEL είναι ψηλά. Στα πρώτα σημεία του κώδικα μέχρι τη γραμμή 10, δεν υπάρχει κάποια διαφορά με τον κώδικα or_top, μόνο ότι άλλαξαν οι ονομασίες των pins. Η λογική του mux, συνοψίζεται στις γραμμές 17-27, όπου η process, υλοποιεί έναν αλγόριθμο που λαμβάνονται αποφάσεις, ανάλογα την τιμή μίας μεταβλητής (εντολή case), η οποία εν προκειμένω είναι το SEL. Το επόμενο κύκλωμα σε VHDL, θα κάνει σαφή τον λόγο παράθεσης του κώδικα του mux:

VHDL Code B'3: or_top2

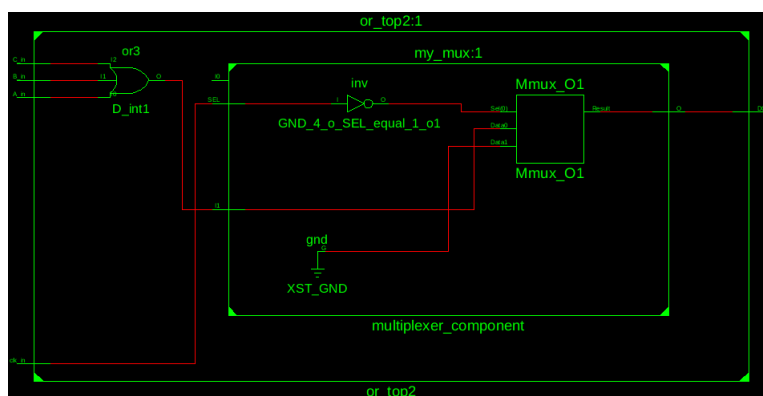
```

1 library IEEE ;
2 use IEEE.STD_LOGIC_1164.ALL ;
3

```

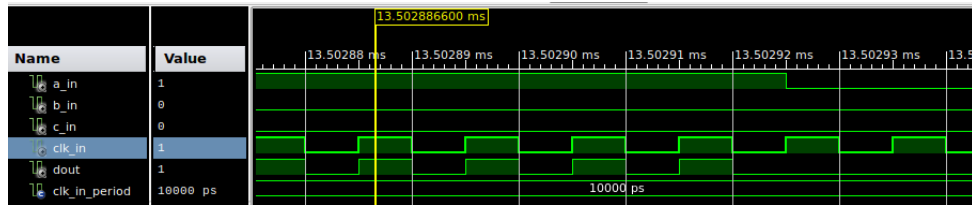
```
4 entity or_top is
5
6     port(
7         A_in  : in  std_logic;
8         B_in  : in  std_logic;
9         C_in  : in  std_logic;
10        clk_in : in std_logic;
11        DOUT  : out std_logic);
12 end or_top;
13
14 architecture RTL of or_top is
15
16 COMPONENT my_mux
17     PORT(
18         I0 : IN std_logic;
19         I1 : IN std_logic;
20         SEL : IN std_logic;
21         O  : OUT std_logic
22     );
23 END COMPONENT;
24
25     signal tied_to_gnd_i : std_logic;
26     signal D_int        : std_logic;
27
28 begin
29
30     tied_to_gnd_i <= '0';
31
32     multiplexer_component: my_mux PORT MAP(
33         I0 => tied_to_gnd_i ,
34         I1 => D_int ,
35         SEL => clk_in ,
36         O => DOUT
37     );
38
39     D_int <= A_in OR B_in OR C_in;
40
41 end RTL;
```

Το κύκλωμα που περιγράφεται από τον κώδικα Β'.3, έχει παρόμοια σήματα εισόδου/εξόδου με τον πρώτο κώδικα or_top, με μερικές διαφορές. Τώρα το σήμα εξόδου καλείται DOUT (σύντμηση της λέξης "Data Out"), και υπάρχει και μία είσοδος που δέχεται ένα ρολόι (clk_in). Στις γραμμές 16-23, γίνεται το component declaration. Ο αναγνώστης παραπέμπεται στον κώδικα Β'.2, όπου περιγράφεται ο multiplexer. Το component του παρόντος κώδικα έχει τα ίδια ports με τον my_mux, και όπως μπορεί να φανταστεί κανείς, το my_mux είναι τώρα ένα ηλεκτρονικό υποσύστημα του κυκλώματος του or_top2. Στις γραμμές 25 και 26, ορίζονται τα signals. Τα signals είναι εσωτερικά σήματα, τα οποία δεν βγαίνουν έξω από το κύκλωμα στο οποίο ορίζονται. Το ένα αντιπροσωπεύει τη γείωση (tied_to_gnd_i), και το άλλο ένα εσωτερικό σήμα που θα χρησιμοποιηθεί παρακάτω. Στη γραμμή 30, ορίζεται και επίσημα η γείωση (το $=$ '0' δεσμεύει ένα σήμα με το λογικό μηδέν), ενώ στις γραμμές 32-37 γίνεται η διασύνδεση του υποκυκλώματος του mux που ορίστηκε προηγουμένως με τα υπόλοιπα εσωτερικά και εξωτερικά σήματα του κυκλώματος. Το αποτέλεσμα της πύλης OR, που υπολογίζεται στη γραμμή 39, δεν πηγαίνει πλέον κατευθείαν στην έξοδο, αλλά στο I1 του πολυπλέκτη. Το I0 είναι γειωμένο. Το σήμα επιλογής, είναι το ρολόι εισόδου. Η έξοδος του πολυπλέκτη, είναι και η έξοδος του top_level του κυκλώματος. Αυτό σημαίνει αμέσως-αμέσως, ότι όταν το ρολόι είναι μηδέν, τότε η έξοδος DOUT θα είναι και αυτή μηδέν (αφού θα επιλέγεται η γείωση από τον mux). Ο μόνος τρόπος να είναι η έξοδος ψηλά, είναι κατ' αρχάς να είναι και το ρολόι ψηλά, και μία τουλάχιστον εκ των λογικών εισόδων (A_in κ.ο.κ.) να είναι 1, ώστε η πύλη OR να δώσει λογική τιμή 1 στο I1 του πολυπλέκτη. Το κύκλωμα περιγράφεται σχηματικά έτσι:



Σχήμα Β'.2.2: Αναπαράσταση του κώδικα Β'.3 σε κύκλωμα απλών πυλών. RTL Schematic.

Η επιβεβαίωση ότι το κύκλωμα αυτό λειτουργεί όπως έχει προβλεφθεί, δίνεται από το ISim.



Σχήμα Β'.2.3: Προσομοίωση του κυκλώματος από το ISim. Όσο το A_in είναι ψηλά, τότε το D_int (δεν απεικονίζεται εδώ) είναι και αυτό ψηλά λόγω της πύλης OR. Επομένως, το ρολόι καθορίζει πλέον αν θα προκύψει στο DOUT το ένα ή το μηδέν. Όσο το clk_in είναι ένα, τότε επιλέγεται το D_int να προωθηθεί στην έξοδο. Όσο είναι μηδέν, η έξοδος αντιστοιχίζεται με τη γείωση του πολυπλέκτη. Όταν όμως το A_in γίνει μηδέν (και δεδομένου ότι και οι άλλες εισοδοί της OR είναι χαμηλά), τότε το DOUT είναι σίγουρα και αυτό μηδέν.

Με αυτά τα απλά παραδείγματα, ο αναγνώστης πλέον έχει μία γενική ιδέα για το πως λειτουργεί η VHDL. Πριν παρατεθούν όμως λίγο πιο εξελιγμένοι κώδικες που αφορούν κυκλώματα ακολουθιακής λογικής και όχι μόνο συνδυαστικής, θα γίνει μία σύντομη αναφορά στα IP Cores της Xilinx®.

IP Cores

Τα *Intellectual Property Cores (IP Cores)*, είναι εξειδικευμένα υποσυστήματα τα οποία παρέχει η Xilinx®, και μπορούν να υλοποιηθούν από το FPGA, είτε από εξειδικευμένα hard blocks, είτε από κοινά CLB. Τα IP Cores διαφέρουν συνήθως από FPGA σε FPGA, ιδίως όταν αυτά υλοποιούνται από hard blocks. Μερικά IP Cores που χρησιμοποιούνται κατά κόρον, είναι τα IP Cores της FIFO, των Transceivers, ή των μικροεπεξεργαστών. Ο προγραμματιστής μπορεί να συμπεριλάβει ελεύθερα ένα τέτοιο υποσύστημα στον κώδικά του, με τη βοήθεια του ISE και του IP Core Wizard. Ένα γραφικό μενού επιτρέπει στον χρήστη να κάνει μικροαλλαγές στον τρόπο λειτουργίας του IP Core. Τελικά ο wizard παράγει το ηλεκτρονικό αυτό υποσύστημα, και παρέχει στο χρήστη ένα αρχείο .vhd με το component declaration του IP Core προκειμένου να το χρησιμοποιήσει στο firmware του όπως αυτός θελήσει. Για την εκπόνηση της παρούσας διπλωματικής, χρησιμοποιήθηκαν σε πολλές περιπτώσεις αρκετά IP Cores.

B'.3 Serializer

Προκειμένου ο αναγνώστης να εξοικειωθεί περισσότερο με τον τρόπο σύνταξης της VHDL και με τις δυνατότητές της, κρίνεται απαραίτητη η παράθεση ενός κυκλώματος ακολουθιακής λογικής. Σε αυτή την ενότητα, θα κατασκευαστεί ένα firmware που περιγράφει έναν serializer. Ο serializer αυτός, θα δέχεται στην είσοδό του ένα διάνυσμα από 16 bit, τα οποία και θα δρομολογεί σειριακά σε μία διαφορετική γραμμή εκτός του FPGA. Η ροή των bit θα εναλλάσσεται κάθε ένα δευτερόλεπτο, όπου για ένα *sec* θα είναι στα 80 Mb/s, και μετά για το επόμενο *sec* θα ανεβαίνει στα 160 Mb/s. Μετά θα ξαναπέφτει στα 80, κ.ο.κ. Αυτό το firmware, θα συνταχθεί με τέτοιο τρόπο ώστε να μπορεί να υλοποιηθεί σε πραγματικό FPGA, επομένως θα γίνει και εκτενής χρήση I/O buffers, εν αντιθέσει με την προηγούμενη ενότητα, όπου κάτι τέτοιο δε συνέβη. Αρχικά λοιπόν παρατίθεται ο κώδικας του μετρητή (counter), ο οποίος είναι βασικό εργαλείο για τη μέτρηση χρόνου σε ένα FPGA:

VHDL Code B'.4: counter.vhd

```
1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity counter is
6     generic (cnt_max : natural := 40_000_000); --40_000_000
7
8     port(
9         clk : in std_logic;
10        rst : in std_logic;
11        slow : out std_logic);
12 end counter;
13
14 architecture Behavioral of counter is
15
16 begin
17
18     process(clk , rst) is
19
20         variable cnt0 : natural range 0 to cnt_max := 0;
21         variable cnt1 : natural range 0 to cnt_max := 0;
22
```

```

23      begin
24          if (rising_edge (clk)) then—clk
25              if (rst = '1') then—rst
26                  cnt1 := 0;
27                  cnt0 := 0;
28                  slow <= '0';
29              else
30                  if (cnt0 < cnt_max) then—cnt0
31                      cnt0 := cnt0 + 1;
32                      slow <= '1';
33                  elsif (cnt0 = cnt_max) then
34                      slow <= '0';
35                      if (cnt1 < cnt_max) then—cnt1
36                          cnt1 := cnt1 + 1;
37                      elsif (cnt1 = cnt_max) then
38                          cnt1 := 0;
39                          cnt0 := 0;
40                      else null;
41                  end if ;—cnt1
42                  else null;
43              end if ;—cnt0
44          end if ;—rst
45      end if ;—clk
46
47      end process ;
48
49 end Behavioral ;

```

Ο κώδικας του counter δέχεται ένα ρολόι, ένα σήμα reset (rst) και η έξοδος του είναι ένα λογικό σήμα που καλείται slow. Η χρήση του θα γίνει σαφής παρακάτω. Αρχικά υλοποιείται μία process, η οποία είναι η καρδιά των κυκλωμάτων ακολουθιακής λογικής όπως αυτά ορίζονται στη VHDL. Στους κώδικες που έχουν παρατεθεί μέχρι στιγμής, η επεξεργασία των σημάτων γινόταν παράλληλα, καθώς δεν υπήρχε κάποιο σήμα ρολογιού που να χρόνιζε κάποια flip-flop. Εδώ όμως, ορίζεται ουσιαστικά μία πύλη flip-flop, στην γραμμή 24. Το if(rising_edge(clk)), ορίζει μία συνθήκη: ότι οποιαδήποτε αλλαγή κατάστασης, θα συμβεί μόνο αν εντοπιστεί από το κύκλωμα ένας παλμός ρολογιού με ανοδική τάση (ο ορισμός δηλαδή της πύλης flip-flop). Η ιδιομορφία της σύνταξης της process, έγκειται στο γεγονός ότι οι εντολές που

βρίσκονται μέσα της, εκτελούνται *σειριακά*, εν αντιθέσει με τη VHDL γενικότερα, που επειδή περιγράφει κυκλώματα, όλα συμβαίνουν παράλληλα. Όταν ικανοποιείται μία συνθήκη μέσα στην process, η process μεταβαίνει στην επόμενη γραμμή, και προβαίνει σε μία εντολή, είτε αυτή είναι κάποια ανάθεση σήματος, είτε προβαίνει σε άλλο έλεγχο συνθήκης (if statement).

Αν λοιπόν το κύκλωμα εντοπίσει παλμό ρολογιού, ελέγχει αρχικά αν το reset είναι ψηλά (synchronous reset). Στη συνέχεια ελέγχει την τιμή του counter και αν αυτός είναι μικρότερος από το cnt_max, τότε προσθέτει +1 στην τιμή του. Μετά βγαίνει από τη λούπα, και επιστρέφει στο ίδιο σημείο, στον επόμενο παλμό ρολογιού, και επανελέγχει τη συνθήκη. Ο κώδικας είναι φτιαγμένος με τέτοιο τρόπο, ώστε όταν το cnt0/cnt1 είναι αρχικά μηδέν, να φτάνει στη μέγιστη τιμή του μετά από ένα δευτερόλεπτο. Αυτό επιτυγχάνεται αν το ρολόι εισόδου έχει συχνότητα 40 *Mhz*. Η περίοδος τότε είναι 25 *ns*. Στη γραμμή 6, βρίσκεται η μέγιστη τιμή του μετρητή, η οποία είναι 40000000. Πολλαπλασιάζοντας την περίοδο με το cnt_max, λαμβάνει κανείς ένα δευτερόλεπτο. Στον κώδικα ο counter είναι διπλός, δηλαδή για ένα δευτερόλεπτο κρατάει την τιμή του slow ψηλά (γραμμή 32) και μετά την κατεβάζει για ένα ακόμα δευτερόλεπτο (cnt1). Το κύκλωμα ενός απλού serializer τώρα, ορίζεται ως εξής :

VHDL Code B'5: simple_serializer.vhd

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity ser_component is
5
6     generic(len : natural := 16);
7
8     port(
9         clk : in std_logic;
10        rst : in std_logic;
11        serData : out std_logic);
12 end ser_component;
13
14 architecture Behavioral of ser_component is
15
16 begin
17
18 process(clk , rst) is

```



```

19
20 variable index: natural range 0 to len := 0;
21 constant vector: std_logic_vector(len-1 downto 0):="1101110011010001";
22
23     begin
24         if (rising_edge (clk)) then—clk
25             if (rst = '1') then—rst
26                 index := 0;
27             else
28                 if (index < len-1) then—index
29                     serData <= vector(index);
30                     index := index + 1;
31                 elsif (index = len-1) then
32                     serData <= vector(index);
33                     index := 0;
34                 else null;
35                 end if;—index
36             end if; —rst
37
38         end if;—clk
39
40     end process;
41
42 end Behavioral;

```

Ο serializer, δέχεται στην είσοδό του, ή έχει αποθηκευμένο στη μνήμη του (όπως συμβαίνει εδώ), το διάνυσμα από bit που θα μεταδώσει σειριακά στην έξοδό του (serData). Η τιμή αυτού του 16-bit διανύσματος, βρίσκεται στη γραμμή 21. Πάλι, όπως και πριν, το κύκλωμα είναι ακολουθιακό, και η πρώτη συνθήκη της process ικανοποιείται μόνο αν εντοπιστεί παλμός ρολογιού (γραμμή 25). Στη γραμμή 28 ξεκινάει ουσιαστικά το serialization. Υπάρχει ένας index, ο οποίος διατρέχει τις θέσεις του διανύσματος (vector), ξεκινώντας από τη θέση μηδέν. Στο πρώτο tick, ο έλεγχος για το αν ο index είναι μικρότερος της τιμής της τελευταίας θέσης (len-1) προφανώς ικανοποιείται, αφού ο index αρχικοποιείται από το μηδέν (γραμμή 20). Η έξοδος τότε, παίρνει την τιμή του διανύσματος vector, στην αντίστοιχη θέση που βρίσκεται ο index (γραμμή 29). Ο index στη συνέχεια αυξάνει κατά ένα (γραμμή 30). Στο επόμενο tick, η έξοδος serData θα αλλάξει, αφού ο index αυξήθηκε προηγουμένως κατά ένα. Η διαδικασία συνεχίζεται μέχρι ο index να τερματίσει στην τελική θέση, όπου

και μηδενίζεται για να επανέλθει στο μηδέν για να συνεχίσει η διαδικασία. Από όλη αυτή τη διαδικασία, προκύπτει μία κυματομορφή, η οποία θα αλλάζει καταστάσεις σύμφωνα με την περίοδο του ρολογιού εισόδου. Όσο μεγαλύτερη η συχνότητα του ρολογιού, τόσο πιο γρήγορα θα γίνονται οι μεταβάσεις, αφού τόσο γρηγορότερα θα αλλάζει θέσεις και ο index.

Επειδή όμως αναφέρθηκε πριν ότι ο σκοπός του κυκλώματος είναι να προβαίνει σε serialization των δεδομένων με δύο διαφορετικές ταχύτητες, εμφανίζεται η ανάγκη για ενεργοποίηση ενός PLL. Επίσης, το ρολόι του serializer πρέπει να παίρνει δύο διαφορετικές συχνότητες. Η υλοποίηση όλων αυτών, θα γίνει στο top_level του κυκλώματος.

VHDL Code B'6: serializer_top.vhd

```

1 library IEEE;
2 library UNISIM;
3 use IEEE.STD_LOGIC_1164.ALL;
4 use UNISIM.VComponents.all;
5
6 entity serializer_top is
7     port(
8         clk40_in : in  std_logic;
9         tx_p     : out std_logic;
10        tx_n     : out std_logic);
11 end serializer_top;
12
13 architecture RTL of serializer_top is
14
15     COMPONENT ser_component
16     PORT(
17         clk : IN std_logic;
18         rst : IN std_logic;
19         serData : OUT std_logic
20     );
21     END COMPONENT;
22
23     COMPONENT counter
24     PORT(
25         clk : IN std_logic;
26         rst : IN std_logic;

```

```
27         slow : OUT std_logic
28         );
29     END COMPONENT;
30
31     signal tied_to_gnd_i : std_logic;
32     signal tied_to_vcc_i : std_logic;
33     signal clk40         : std_logic;
34     signal clk40_buf    : std_logic;
35     signal clk80        : std_logic;
36     signal clk80_buf    : std_logic;
37     signal clk160       : std_logic;
38     signal clk160_buf   : std_logic;
39     signal clk40_cnt    : std_logic;
40     signal feedback    : std_logic;
41     signal pll_locked  : std_logic;
42     signal rst_i       : std_logic;
43     signal txclk       : std_logic;
44     signal slow        : std_logic;
45
46     signal serData      : std_logic;
47
48 begin
49
50     tied_to_gnd_i <= '0';
51     tied_to_vcc_i <= '1';
52     rst_i <= not pll_locked;
53
54     global_clk : IBUFG
55         generic map (
56             IBUF_LOW_PWR => FALSE,
57             IOSTANDARD => "LVCMOS33")
58         port map (
59             O => clk40 ,
60             I => clk40_in
61         ); -- End of IBUFG_inst instantiation
62
63     buf40 : BUFG
64         port map (
65             O => clk40_cnt , -- 1-bit output: Clock output
66             I => clk40_buf -- 1-bit input: Clock input
67         );
```

```

68
69     buf80 : BUFG
70         port map (
71             O => clk80 , — 1-bit output: Clock output
72             I => clk80_buf — 1-bit input: Clock input
73         );
74
75     buf160 : BUFG
76         port map (
77             O => clk160 , — 1-bit output: Clock output
78             I => clk160_buf — 1-bit input: Clock input
79         );
80
81     output_diff_buff : OBUFDS
82         generic map (
83             IOSTANDARD => "LVDS_25", — Specify the output
84             SLEW => "SLOW") — Specify the output
85         port map (
86             O => tx_p , — Diff_p output
87             OB => tx_n , — Diff_n output
88             I => serData — Buffer input
89         );
90 — End of OBUFDS_inst instantiation
91
92     BUFGMUX_component : BUFGMUX
93         port map (
94             O => txclk ,
95             I0 => clk160 ,
96             I1 => clk80 ,
97             S => slow
98         );
99
100     serializing_component : ser_component PORT MAP(
101     clk => txclk ,
102     rst => rst_i ,
103     serData => serData
104 );
105
106     counting_component: counter PORT MAP(
107     clk => clk40_cnt ,
108     rst => rst_i ,

```

```

109         slow => slow
110     );
111
112
113
114     PLLE2_BASE_inst : PLLE2_BASE
115 generic map (
116     BANDWIDTH      => "OPTIMIZED", — OPTIMIZED, HIGH, LOW.
117     CLKFBOUT_MULT  => 40, — Multiply value dor all CLKOUT, (2–64).
118     CLKFBOUT_PHASE => 0.0, — Phase offset.
119     CLKIN1_PERIOD  => 25.0, — Input clock period.
120     CLKOUT0_DIVIDE => 1,
121     CLKOUT1_DIVIDE => 10,
122     CLKOUT2_DIVIDE => 20,
123     CLKOUT3_DIVIDE => 40,
124     CLKOUT4_DIVIDE => 1,
125     CLKOUT5_DIVIDE => 1,
126
127 port map(
128     — Clock Outputs: 1-bit (each) output.
129     CLKOUT0 => open ,
130     CLKOUT1 => clk160_buf ,
131     CLKOUT2 => clk80_buf ,
132     CLKOUT3 => clk40_buf ,
133     CLKOUT4 => open ,
134     CLKOUT5 => open ,
135     _____
136     — Feedback Clocks: 1-bit (each) output: Clock feedback ports
137     CLKFBOUT => feedback , — 1-bit output: Feedback clock
138     _____
139     — Status Port: 1-bit (each) output: PLL status ports
140     LOCKED => pll_locked , — 1-bit output: LOCK
141     _____
142     — Clock Input: 1-bit (each) input: Clock input
143     CLKIN1 => clk40 , — 1-bit input: Input clock
144     _____
145     — Control Ports: 1-bit (each) input: PLL control ports
146     PWRDWN => tied_to_gnd_i , — 1-bit input: Power-down
147     RST     => tied_to_gnd_i , — 1-bit input: Reset
148     _____
149     — Feedback Clocks: 1-bit (each) output: Clock feedback ports

```

```

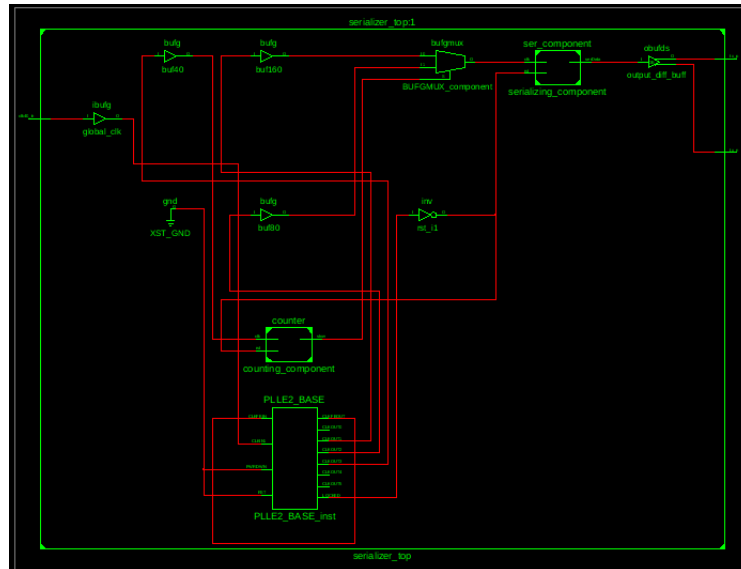
150     CLKFBIN => feedback — 1-bit input: Feedback clock
151     );
152
153 — End of PLLE2_BASE_inst instantiation
154
155
156 end RTL;

```

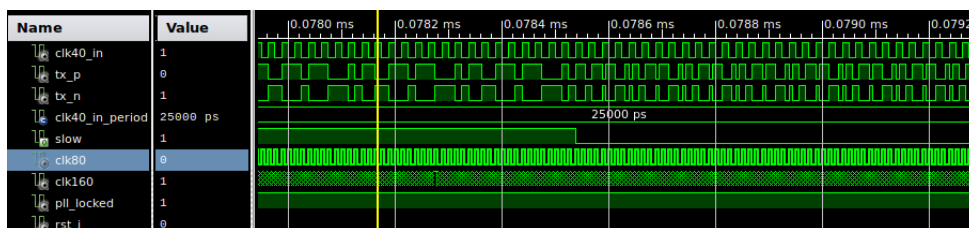
Αρχικά, το κύκλωμα επικοινωνεί με το εξωτερικό του FPGA με τρία pins: Το clk40_in, που δέχεται ένα ρολόι 40 Mhz, και τα pins εξόδου tx_p και tx_n, που υλοποιούν τη διαφορική μετάδοση της πληροφορίας έξω από το chip. Μετά υλοποιούνται τα components που ορίζονται από τους δύο προηγούμενους κώδικες που παρατέθηκαν προηγουμένως. Από τις γραμμές 54 μέχρι 79, ορίζονται οι clock buffers. Υπάρχει ένας global clock input buffer, (IBUFG), ο οποίος δέχεται στην είσοδό του το ρολόι clk40_in και το διοχετεύει στο PLL (γραμμή 143). Οι υπόλοιποι buffers λαμβάνουν τα ρολόγια εξόδου του PLL (γραμμές 130-132) και τα διοχετεύουν σε έναν πολυπλέκτη για ρολόγια (92-98). Το σήμα επιλογής του πολυπλέκτη, έρχεται από το component του counter (γραμμή 109), το οποίο όπως έχει ήδη γίνει σαφές, εναλλάσσει την τιμή του από 0 σε 1 με περίοδο ενός δευτερολέπτου. Δηλαδή η συχνότητα του σήματος εξόδου από τον πολυπλέκτη, αλλάζει κάθε ένα δευτερόλεπτο. Τα ρολόγια αυτά έχουν συχνότητα 160 και 80 Mhz, και παράγονται από το PLL. Ο τρόπος παραγωγής ρολογιών από το PLL, γίνεται σαφής στις γραμμές 117-123, όπου η συχνότητα του ρολογιού αναφοράς πολλαπλασιάζεται με 40, παράγοντας ένα ρολόι 1.6 Ghz. Διαιρώντας δια δέκα (121) το ρολόι στην έξοδο 1 του PLL φτάνει στα 160 Mhz. Ανάλογα παράγονται και τα υπόλοιπα ρολόγια. Τέλος, το σήμα του pll_locked, που υποδεικνύει ουσιαστικά ότι το PLL έχει σταθεροποιηθεί (βλ. υποεν. 6.1.2), διοχετεύεται σε έναν αναστροφέα (52). Το αποτέλεσμα αυτού (rst_i) είναι το εσωτερικό reset, το οποίο δρομολογείται στον μετρητή και στον serializer. Ο λόγος που έχει σχεδιαστεί έτσι το reset, είναι για να ενεργοποιείται μόνο όταν υπάρχουν αστάθειες στις συχνότητες των ρολογιών που παράγει το PLL, γεγονός ανεπιθύμητο. Το reset είναι μηδέν και όλα βαίνουν ομαλά, μόνο όταν το PLL συνθέτει ρολόγια καλής ποιότητας. Τέλος, το αποτέλεσμα του serialization, διοχετεύεται σε έναν διαφορικό buffer εξόδου (γραμμές 81-89), ο οποίος λειτουργεί με επίπεδα LVDS 2.5 V. Οι έξοδοι του buffer είναι τα output pins του FPGA.

Το σχεδιάγραμμα RTL του κυκλώματος απεικονίζεται στο παρακάτω Σχήμα, ενώ η

προσομοίωση του ISim δείχνει ότι πράγματι το κύκλωμα ανταποκρίνεται στις προβλέψεις.

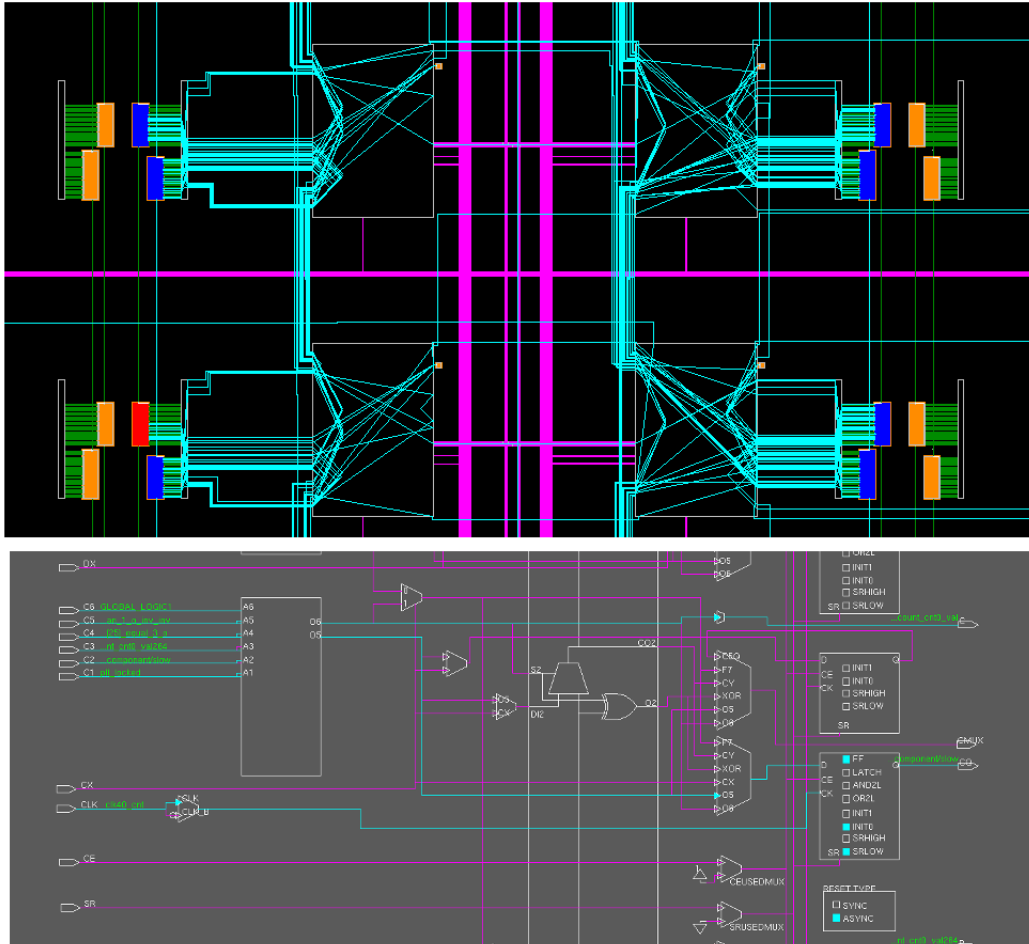


Σχήμα Β'.3.1: Αναπαράσταση του top_level του serializer δύο ταχυτήτων. Στο κάτω μέρος φαίνεται το PLL, ενώ τριγύρω φαίνονται οι buffers. Πάνω φαίνεται ο πολυπλέκτης των ρολογιών και το μαύρο κουτί που αναπαριστά τον serializer. Αριστερά στην άκρη φαίνεται το pin του ρολογιού εισόδου, και δεξιά τα διαφορεικά σήματα εξόδου που μεταδίδουν σειριακά το διάνυσμα vector του serializer.



Σχήμα Β'.3.2: Προσομοίωση του κυκλώματος από το iSim. Τη στιγμή που το σήμα slow γίνεται μηδέν, γεγονός που σημαίνει ότι ο μετρητής cnt0 του counter μεγιστοποιήθηκε, αλλάζει η έξοδος του πολυπλέκτη των ρολογιών με αποτέλεσμα να αλλάξει η συχνότητα του ρολογιού αναφοράς του serializer. Φαίνεται ξεκάθαρα ότι στην αλλαγή της κατάστασης, τα tx_p και tx_n, ξεκινάνε να μεταδίδουν τα σήματα με πιο πυκνές μεταβάσεις, γεγονός που υποδεικνύει αύξηση της συχνότητας του serialization.

Προκειμένου να υλοποιηθεί το παρών κύκλωμα, ο χρήστης οφείλει να θέσει τα constraints, δηλαδή να ορίσει σε ποια ακριβώς pins του FPGA, θα αντιστοιχούν τα ports clk40_in, tx_p και tx_n. Μετά από αυτό, γίνεται η σύνθεση του κώδικα, και σύμφωνα με την ενότητα 6.3, το κύκλωμα αντιστοιχίζεται σε LUT και ενεργοποιημένα καλώδια μέσα στο FPGA. Η αναπαράσταση των κυκλωμάτων αυτών, γίνεται μέσω του FPGA Editor. Φωτογραφίες των LUT/slice παρατίθεται παρακάτω:



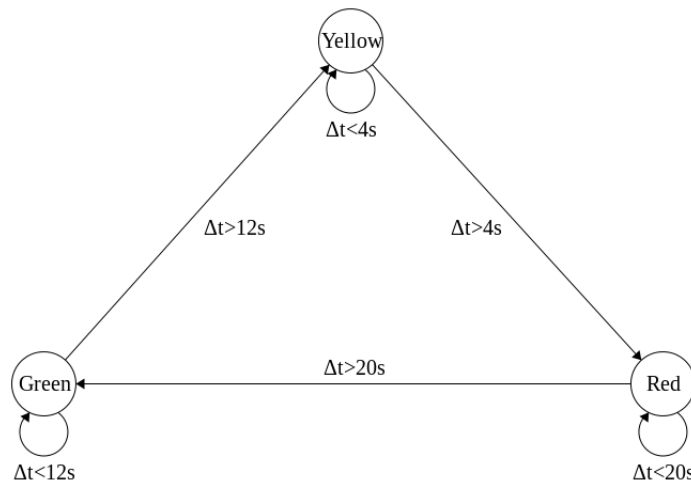
Σχήμα Β'.3.3: Φωτογραφίες από τον FPGA Editor. Πάνω απεικονίζονται τα slice που υλοποιούν τους μετρητές. Φαίνονται επίσης οι διασυνδέσεις με τους switch matrix, και τα καλώδια που καταλήγουν στα switch boxes. Κάθετα και οριζόντια στο κέντρο, βρίσκεται το global routing. Κάτω φαίνεται με λεπτομέρεια ένα slice. Τα LUT, οι flip-flop και οι πολυπλέκτες δέχονται εξωτερικά σήματα (χρονισμού και ελέγχου) που εισέρχονται στο slice από τα pins αριστερά.

B.4 Finite-State Machines

Μετά τη σχετική εμβάθυνση στη VHDL, ήρθε η ώρα να εισαχθεί και μία πολύ σημαντική κατηγορία κυκλωμάτων, που χρησιμοποιούνται κατά κόρον στα FPGA. Πρόκειται για την έννοια της *Μηχανής Πεπερασμένων Καταστάσεων* (*Finite-State Machine (FSM)*). Η FSM δεν είναι τίποτε παραπάνω από μία βολική και εύκολη μοντελοποίηση ακολουθιακών κυκλωμάτων. Πρόκειται ουσιαστικά για μία αφηρημένη έννοια, η οποία ορίζεται από ένα πεπερασμένο πλήθος *καταστάσεων* (*States*). Επίσης μία FSM, όπως κάθε κύκλωμα που περιγράφεται από κάποια HDL, έχει σήματα εισόδου και εξόδου. Τέλος, υπάρχει και ένα σήμα χρονισμού. Σε κάθε στιγμή, η μηχανή βρίσκεται σε μία και μοναδική κατάσταση. Σε κάθε state, οι συνθήκες μετάβασης από την υπάρχουσα κατάσταση σε μία άλλη, είναι πλήρως καθορισμένες, και εξαρτώνται από τα σήματα εισόδου. Σε κάθε tick του ρολογιού, η FSM αναγνωρίζει την κατάσταση στην οποία είναι, και μελετάει τις συνθήκες μετάβασης για κάποια άλλη κατάσταση. Εάν οι συνθήκες για τη μετάβαση σε μία άλλη κατάσταση ικανοποιούνται, τότε στο επόμενο tick, αυτή θα είναι η νέα κατάσταση της FSM. Αν δεν ικανοποιείται ικανοποιητικός αριθμός συνθηκών, η FSM παραμένει ως έχει. Τονίζεται ότι η FSM είναι ουσιαστικά ένα ακολουθιακό κύκλωμα, αφού η κατάσταση της αλλάζει μόνο τη στιγμή που φτάνουν οι παλμοί χρονισμού (στα rising edges δηλαδή) [26].

Ένα απλό παράδειγμα FSM, θα είναι ένας microcontroller ο οποίος ελέγχει το πως λειτουργεί ένα φανάρι. Έστω ότι αυτό το φανάρι, κρατάει το κόκκινο φως αναμμένο για $t_r = 20\text{ s}$, το πράσινο για $t_g = 12\text{ s}$, και το κίτρινο για $t_y = 4\text{ s}$. Η FSM λοιπόν, θα περιγράφεται από τρεις καταστάσεις, μία για κάθε χρώμα. Αν υλοποιηθεί ένα τέτοιο μοντέλο σε VHDL, τότε οι συνθήκες μετάβασης από το ένα χρώμα στο άλλο, θα δίνονται από έναν μετρητή που θα υπολογίζει χρονική διάρκεια (σαν τον κώδικα B.4, counter.vhd). Η FSM εκτός από τα σήματα που θα ανάβουν τα αντίστοιχα χρώματα στο φανάρι, θα στέλνει και σε κάθε αλλαγή κατάστασης ένα σήμα στο μετρητή για να μηδενίσει. Ο κώδικας VHDL ενός τέτοιου μοντέλου παρατίθεται παρακάτω, μαζί με ένα διάγραμμα² της FSM.

²Όλα τα διαγράμματα κατάστασης των FSM έχουν γίνει με τη βοήθεια ελεύθερου λογισμικού από το site: <http://madebyevan.com/fsm/>



Σχήμα Β'.4.1: State Diagram μίας FSM που ελέγχει τα χρώματα σε ένα φανάρι.

VHDL Code Β'.7: trafficLightFSM.vhd

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity traffic_light is
5
6     port(
7         clk      : in  std_logic;
8         t4s      : in  std_logic;
9         t12s     : in  std_logic;
10        t20s     : in  std_logic;
11        rst_cnt  : out std_logic;
12        red      : out std_logic;
13        yellow   : out std_logic;
14        green    : out std_logic);
15 end traffic_light;
16
17 architecture Behavioral of traffic_light is
18
19     type state is (red_state , green_state , yellow_state);
20     signal stateFSM : state := red_state;
21
22 begin
23
24 process (clk , t4s , t12s , t20s)

```

```
25     begin
26         if (rising_edge (clk)) then —clk
27             case stateFSM is
28                 -----
29                 when red_state =>
30                     red     <= '1';
31                     yellow <= '0';
32                     green  <= '0';
33                     rst_cnt <= '0';
34                 if (t20s = '1') then
35                     rst_cnt <= '1';
36                     stateFSM <= green_state;
37                 else null;
38                 end if;
39                 -----
40                 when green_state =>
41                     red     <= '0';
42                     yellow <= '0';
43                     green  <= '1';
44                     rst_cnt <= '0';
45                 if (t12s = '1') then
46                     rst_cnt <= '1';
47                     stateFSM <= yellow_state;
48                 else null;
49                 end if;
50                 -----
51                 when yellow_state =>
52                     red     <= '0';
53                     yellow <= '1';
54                     green  <= '0';
55                     rst_cnt <= '0';
56                 if (t4s = '1') then
57                     rst_cnt <= '1';
58                     stateFSM <= red_state;
59                 else null;
60                 end if;
61
62                 when others => null;
63     end case;
64
65 end if; —clk
```

```

66
67 end process ;
68
69
70
71 end Behavioral ;

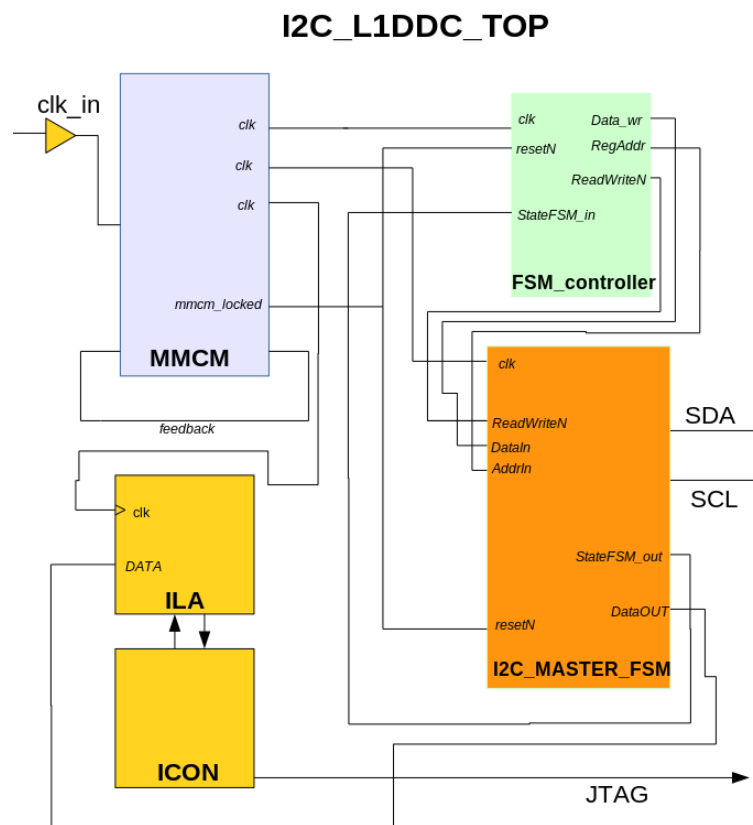
```

Εξωτερικά της FSM, υπάρχει ένας μετρητής, του οποίου ο κώδικας εδώ απλά θα περιγραφεί επιγραμματικά. Θα είναι ένα ξεχωριστό component, όπου στην είσοδο θα έχει ένα ρολόι με το οποίο θα αυξάνει την τιμή του, ένα σήμα `rst_cnt` που θα έρχεται από την FSM και θα μηδενίζει τον μετρητή, και τρία σήματα εξόδου. Το κάθε σήμα εξόδου (`t4s`, `t12s`, `t20s`), θα γίνεται ένα όταν ο μετρητής αυξήσει αρκετά την αντίστοιχη τιμή της μεταβλητής στη μνήμη του, ώστε να έχει μετρήσει χρόνο τεσσάρων, δώδεκα, ή είκοσι δευτερολέπτων³. Η FSM τώρα, ορίζεται από τρεις καταστάσεις (γραμμές 19,20), και σε κάθε χτύπο του ρολογιού (γραμμή 26), ελέγχει τις συνθήκες εισόδου. Ξεκινάει από την κατάσταση του κόκκινου (`red_state`) όπου ανάβει το κόκκινο φανάρι και αφήνει τον εξωτερικό μετρητή να "τρέξει" (29-33). Αν ο μετρητής μετρήσει είκοσι δεύτερα (34), τότε η FSM τον μηδενίζει και μεταπίπτει στην πράσινη κατάσταση (35,36). Πριν γίνει αυτό, η κατάσταση μένει ίδια (37, `else => null;`). Στην πράσινη κατάσταση ξεκινάει πάλι ο μετρητής (44), και ενεργοποιούνται τα αντίστοιχα σήματα. Έτσι η FSM κάνει κύκλους μεταξύ των καταστάσεων της σε αυστηρά ορισμένα χρονικά πλαίσια, σύμφωνα πάντα με το μοντέλο που έχει σχεδιαστεί. Οι FSM παίζουν σημαντικό ρόλο σε πολλά κομμάτια του firmware της L1DDC γιατί πολύ απλά, είναι πολύ εύκολο να προσομοιάσει κανείς ένα ακολουθιακό κύκλωμα με μία FSM.

³Αν για παράδειγμα το ρολόι αναφοράς του μετρητή έχει συχνότητα $f_{clk} = 1 \text{ kHz}$, τότε για να μετρήσει 4 δευτερόλεπτα, πρέπει ο μετρητής να φτάσει στην τιμή $cnt = 4000$.

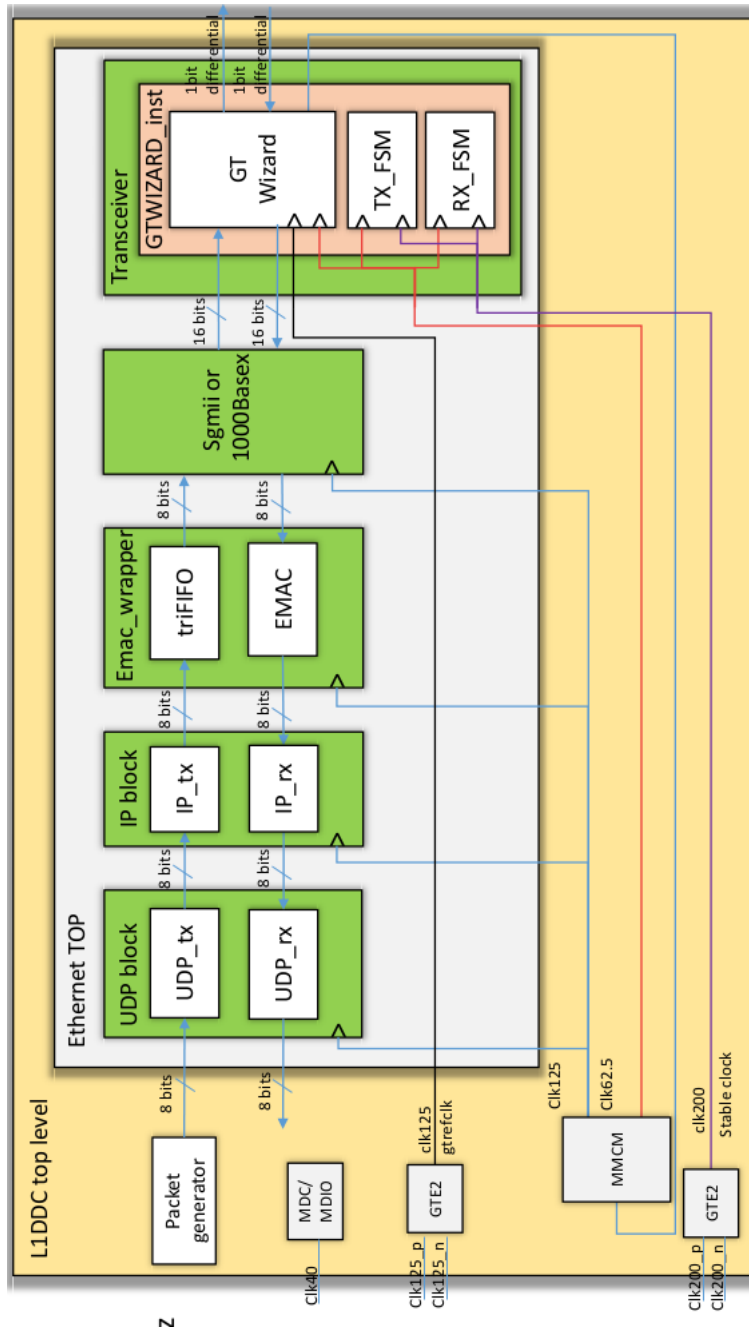
Block Diagrams

I²C Block Diagram



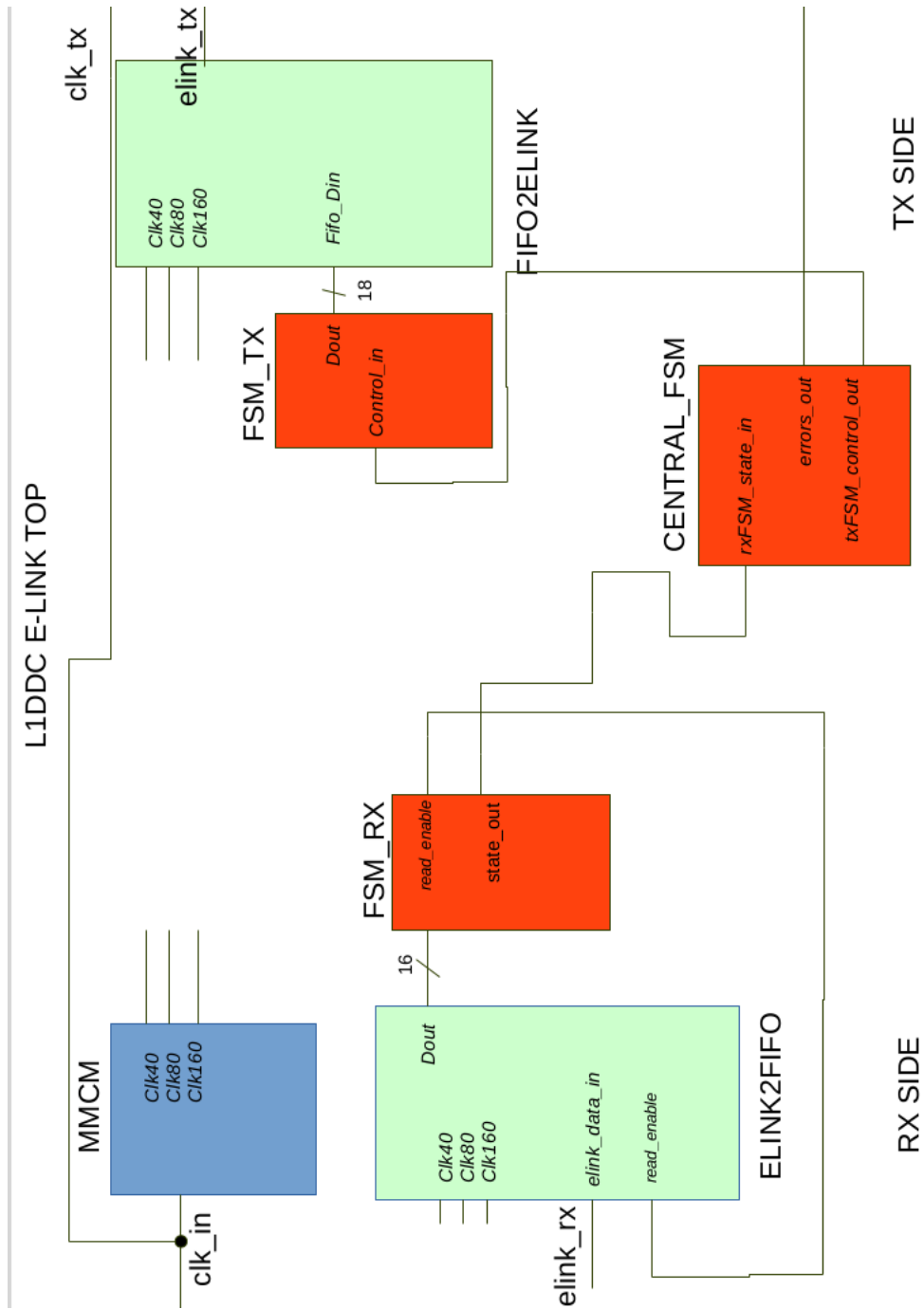
Σχήμα Γ'.0.1

Ethernet Block Diagram



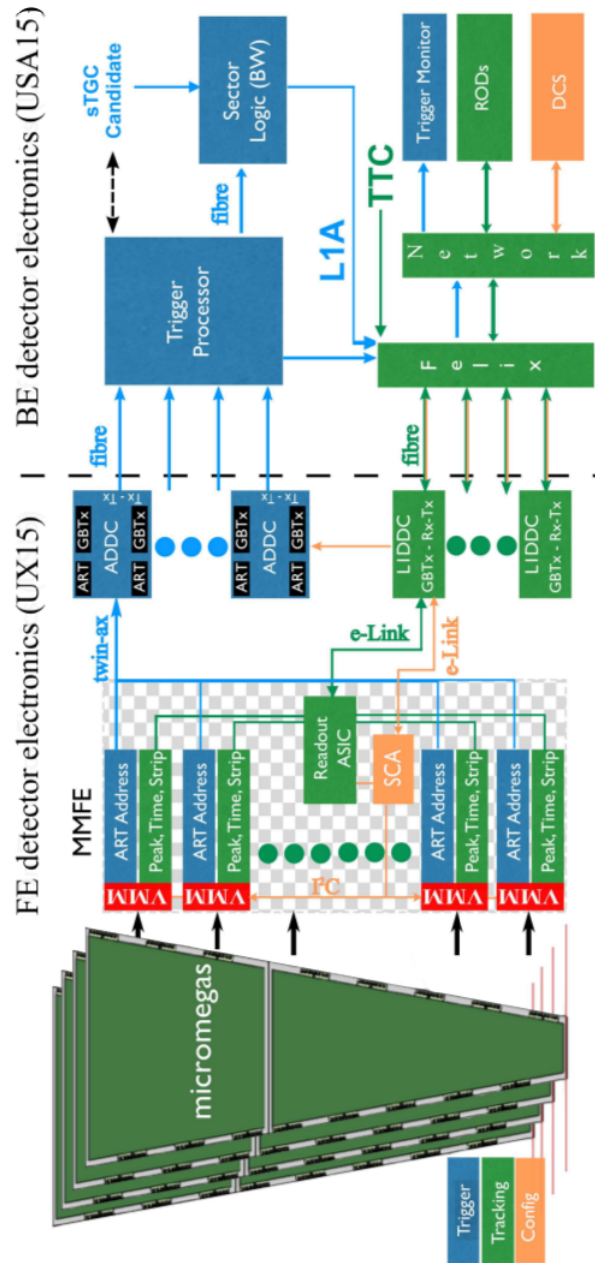
Σχήμα Γ.0.2

E-Link Block Diagram (for Debugging)



Σχήμα Γ'.0.3

NSW Electronics Overview



Σχήμα Γ.0.4

Δείγματα Firmware σε VHDL

Σε αυτό το Παράρτημα, παρατίθενται μερικά δείγματα κώδικα σε VHDL, τα οποία υλοποιούν την διαδικασία ελέγχου (debugging) των e-links. Σχηματική αναπαράσταση σε block diagram του firmware αυτού μπορεί να βρεθεί στο αντίστοιχο Παράρτημα.

VHDL Code Δ'1: central_fsm_11ddc

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity central_fsm_11ddc is
5
6
7 generic(cnt_max : natural := 1_600_000_000);
8 —for clk160, this is 10 seconds.
9
10 port(
11     clk           : in std_logic;
12     rst           : in std_logic;
13     header_sig    : in std_logic;
14     packet_sig    : in std_logic;
15     error_header  : out std_logic;
16     done_header   : out std_logic;
17     error_packet  : out std_logic;
18     done_packet   : out std_logic);
19 end central_fsm_11ddc;
20
21 architecture Behavioral of central_fsm_11ddc is
```

```
22
23 type state is (header_check , header_done , header_error ,
24                packet_check , packet_done , packet_error);
25
26     signal stateFSM : state := header_check;
27
28 begin
29
30 process(clk , rst , header_sig , packet_sig)
31
32     variable cnt : natural range 0 to cnt_max := 0;
33
34     begin
35
36         if(rst = '1')then
37             stateFSM <= header_check;
38             error_header <= '0';
39             done_header <= '0';
40             error_packet <= '0';
41             done_packet <= '0';
42             cnt := 0;
43
44         elsif(rising_edge(clk))then
45
46             case stateFSM is
47
48                 when header_check =>
49                     if(header_sig = '0')then
50                         if(cnt < cnt_max)then
51                             cnt := cnt + 1;
52                         elsif(cnt = cnt_max)then
53                             stateFSM <= header_error;
54                             cnt := 0;
55                         else null;
56                     end if;
57                     elsif(header_sig = '1')then
58                         cnt := 0;
59                         stateFSM <= header_done;
60                     else null;
61                     end if;
62
```

```
63     when header_done =>
64         error_header <= '0';
65         done_header  <= '1';
66         stateFSM <= packet_check;
67
68         -----
69     when header_error =>
70         error_header <= '1';
71         done_header  <= '0';
72
73         -----
74     when packet_check =>
75         if (packet_sig = '0') then
76             if (cnt < cnt_max) then
77                 cnt := cnt + 1;
78             elsif (cnt = cnt_max) then
79                 stateFSM <= packet_error;
80                 cnt := 0;
81             else null;
82             end if;
83         elsif (packet_sig = '1') then
84             cnt := 0;
85             stateFSM <= packet_done;
86             else null;
87             end if;
88
89         -----
90     when packet_done =>
91         error_packet <= '0';
92         done_packet  <= '1';
93
94         -----
95     when packet_error =>
96         error_packet <= '1';
97         done_packet  <= '0';
98
99         -----
100     end case;
101
102     end if;
103
104 end process;
105
106 end Behavioral;
```

VHDL Code Δ.2: packet_sender_1lddc

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity packet_sender_1lddc is
5
6     generic(cnt_max : natural := 1_000;
7             init_max : natural := 2_000;
8             array_len : natural := 3);
9
10    port(
11        clk          : in  std_logic;
12        rst          : in  std_logic;
13        send_ack     : in  std_logic;
14        write_enable : out std_logic;
15        state_out    : out std_logic_vector(1 downto 0);
16        fifo_empty   : in  std_logic;
17        doutFifo     : out std_logic_vector(17 downto 0));
18
19
20 end packet_sender_1lddc;
21
22 architecture Behavioral of packet_sender_1lddc is
23
24 _____
25 _____
26 _____
27 signal MOP : std_logic_vector(1 downto 0) := "00";
28 signal SOP : std_logic_vector(1 downto 0) := "10";
29 signal EOP : std_logic_vector(1 downto 0) := "01";
30
31 type type1 is array (0 to array_len) of std_logic_vector(17 downto 0);
32
33 signal byte00 : std_logic_vector(7 downto 0) := (others => '0');
34 signal byteA1 : std_logic_vector(7 downto 0) := X"A1";
35 signal byteB1 : std_logic_vector(7 downto 0) := X"B1";
36 signal byteC1 : std_logic_vector(7 downto 0) := X"C1";
37 signal byteE1 : std_logic_vector(7 downto 0) := X"E1";
38
39
40
```

```
41 signal input_array : type1 := (SOP & byte00 & byteA1 ,
42                               MOP & byteB1 & byteC1 ,
43                               MOP & byteD1 & byteE1 ,
44                               EOP & byte00 & byte00 ); --5 bytes
45
46 _____
47 _____
48 _____
49
50 type state is (idle , filling , hold_one_cycle , done);
51 signal current_state : state;
52
53 begin
54
55   process(clk , rst , fifo_empty)
56
57     variable index : natural range 0 to array_len := 0;
58     variable cnt   : natural range 0 to cnt_max := 0;
59     variable init_cnt : natural range 0 to init_max := 0;
60
61     begin
62       if(rst = '1')then
63         current_state <= idle;
64         init_cnt := 0;
65       elsif(rising_edge(clk))then
66
67         case current_state is
68           _____
69         when idle =>
70           write_enable <= '0';
71           state_out <= "00";
72
73           if(fifo_empty = '1' and send_ack = '1')then
74             if(init_cnt < init_max)then
75               init_cnt := init_cnt + 1;
76             elsif(init_cnt = init_max)then
77               current_state <= filling;
78             else null;
79             end if;
80           else null;
81           end if;
```

```
82 | _____
83 | when filling =>
84 | if(index < array_len)then
85 |   write_enable <= '1';
86 |   state_out <= "01";
87 |   doutFifo <= input_array(index);
88 |   index := index + 1;
89 |   elsif(index = array_len)then
90 |     doutFifo <= input_array(index);
91 |     index := 0;
92 |     current_state <= hold_one_cycle;
93 |   else null;
94 | end if;
95 | _____
96 | when hold_one_cycle =>
97 |   state_out <= "10";
98 |   doutFifo <= (others => '0');
99 |   write_enable <= '0';
100 |   current_state <= done;
101 | _____
102 | when done =>
103 |   state_out <= "11";
104 |
105 |   if(fifo_empty = '1')then
106 |     if(cnt < cnt_max)then
107 |       cnt := cnt + 1;
108 |     elsif(cnt = cnt_max)then
109 |       cnt := 0;
110 |       current_state <= idle;
111 |     else null;
112 |   end if;
113 |   else null;
114 | end if;
115 |
116 | end case;
117 |
118 | end if;
119 |
120 | end process;
121 |
122 | end Behavioral;
```

VHDL Code Δ'.3: fsm_packet_checker_fe

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.NUMERIC_STD.ALL;
4
5 entity fsm_packet_checker_fe is
6
7     port(
8         clk          : in std_logic;
9         rst          : in std_logic;
10        packet_in   : in std_logic_vector(15 downto 0);
11        header_sig   : out std_logic;
12        done_sig     : out std_logic);
13
14 end fsm_packet_checker_fe;
15
16 architecture Behavioral of fsm_packet_checker_fe is
17
18     type state is (idle , header , packet_1 ,
19                  packet_2 , trailer , doneFSM, errorFSM);
20
21     signal stateFSM : state;
22
23     signal l_packet : std_logic_vector(7 downto 0);
24     signal r_packet : std_logic_vector(7 downto 0);
25
26 begin
27
28     process(clk , rst , packet_in)
29     begin
30         if(rst = '1') then
31             stateFSM <= idle;
32             done_sig <= '0';
33             header_sig <= '0';
34
35         elsif(rising_edge(clk)) then
36
37             l_packet <= packet_in(15 downto 8);
```

```
38     r_packet <= packet_in(7 downto 0);
39
40     case stateFSM is
41     -----
42     when idle =>
43         if (l_packet = X"AB" and r_packet = X"CD") then
44             stateFSM <= header;
45         else null;
46         end if;
47     -----
48     when header =>
49         header_sig <= '1';
50         if (l_packet = X"AB" and r_packet = X"CD") then null; --discard
51         elsif (r_packet = X"A1" and l_packet = X"B1") then
52             stateFSM <= packet_1;
53         else
54             stateFSM <= errorFSM;
55         end if;
56     -----
57     when packet_1 =>
58         if (r_packet = X"A1" and l_packet = X"B1") then null; --discard
59         elsif (r_packet = X"C1" and l_packet = X"D1") then
60             stateFSM <= packet_2;
61         else
62             stateFSM <= errorFSM;
63         end if;
64     -----
65     when packet_2 =>
66         if (r_packet = X"C1" and l_packet = X"D1") then null; --discard
67         elsif (r_packet = X"E1" and l_packet = X"00") then
68             stateFSM <= trailer;
69         else
70             stateFSM <= errorFSM;
71         end if;
72     -----
73     when trailer =>
74         if (r_packet = X"E1" and l_packet = X"00") then null; --discard
75         elsif (l_packet = X"60" and r_packet = X"05") then
76
77             stateFSM <= doneFSM; -- correct trailer. Done.
78         else
```



```
79         stateFSM <= errorFSM;
80     end if;
81     -----
82     when doneFSM =>
83         done_sig <= '1';
84     -----
85     when errorFSM =>
86         done_sig <= '0';
87     -----
88
89     end case;
90
91 end if;
92
93 end process;
94
95 end Behavioral;
```


Βιβλιογραφία

- [1] David Griffiths - *Introduction to Elementary Particles, 2nd Revised Edition*. Wiley-VCH, 2008.
- [2] Chinese Physical Society - *Review of Particle Physics*. Volume 38, Number 9, September 2014.
- [3] Claus Grupen, Boris Shwartz - *Particle Detectors, 2nd Revised Edition*. Cambridge University Press, 2008.
- [4] William R. Leo - *Techniques for Nuclear and Particle Physics Experiments, 2nd Revised Edition*. Springer-Verlag, 1994.
- [5] Glenn F. Knoll - *Radiation Detection and Measurement, 4th Edition*. Wiley, 2010.
- [6] CERN Homepage -
<http://home.web.cern.ch/>
- [7] ATLAS Experiment Homepage -
<http://atlas.ch>.
- [8] ATLAS Collaboration - *ATLAS New Small Wheel Technical Design Report*. CERN-LHCC-2013-006, ATLAS-TDR-020, July 2013
- [9] Georgios A. Iakovidis - *Research and Development in Micromegas Detector for the ATLAS Upgrade*. Ph.D Thesis, NTU Athens, October 2014.
- [10] T. Alexopoulos, C. Bakalis, P. Gkoutoumis, G. Iakovidis, A. Koulouris - *Level-1 Data Driver Card Design Review Report*. ©2015 CERN for the benefit of the ATLAS collaboration.

- [11] T. Alexopoulos, P. Gkountoumis, G. Iakovidis, A. Koulouris - *Level-1 Data Driver Card of the ATLAS New Small Wheel Upgrade*. 2015, 4th International conference on Modern Circuits and Systems Technologies.
- [12] *Preliminary MMFE-8 Specification*. ATLAS Collaboration, 2015.
- [13] G. De Geronimo, J. Fried, S. Li, J. Metcalfe, N. Nambiar, E. Vernon, V. Polychronakos - *VMMI - An ASIC for Micropattern Detectors*.
- [14] *VMM2 (ic134) Architecture and Functionality*. rev. 16, 4/21/2015, ATLAS Collaboration.
- [15] Georgios A. Iakovidis - *VMM - An ASIC for Micropattern Detectors*. MPGD 2015 - Trieste, Italy.
- [16] T. Alexopoulos, C. Bakalis, G. Boukli, P. Gkountoumis, G. Iakovidis, P. Moschovakos, V. Polychronakos - *LIDDC Firmware Status*. 02/10/2015, ATLAS muon week.
- [17] GBT Project - *GBTx Manual*. v0.10 DRAFT, 11/08/2015.
- [18] P. Horowitz, W. Hill - *The Art of Electronics, 3rd Edition*. Cambridge University Press, 2015
- [19] James F. Kurose, Keith W. Ross - *Computer Networking, A Top-Down Approach, 6th Edition*. Pearson Education, 2013.
- [20] A. X. Widmer, P.A. Franaszek - *A DC-Balanced, Partitioned Block, 8B/10B Transmission Code*. IBM Journal of research and development. Volume 27, Number 5, September 1983.
- [21] S. Hauck, A. DeHon - *Reconfigurable Computing. The Theory and Practice of FPGA-Based Computation*. Elsevier, 2008.
- [22] V. Betz, J. Rose, A. Marquardt - Kluwer Academic Publishers, 2000.
- [23] I. Kuon, R. Tessier, J. Rose - *FPGA Architecture: Survey and Challenges*. Foundations and Trends in Electronics Design Automation, vol. 2, no 2, pp. 135-253, 2007.

- [24] H. Parvez, H. Mehrez - *Application-Specific, Mesh-Based, Heterogenous FPGA Architectures*. Springer, 2011.
- [25] Volnei A. Pedroni - *Circuit Design and Simulation with VHDL, 2nd Edition*. The MIT Press, 2010.
- [26] Volnei A. Pedroni - *Finite State Machines in Hardware. Theory and Design (with VHDL and SystemVerilog)*. The MIT Press, 2013.
- [27] Xilinx® *User Guides on 7-Series FPGAs. UG470-475*. Xilinx® Corporation.
- [28] Xilinx® *Product Guides on EMAC and PCS/PMA, and User Guides on 7-Series GTP Transceivers. PG047, PG051, UG482*. Xilinx® Corporation.
- [29] IEEE Computer Society - *IEEE Standard for Ethernet, IEEE Std 802.3™-2012*. Copyright ©2012 by The Institute of Electrical and Electronics Engineers, Inc.
- [30] Xilinx® Home Page
<http://www.xilinx.com>
- [31] OpenCores.org
<http://opencores.org/>