

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών  
Εργαστήριο Μικροεπεξεργαστών και Υλικού

**Πειραματική Μελέτη Σειριακής Εισόδου/Εξόδου  
Υψηλών Ταχυτήτων σε Ενσωματωμένα Συστήματα  
FPGA**

**ΣΠΥΡΟΠΟΥΛΟΣ ΚΥΡΙΑΚΟΣ**

Εξεταστική Επιτροπή :

Αναπληρωτής Καθηγητής Πνευματικάτος Διονύσιος (Επιβλέπων)  
Καθηγητής Δόλλας Απόστολος  
Καθηγητής Καλαϊτζάκης Κωνσταντίνος

Νοέμβριος 2008, Χανιά

# ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΚΕΦΑΛΑΙΟ 1.....</b>	<b>4</b>
<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>4</b>
1.1 Λίγα λόγια για τη σειριακή μεταφορά δεδομένων.....	4
1.1.1 Ιστορία και θεωρία σειριακής μεταφοράς δεδομένων.....	4
1.1.2 Πλεονεκτήματα –Μειονεκτήματα – Προκλήσεις.....	6
1.1.3 Εφαρμογές της σειριακής μεταφοράς δεδομένων.....	8
1.1.4 Μέχρι που μπορεί να φτάσει - Δυνατότητες.....	10
1.2 ΣΚΟΠΟΣ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ.....	10
<b>ΚΕΦΑΛΑΙΟ 2.....</b>	<b>13</b>
<b>ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....</b>	<b>13</b>
2.1 ΓΝΩΡΙΜΙΑ ΜΕ ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΑΝΑΠΤΥΞΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	13
2.1.1 Επισκόπηση.....	13
2.1.2 Λίγα λόγια για την XUP Virtex II Pro FPGA.....	13
2.2 ΑΝΑΛΥΣΗ ΤΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΚΑΙ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ ROCKET I/O.....	19
2.2.1 Επισκόπηση.....	19
2.2.2 Λίγα λόγια για το Rocket I/O.....	19
2.2.3 Θέματα Ψηφιακού Σχεδιασμού (Digital Design Considerations).....	22
2.2.4 Θέματα Αναλογικού Σχεδιασμού (Analog Design Considerations).....	38
2.3 ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΠΕΔΟΥ ΔΙΑΣΥΝΔΕΣΗΣ AURORA.....	42
2.3.1 Επισκόπηση.....	42
2.3.2 Γενικά για τα επίπεδα δικτύου.....	42
2.3.3 Επισκόπηση πρωτοκόλλου επιπέδου διασύνδεσης Aurora.....	42
<b>ΚΕΦΑΛΑΙΟ 3.....</b>	<b>62</b>
<b>ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ.....</b>	<b>62</b>
3.1 ΕΠΙΣΚΟΠΗΣΗ.....	62
3.2 ΣΧΕΔΙΑΣΗ ΧΩΡΙΣ ΧΡΗΣΗ ΤΟΥ POWERPC.....	62
3.2.1 Το Bus Functional μοντέλο για πρωτόκολλο επιπέδου διασύνδεσης Aurora.....	62
3.2.2 Υλοποίηση πρωτοκόλλου Aurora σε ένα XUP με Iorback.....	63
3.2.3 Υλοποίηση πρωτοκόλλου Aurora σε δύο XUP.....	65
3.3 ΣΧΕΔΙΑΣΗ ΜΕ ΧΡΗΣΗ ΤΟΥ POWERPC.....	66
3.3.1 Λίγα λόγια για το Bit Error Rate Tester.....	66
3.3.2 Υλοποίηση πρωτοκόλλου Aurora με χρήση επεξεργαστή σε ένα XUP.....	67
3.3.2 Υλοποίηση πρωτοκόλλου Aurora με χρήση επεξεργαστή σε δύο XUP.....	82
<b>ΚΕΦΑΛΑΙΟ 4.....</b>	<b>84</b>
<b>ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ.....</b>	<b>84</b>
4.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ.....	84
4.1.1 Γενικά.....	84
4.2 ΠΡΟΔΙΑΓΡΑΦΕΣ ΠΡΩΤΟΚΟΛΛΟΥ.....	87
4.3 ΒΙΒΛΙΟΘΗΚΗ API.....	88
4.3.1 Διεπαφή με το χρήστη.....	88

4.4 MAIN (ΚΥΡΙΩΣ ΕΦΑΡΜΟΓΗ).....	90
4.4.1 Για <i>Ioorbask</i> σε μια πλακέτα XUP.....	90
4.4.2 Για μεταφορά δεδομένων από μια πλακέτα XUP σε μια άλλη.....	91
<b>ΚΕΦΑΛΑΙΟ 5.....</b>	<b>93</b>
<b>ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ ΣΧΕΔΙΑΣΗΣ.....</b>	<b>93</b>
5.1 ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΚΑΙ ΠΡΟΕΤΟΙΜΑΣΙΑ ΓΙΑ ΤΟ ΠΕΙΡΑΜΑ.....	93
5.1.1 Προϋποθέσεις.....	93
5.2 ΔΟΚΙΜΕΣ ΑΝΑΛΥΤΙΚΑ.....	93
5.2.1 Έλεγχος ορθότητας λειτουργίας σημάτων κατάστασης .....	93
5.2.2 Χαρακτηριστικά διαφορετικών σχεδιάσεων.....	94
5.2.3 Δοκιμή αποστολής και λήψης δεδομένων με χρήση των συναρτήσεων <i>send receive</i> με αντιγραφή των δεδομένων μέσω DMA.....	96
5.2.4 Δοκιμή αποστολής και λήψης δεδομένων με χρήση των συναρτήσεων <i>send receive</i> με αντιγραφή των δεδομένων μέσω διαύλου PLB.....	97
5.2.5 Δοκιμή αποστολής και λήψης απεριόριστου αριθμού δεδομένων απρόσκοπτα με χρήση των συναρτήσεων <i>send receive</i> και έλεγχο των FIFO.....	97
5.2.6 Μέτρηση του ρυθμού μεταφοράς δεδομένων με χρήση <i>custom component</i> για αυτό το σκοπό είτε με χρήση <i>timer</i> του EDK.....	98
5.2.7 Επιβεβαίωση λήψης σταλμένων δεδομένων και μέτρηση ρυθμού σφαλμάτων.....	98
5.3 ΠΕΡΙΟΡΙΣΜΟΙ.....	99
5.4 ΣΤΑΤΙΣΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ, ΣΥΜΠΕΡΑΣΜΑΤΑ.....	99
<b>ΚΕΦΑΛΑΙΟ 6.....</b>	<b>101</b>
<b>ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ.....</b>	<b>101</b>
<b>ΠΑΡΑΡΤΗΜΑ.....</b>	<b>102</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>106</b>

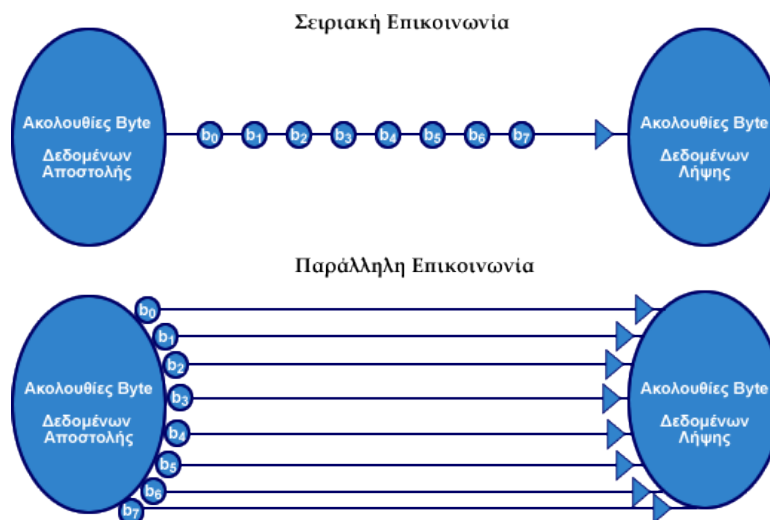
# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

### 1.1 Λίγα λόγια για τη σειριακή μεταφορά δεδομένων

#### 1.1.1 Ιστορία και θεωρία σειριακής μεταφοράς δεδομένων

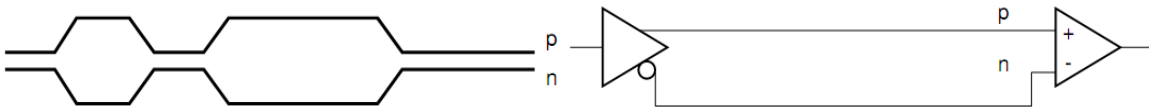
Η παράλληλη επικοινωνία αποτελούσε παραδοσιακά τον κυρίαρχο τρόπο ανταλλαγής δεδομένων για τις πλακέτες τυπωμένων κυκλωμάτων (PCB). Ωστόσο σε πρώιμο στάδιο τα προβλήματα ευθυγράμμισης σήματος αποτελούσαν τροχοπέδη στην περαιτέρω ανάπτυξη αυτής της αρχιτεκτονικής επικοινωνιών. Έτσι η σειριακή επικοινωνία ήταν πιο διαδεδομένη όσον αφορά την ανταλλαγή δεδομένων από συσκευή-σε-συσκευή με πιο αντιπροσωπευτικό παράδειγμα εκτυπωτές σειριακής θύρας στους πρώτους υπολογιστές. Έπειτα τα προβλήματα ευθυγράμμισης σήματος λύθηκαν και νομοτελειακά αναπτύχθηκαν ακόμα περισσότερο οι παράλληλες επικοινωνίες. Χαρακτηριστικά εμφανίστηκαν οι εκτυπωτές παράλληλης θύρας και διαμορφώθηκαν τεχνολογίες όπως Industry-Standard Architecture (ISA), Extended Industry-Standard Architecture (EISA), Small Computer Systems Interface (SCSI), Peripheral Component Interconnect (PCI) και Personal Computer Memory Card Industry Association (PCMCIA). Τότε εισήχθη ο fractional phase detector, τεχνολογία που εκτόξευσε την ταχύτητα της σειριακής γραμμής μετάδοσης στις τάξεις των gigabit. Οι δυο αρχιτεκτονικές επικοινωνιών συνεχίζουν να συνυπάρχουν μέχρι και σήμερα όπου φαντάζει ενδεχομένως πιο ανταγωνιστική η σειριακή επικοινωνία για τις πλακέτες τυπωμένων κυκλωμάτων (PCB).



Σχήμα 1-1 : Παράλληλη και Σειριακή Επικοινωνία

Η επικοινωνία πλακετών τυπωμένων κυκλωμάτων (PCB), ταυτίζεται με την έννοια της Εισόδου/Εξόδου προς και από αυτές. Είναι λοιπόν σκόπιμο να εξηγηθούν οι βασικές έννοιες Εισόδου /Εξόδου των πλακετών τυπωμένων κυκλωμάτων (PCB) που

εφαρμόζονται με παρόμοιο τρόπο και στην επικοινωνία γενικά μεταξύ ολοκληρωμένων κυκλωμάτων (IC). Η επικοινωνία μεταξύ δυο πλακετών τυπωμένων κυκλωμάτων (PCB) πραγματοποιείται μέσω μιας γραμμής μετάδοσης (transmission line). Πιο αναλυτικά, η γραμμή μετάδοσης αποτελεί το υλικό μέσο ή τη δομή που χρησιμοποιείται ώστε να σχηματιστεί ένα μονοπάτι ροής της πληροφορίας από ένα σημείο σε ένα άλλο. Στην παράλληλη επικοινωνία η γραμμή μετάδοσης διαμορφώνεται από πολλές μονοτερματικές (single ended) συνδέσεις σήματος (στην πράξη ακροδεκτών pins καλωδίων), βλέπε Σχήμα 1-1. Συγκεκριμένα, οι μονοτερματικοί είσοδοι/έξοδοι (single-ended I/O), συνδέουν δύο πλακέτες τυπωμένων κυκλωμάτων (PCB) ή ολοκληρωμένα κυκλώματα (IC) με μια σύνδεση σήματος, που ερμηνεύεται ως μια φυσική σύνδεση ενός ακροδέκτη με έναν άλλον ακροδέκτη (pins) καλωδίου. Η μονοτερματική (single ended) σηματοδοσία λειτουργεί συγκρίνοντας την τάση σήματος είτε με ένα συγκεκριμένο εύρος τάσης είτε με μια τάση αναφοράς. Αντίθετα στη σειριακή επικοινωνία η γραμμή μετάδοσης διαμορφώνεται από μια διαφορική (differential) σύνδεση σήματος (στην πράξη, διαφορικό ζεύγος ακροδεκτών pins καλωδίων), βλέπε Σχήμα 1-1. Συγκεκριμένα, οι διαφορικοί είσοδοι/έξοδοι (differential I/O), συνδέουν δύο πλακέτες τυπωμένων κυκλωμάτων (PCB) ή ολοκληρωμένα κυκλώματα (IC) με μια διαφορική σύνδεση σήματος, που ερμηνεύεται ως μια φυσική σύνδεση ενός ζεύγους ακροδεκτών (pins) καλωδίου. Η διαφορική (differential) σηματοδοσία αντί να λειτουργεί συγκρίνοντας την τάση σήματος με συγκεκριμένη τάση ή τάση αναφοράς όπως η μονοτερματική, λειτουργεί συγκρίνοντας δύο τάσεις σήματος μεταξύ τους. Έτσι αν το σήμα που αναφέρεται ως θετικός κόμβος διαθέτει μεγαλύτερη τιμή τάσης από ότι αυτό που αναφέρεται ως αρνητικός κόμβος το σήμα θεωρείται στο θετικό επίπεδο ή σε λογικό '1'. Αν το σήμα που αναφέρεται ως αρνητικός κόμβος διαθέτει μεγαλύτερη τιμή τάσης από ότι αυτό που αναφέρεται ως θετικός κόμβος το σήμα θεωρείται στο αρνητικό επίπεδο ή σε λογικό '0'. Οι θετικοί και αρνητικοί ακροδέκτες (pins) οδηγούνται με ακριβώς συμπληρωματικά σήματα όπως φαίνεται παρακάτω:



Σχήμα 1-2 : Διαφορική Σηματοδοσία

Για αυτόν τον λόγο η διαφορική σηματοδοσία είναι πιο ανθεκτική στο θόρυβο. Επίσης προσφέρεται για τη διατήρηση μιας σταθερής ροής πληροφορίας και συνεχούς μεταφοράς δεδομένων με μορφή ομοχειρίας (pipeline) διαμέσου της γραμμής μετάδοσης. Αυτά τα χαρακτηριστικά της διαφορικής σηματοδοσίας σε συνδυασμό με το μικρότερο πλήθος ακροδεκτών (pins) που χρειάζεται η γραμμή μετάδοσης στη σειριακή επικοινωνία, αφού της αρκεί ένα μόνο διαφορικό ζεύγος ακροδεκτών (pins) για μονόδρομη επικοινωνία και δύο διαφορικά ζεύγη ακροδεκτών (pins) στην αμφίδρομη επικοινωνία, καθιστούν τη σειριακή επικοινωνία άκρως αποδοτική.

Για το σκοπό της μεταφοράς δεδομένων από ή προς chip, πλακέτες, ή συσκευές (box), μια σειριακή σύνδεση υψηλών ταχυτήτων είναι πολύ ανταγωνιστική. Με ταχύτητες γραμμής μετάδοσης από 0.660 ως 12 Gbaud/s και φορτία από 0.8 ως 10Gb, αποτελεί μια πολύ καλή λύση. Απαραίτητο κύκλωμα για τη μετακίνηση πολλών δεδομένων γρήγορα αποτελούν οι Multi-gigabit πομποδέκτες (MGTs). Το MGT (Multi-Gigabit Transceiver) αποτελεί ένα άλλο όνομα για το multi-gigabit Serializer/Deserializer (SERDES). Ένας MGT έχει ως έργο την παράλληλη λήψη δεδομένων και τη μετατροπή

τους σε σειριακά δεδομένα και κατόπιν την αποστολή τους με πολύ υψηλές ταχύτητες μέσω μια γραμμής μετάδοσης σειριακής επικοινωνίας σε έναν άλλο MGT. Ταυτόχρονα ένας MGT έχει ως έργο την λήψη σειριακών δεδομένων με πολύ υψηλές ταχύτητες μέσω μια γραμμής μετάδοσης σειριακής επικοινωνίας από έναν άλλο MGT και τη μετατροπή τους σε παράλληλα δεδομένα.

### 1.1.2 Πλεονεκτήματα –Μειονεκτήματα – Προκλήσεις

Ποιο είναι το κυριότερο πλεονέκτημα της σειριακής Εισόδου/Εξόδου πολύ υψηλών ταχυτήτων; Η ταχύτητα. Ακόμα, οι λιγότεροι ακροδέκτες (pins), χωρίς προβλήματα μαζικών σύγχρονων δρομολογήσεων εξόδων δεδομένων (SSO), μικρότερη EMI, και μικρότερο κόστος. Πιο αναλυτικά τα πλεονεκτήματα:

#### Μέγιστη Ροή Δεδομένων

Μερικές μεγάλες επαναπρογραμματιζόμενες λογικές συσκευές έχουν 20 και πλέον σειριακούς πομποδέκτες των 10-Gb για συνολικό εύρος ζώνης 200 Gb/s εισόδου κι εξόδου. Ενώ αυτό είναι το άκρο, ας δούμε ένα παράδειγμα εφαρμογής που μας δείχνει πως η ταχύτητα της σειριακής Εισόδου/Εξόδου μπορεί να βοηθήσει τους αρχιτέκτονες συστήματος, τους σχεδιαστές πλακετών, και σχεδιαστές λογικής. Η ροή κάθε βίντεο υψηλής ευκρίνειας (High-Definition) χρειάζεται 1.5 Gb/s όταν μεταφέρεται στη βασική ζώνη ή σε μη συμπίεσμένη μορφοποίηση. Ένα σενάριο για την κατασκευή αυτού του συστήματος περιλαμβάνει chip διακριτής σειριοποίησης κι αποσειριοποίησης για τις ροές σειριακού βίντεο, και παράλληλες διεπαφές για το δίαυλο επέκτασης και την αποθήκευση του clip. Το άλλο σενάριο χρησιμοποιεί gigabit πομποδέκτες μέσα στο τμήμα λογικής για την αποκωδικοποίηση και την κωδικοποίηση των σειριακών ροών. Η σειριακή ροή ενεργεί ως διεπαφή στην υποδοχή επέκτασης (expansion connector) και στην αποθήκευση του clip, είναι ταχύτερη κι αποτελεί την επιλογή που προτιμάται.

#### Αριθμός των Ακροδεκτών (Pins)

Ο αριθμός των ακροδεκτών (pins) είναι το πρώτο πρόβλημα που συναντάται στην προσπάθεια μετακίνησης πολλών δεδομένων μέσα κι έξω από ένα chip ή πλακέτα. Ο αριθμός των ακροδεκτών (pins) εισόδου κι εξόδου είναι πάντα περιορισμένος. Παρόλο που ο αριθμός των διαθέσιμων ακροδεκτών (pins) τείνει να αυξάνεται με τον καιρό, δεν είναι ποτέ αρκετός για να ακολουθήσει τις ανάγκες. Για να είμαστε δίκαιοι, υπάρχουν κάποια θέματα ακροδεκτών (pins) τα οποία δε λογαριάζουμε. Για παράδειγμα, κάποιοι MGTs χρειάζονται περισσότερους ακροδέκτες (pins) τροφοδοσίας και γείωσης απ' ότι ζεύγη πιο αργών ακροδεκτών (pins) . Και μια παράλληλη διεπαφή μπορεί να χρειάζεται ειδικούς ακροδέκτες (pins) αναφοράς. Αλλά αυτό το παράδειγμα είναι εξίσου σχετικό για σύγκριση. Ο χρόνος σχεδίασης πλακέτας και το κόστος μπορεί να αυξηθεί δραματικά όταν χρησιμοποιείται ένας μεγάλος αριθμός ακροδεκτών (pins) . Ο αριθμός των ακροδεκτών (pins) των υποδοχών διασύνδεσης (connectors) είναι επίσης εξαιρετικά σημαντικός για την επιλογή υποδοχής διασύνδεσης(connector)/καλωδίου και τη μελέτη της απόδοσης. Και η χρήση όλων των διαθέσιμων ακροδεκτών (pins) του Ball Grid Array (BGA) μπορεί να μην είναι βολική.

#### Ταυτόχρονα Διακοπτόμενες Έξοδοι

Το φαινόμενο των ταυτόχρονα διακοπτόμενων εξόδων (Simultaneous Switching Outputs ή ground bounce) έγκειται στο γεγονός ότι όταν πολύ μεγάλος αριθμός εξόδων ανοιγοκλείνουν ταυτόχρονα, η σύζευξη με τη γείωση δημιουργεί πολύ θόρυβο και μια λογική πύλη μεταβαίνει στην αβέβαιή της κατάσταση. Ένας σχεδιαστής πρέπει να προνοήσει για το πρόβλημα SSO όταν χρησιμοποιεί παράλληλους μονοτερματικούς διαύλους. Είναι δυνατόν να χρησιμοποιηθεί επεξεργασία διαφορικού σήματος σε όλες τις Εισόδους/Εξόδους ώστε να αντιμετωπίσει το πρόβλημα του SSO, αλλά αυτό διπλασιάζει τον αριθμό των ακροδεκτών (pins).

### EMI

Η εμπειρία έχει αποδείξει ότι όσο τα ρολόγια γίνονται ταχύτερα, οι δοκιμές μέτρησης εκπομπών ηλεκτρομαγνητικής ακτινοβολίας γίνονται πιο δύσκολες. Συνεπώς, η σχεδίαση gigabit μπορεί να φαντάζει δύσκολη. Όμως μια σειριακή σύνδεση υψηλών ταχυτήτων συνήθως έχει να επιδείξει λιγότερες εκπομπές ακτινοβολίας από ότι ένας μεγάλος δίαυλος με ροή σε χαμηλότερο ρυθμό. Αυτό γιατί οι εν λειτουργία σειριακές συνδέσεις gigabit προϋποθέτουν τέλεια ακεραιότητα σήματος σε αντίθεση με τις παράλληλες συνδέσεις που κάτι τέτοιο δεν ισχύει. Έτσι οι σειριακές συνδέσεις gigabit ελαχιστοποιούν τις ανακλάσεις σήματος στη γραμμή μετάδοσης κι εκμεταλλεύονται την ηλεκτρομαγνητική ενέργεια πιο αποδοτικά. Σύμφωνα με μια ρήση της επιστημονικής κοινότητας, “Τα προβλήματα ακτινοβολίας εκπομπών είναι στην πραγματικότητα απλά προβλήματα ακεραιότητας σήματος”.

### Κόστος

Χρησιμοποιώντας τους MGTs θα έχει συχνά αποτέλεσμα το χαμηλότερο συνολικό κόστος του συστήματος. Με ένα μικρότερο, φθηνότερο πακέτο, οι υποδοχές διασύνδεσης (connectors) μπορούν να έχουν λιγότερους ακροδέκτες (pins) και η σχεδίαση της πλακέτας μπορεί να είναι επίσης πιο απλή. Σε μια εφαρμογή μίκτη βίντεο (video mixer), η παράλληλη λύση είχε επιπλέον εννιά ICs (ολοκληρωμένα κυκλώματα) από ότι η σειριακή λύση. Στο εν λόγω παράδειγμα, το κόστος τη σειριακής λύσης είναι εκατοντάδες δολάρια λιγότερο από την παράλληλη λύση.

### Πυρήνες Προκαθορισμένων Πρωτοκόλλων

Ένα άλλο κέρδος από τη χρήση των MGTs είναι η διαθεσιμότητα πυρήνων IP (Intellectual Property Cores) προκαθορισμένων πρωτοκόλλων και προτύπων διεπαφών. Από το Aurora ως το XAUI, υφίστανται ήδη σχεδιάσεις για πολλές διαφορετικές ανάγκες.

### Μειονεκτήματα

Πριν αναλογιστούμε ότι η σειριακή Έξοδος/Είσοδος πολύ υψηλών ταχυτήτων ακούγεται πολύ καλή για να είναι αληθινή, ας δούμε τα μειονεκτήματα. Στις σχεδιάσεις μας πρώτα πρέπει να προσέξουμε θέματα ακεραιότητας των σημάτων. Για παράδειγμα, έχει αναφερθεί ακόμα και ρυθμός αποτυχίας 90% στις πρώτες προσπάθειες για σχεδιάσεις υψηλών ταχυτήτων για συγκεκριμένη εφαρμογή. Για να βελτιώσουμε τις πιθανότητες, ίσως χρειαστεί να διενεργηθούν αναλογικές προσομοιώσεις και να

χρησιμοποιηθούν νέα πιο πολύπλοκα σχέδια παράκαμψης των δοκιμών σε πραγματικό υλικό. Κατ' ακρίβειαν, μπορεί να χρειαστεί πιο ακριβής και χρονοβόρα μοντελοποίηση και προσομοίωση της εκάστοτε σχεδίασης σειριακής Εισόδου/Εξόδου.

Επίσης έχουν μεγαλύτερο κόστος οι πλακέτες τυπωμένων κυκλωμάτων (PCB) με ελεγχόμενη αντίσταση, οι υποδοχές διασύνδεσης (connectors) υψηλών ταχυτήτων, και τα καλώδια των σχεδιάσεων αυτών. Θα αντιμετωπιστούν επιπλοκές και μικρότερες βάσεις χρόνου στις ψηφιακές προσομοιώσεις. Και όταν γίνεται εκμετάλλευση των οφελών ενός προϋπάρχοντος πρωτοκόλλου, πρέπει να προβλεφθεί ο χρόνος για την ενσωμάτωση και τις επιπλέον πύλες ή ο αριθμός των κύκλων της Κεντρικής μονάδας επεξεργασίας (CPU) για την επιπλέον πληροφορία (overhead) του πρωτοκόλλου.

### 1.1.3 Εφαρμογές της σειριακής μεταφοράς δεδομένων

Αρχικά, ο gigabit SERDES ήταν περιορισμένος στην βιομηχανία τηλεπικοινωνιών και σε μερικές εντός αυτής αγορές όπως της μετάδοσης βίντεο. Σήμερα, οι εφαρμογές MGT εμφανίζονται σε κάθε τομέα στην αγορά ηλεκτρονικών στρατού, φαρμακευτική, δικτύων, βίντεο, τηλεπικοινωνιών, κτλ. Επίσης χρησιμοποιούνται σε συνδεσμολογίες πλακετών τυπωμένων κυκλωμάτων (PCB) μέσω συνδέσεων backplane και μεταξύ συσκευών. Οι MGTs είναι κρίσιμοι για το μέλλον των ηλεκτρονικών. Παρατίθενται τα πρότυπα της βιομηχανίας που κάνουν χρήση των multi-gigabit SERDES:

- Κανάλι Οπτικών Ινών (FC)
- PCI Express
- RapidIO Serial
- Advanced Switching Interface
- Serial ATA
- 1-Gb Ethernet
- 10-Gb Ethernet (XAUI)
- Infiniband 1X, 4X, 12X

### Chip-σε-Chip

Ο SERDES αρχικά χρησιμοποιήθηκε για τη συνομιλία συσκευής με συσκευή. Αλλά εκτινάχθηκε στην αγορά εξ αιτίας του πόσο καλά χειρίζεται τις επικοινωνίες από chip-σε-chip στην ίδια πλακέτα κυκλωμάτων. Η επικοινωνία από Chip-σε-chip προηγουμένως ήταν σχεδόν αποκλειστικά πεδίο παραλληλίας. Το κόστος σειριακών σχεδιάσεων ήταν αυξημένο καθώς το πλήθος της λογικής που χρειάζεται για τη σειριοποίηση και αποσειριοποίηση ξεπερνά κατά πολύ σε βαρύτητα την οικονομία στο κόστος που προέρχεται από τη μείωση του αριθμού των ακροδεκτών (pins). Αλλά με τις νέες τεχνολογίες σχεδίασης, μια απίστευτη ποσότητα λογικής μπορεί να χωρέσει σε πολύ μικρή ποσότητα πυριτίου. Οι SERDES λοιπόν, μπορούν να συμπεριληφθούν σε εξαρτήματα για πολύ χαμηλό κόστος πυριτίου. Σε αυτό προστίθεται η πάντα αυξανόμενη ανάγκη για εύρος ζώνης Εισόδου/Εξόδου, και ο SERDES γρήγορα γίνεται η λογική επιλογή για τη μετακίνηση όποιου σημαντικού πλήθους δεδομένων από chip-σε-chip. Θεωρούνται τα ακόλουθα πλεονεκτήματα της επικοινωνίας SERDES από chip-σε-chip:

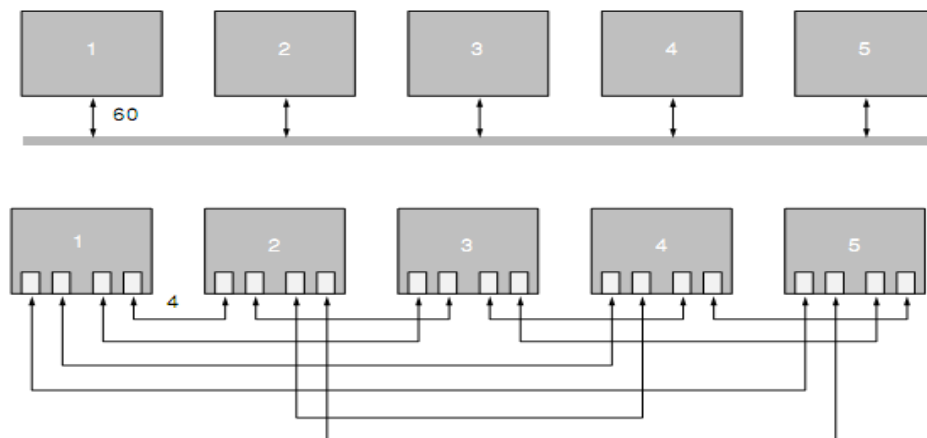
- Αριθμός Ακροδεκτών (Pins): Μικρότερος, φθηνότερες συσκευασίες.
- Αριθμός Ακροδεκτών (Pins): Λιγότερα στρώματα στη συνδεσμολογία PCB.



- Μικρότερες Συσκευασίες: Μικρότερες, φθηνότερες πλακέτες και πιο συμπαγείς σχεδιάσεις.
- SSO: Οι λιγότεροι ακροδέκτες (pins) και η διαφορική σηματοδοσία εξαλείφουν το πρόβλημα του SSO.
- Ενέργεια: Συνήθως μια σειριακή σύνδεση υψηλών-ταχυτήτων θα χρησιμοποιήσει λιγότερη ενέργεια από μια παράλληλη σύνδεση λόγω του ότι η ακεραιότητα του σήματος στη σειριακή σύνδεση είναι «τέλεια» σε σχέση με την παράλληλη σύνδεση καθώς στη σειριακή σύνδεση ελαχιστοποιείται η ανάκλαση σήματος. Αυτό είναι αληθές ειδικά για κάποια από τα ενεργητικά συζευγμένα/τερματισμένα παράλληλα πρότυπα υψηλών-ταχυτήτων όπως λογική τρανζίστορ υψηλών ταχυτήτων (HSTL).
- Οι Γραμμές Ελέγχου συμπεριλαμβάνονται: Συχνά μια παράλληλη διεπαφή χρειάζεται μερικές γραμμές για έλεγχο και ενεργοποίηση επιπρόσθετα στις γραμμές δεδομένων. Οι σειριακές συνδέσεις έχουν ικανότητες ενεργοποίησης και ελέγχου ενσωματωμένες μέσα στα περισσότερα πρωτόκολλα.

### Πλακέτα-σε-Πλακέτα / Πλακέτες βάσης (Backplanes)

Οι σειριακές αρχιτεκτονικές μετάδοσης δεδομένων είναι πιο ανταγωνιστικές από τις παράλληλες. Στην περίπτωση μετάδοσης δεδομένων μέσω διαύλου τα περισσότερα πρωτόκολλα παράλληλου διαύλου έχουν εξελιχθεί στο σημείο όπου η προσθήκη bit δεδομένων είναι από φυσικής άποψης μη πρακτική εξαιτίας του αριθμού των ακροδεκτών (pins) στις υποδοχές διασύνδεσης (connectors). Η στρέβλωση ρολογιού (clock skew), η στρέβλωση δεδομένων (data skew), οι χρόνοι ανόδου και καθόδου, και η απόκλιση σήματος (jitter) περιορίζουν την ικανότητα αύξησης της συχνότητας ρολογιού. Ο διπλασιασμός του ρυθμού δεδομένων μπορεί να βοηθήσει, αλλά συχνά χρειάζεται η μετακίνηση σε διαφορική σηματοδοσία για πιο αξιόπιστη σηματοδοσία, κι αυτό αυξάνει δραστικά τον αριθμό των ακροδεκτών (pins). Επίσης, ο έλεγχος των θεμάτων αμφίδρομης συνομιλίας στους παράλληλους διαύλους είναι δύσκολος. Οι νέες σειριακές πλακέτες βάσης (backplanes) ελέγχου είναι κατά κάποιον τρόπο διαφορετικές από τις παράλληλες. Έχουν αποκλειστικές σειριακές συνδέσεις από κάθε κόμβο σε κάθε άλλο κόμβο. Οι αρχιτεκτονικές σειριακού διαύλου έχουν πολλά να προσφέρουν. Ο αριθμός των ακροδεκτών (pins) σειριακού διαύλου είναι μια συνάρτηση του αριθμού των κόμβων. Για συνηθισμένο αριθμό κόμβων, μια σειριακή αρχιτεκτονική έχει λιγότερους ακροδέκτες (pins) από ότι η παλιά παράλληλη αρχιτεκτονική.



Σχήμα 1-3 : Παράλληλος Δίαυλος και Σειριακός Δίαυλος

Ίσως η πιο σημαντική διαφορά μεταξύ των δύο είναι η μέθοδος προσπέλασης του εύρους ζώνης. Στις παράλληλες αρχιτεκτονικές, ένας κόμβος μπορεί να μεταδώσει σε έναν ή πολλούς κόμβους. Αλλά ενώ αυτός ο κόμβος μεταδίδει, όλοι οι άλλοι κόμβοι είναι μπλοκαρισμένοι. Όλοι οι κόμβοι μοιράζονται το διαθέσιμο εύρος ζώνης. Στους σειριακούς διαύλους, κάθε κόμβος έχει μια αποκλειστική σύνδεση σε κάθε άλλο κόμβο. Οπότε ένας κόμβος μπορεί να μιλήσει με έναν ή όλους τους άλλους κόμβους ενώ ένας άλλος κόμβος μιλάει. Κατ' ακρίβειαν, όλοι οι κόμβοι μπορούν να μιλήσουν ο ένας με τον άλλον την ίδια στιγμή. Φυσικά, οι κόμβοι θα πρέπει να έχουν αποθήκευση ουράς (FIFO) ώστε να μπορούν να επεξεργαστούν όλες τις πληροφορίες που έχουν ληφθεί. Τα πλεονεκτήματα της δομής του σειριακού διαύλου:

- Μεγαλύτερο εύρος ζώνης
- Μειωμένος αριθμός ακροδεκτών (pins)
- Προσδιορισμένο εύρος ζώνης κόμβου-προς-κόμβο (καμία ανάγκη διαμοιρασμού)
- Οι σχεδιάσεις κατασκευάζονται μέσα στον SERDES
- Υποστηρίζει εύκολα πρωτόκολλα

#### Συσκευή-σε-Συσκευή (Box-to-Box)

Ενώ οι SERDES έκαναν την αρχή τους συνδέοντας συσκευές, πολλοί σχεδιαστές δεν θεωρούν τις σειριακές multi-gigabit συνδέσεις για χρήση σε επικοινωνίες συσκευής-σε-συσκευή. Μια κοινή παρεξήγηση είναι πως η επικοινωνία συσκευής-σε-συσκευή δεν μπορεί να είναι γρήγορη χωρίς οπτικές ίνες. Παρ' όλ' αυτά, υπάρχουν πολλές συνδέσεις που πάνε από κουτί-σε-κουτί για σύντομες αποστάσεις μέσω συστημάτων καλωδιώσεων χαλκού. Ένα πρότυπο που χρησιμοποιεί αυτές τις συνδέσεις είναι το Infiniband. Οι προδιαγραφές Infiniband επιτρέπουν 1, 4, ή 12 κανάλια σειριακών δεδομένων 2.5 Gb/s ανά ροή. Το πρότυπο έχει γίνει διαθέσιμο εμπορικά για αρκετά χρόνια και περιλαμβάνει καλώδια, υποδοχές διασύνδεσης (connectors), και πρωτόκολλο που όλα είναι σαφώς ορισμένα και δοκιμασμένα.

#### 1.1.4 Μέχρι που μπορεί να φτάσει - Δυνατότητες

Με την πρώτη ματιά, η multi-gigabit επικοινωνία φαίνεται να προτάσσει σοβαρούς περιορισμούς. Οι σχεδιαστές σειριακών πρέπει να παλέψουν με την ακεραιότητα σήματος, μικρότερες βάσεις χρόνου, και πιθανότατα την ανάγκη για επιπλέον πύλες και πρόσθετους κύκλους CPU. Παρ' όλ' αυτά τα πλεονεκτήματα του multi-gigabit στις επικοινωνίες συσκευής-σε-συσκευή και chip-σε-chip αντισταθμίζουν μακράν τα εκλαμβανόμενα αδιέξοδα. Για παράδειγμα, υψηλή ταχύτητα, λιγότεροι ακροδέκτες (pins), χαμηλότερη EMI, και χαμηλότερο κόστος τον καθιστούν μια ιδανική επιλογή σε πολλές σχεδιάσεις επικοινωνίας. Αυτά τα πλεονεκτήματα θα εξασφαλίσουν την συνεχιζόμενη χρήση του σε εφαρμογές επικοινωνιών μακράν στο μέλλον.

## 1.2 Σκοπός της διπλωματικής

Σκοπός αυτής της διπλωματικής εργασίας είναι η μελέτη της αξιόπιστης μεταφοράς δεδομένων για Είσοδο/Εξοδο σειριακά σε πολύ υψηλές ταχύτητες. Πρόκειται για ταχύτητες μεταφοράς δεδομένων της τάξεως των 600 Mbps έως 3.125 Gbps.

Για αυτούς τους λόγους, στην εργασία αυτή, έχει υλοποιηθεί μια σχεδίαση σειριακής Είσοδου/Εξόδου με πολύ υψηλές ταχύτητες χρησιμοποιώντας το πρωτόκολλο επιπέδου σύνδεσης (Link Layer Protocol) που ονομάζεται Aurora. Χρησιμοποιείται το αναπτυξιακό σύστημα XUP που φέρει την FPGA Virtex™ II Pro. Πιο συγκεκριμένα το σύστημα που υλοποιείται βασίζεται στη λειτουργία του κυκλώματος που καλείται Rocket I/O και περιλαμβάνει τους λεγόμενους MGTs δηλ. Multi-gigabit πομποδέκτες (Multi-Gigabit Transceivers). Πρόκειται για σχεδίαση η οποία περιλαμβάνει τον hardcore επεξεργαστή PowerPC και ένα περιφερειακό, το οποίο υλοποιεί το πρωτόκολλο επιπέδου διασύνδεσης Aurora, συνδεδεμένο με αυτόν μέσω του διαύλου υψηλών ταχυτήτων PLB. Υλοποιούνται ακόμα κι άλλα περιφερειακά για επιμέρους λειτουργίες της σχεδίασης ανάμεσα στα οποία αξίζει να αναφερθεί το περιφερειακό ψηφιακού ελέγχου ρολογιού.

Για την ολοκλήρωση αυτής της διπλωματικής εργασίας εκτελέστηκαν τα ακόλουθα βήματα:

1. Μελέτη σχετικής θεωρίας για σειριακή Είσοδο/Εξοδο και του πρωτοκόλλου επιπέδου διασύνδεσης Aurora.
2. Λήψη της σχεδίασης αναφοράς από το site της Xilinx, δημιουργία του αντίστοιχου πυρήνα Aurora και εφαρμογή των οδηγιών της τεκμηρίωσης της σχεδίασης και προσαρμογή των παραμέτρων της στο διαθέσιμο υλικό και λογισμικό.
3. Εκτέλεση προσομοίωσης της σχεδίασης αναφοράς για τη μεταφορά δεδομένων σειριακά σε υψηλές ταχύτητες.
4. Εκτέλεση μοντελοποίησης του κυκλώματος (bus functional model) με σκοπό προσομοίωση της λειτουργίας της σχεδίασης αναφοράς για τη μεταφορά δεδομένων σειριακά σε υψηλές ταχύτητες.
5. Υλοποίηση της σχεδίασης αναφοράς στην πλακέτα και επιβεβαίωση μεταφοράς δεδομένων με δειγματοληψία των σημάτων της πλακέτας μέσω download cable και με επικοινωνία μέσω σειριακής θύρας με τον υπολογιστή.
6. Δοκιμές με αλλαγές παραμέτρων της σχεδίασης και κατανόηση βασικών αρχών λειτουργίας της.
7. Μελέτη σχεδίασης αναφοράς της Xilinx που περιλαμβάνει σύστημα με έναν επεξεργαστή και τη μνήμη εντολών και δεδομένων του, ένα περιφερειακό σηματοδοσίας ρολογιού κι ένα περιφερειακό για σειριακή επικοινωνία υψηλών ταχυτήτων καθώς και άλλα περιφερειακά.
8. Σχεδίαση και υλοποίηση συστήματος με έναν επεξεργαστή και τη μνήμη εντολών και δεδομένων του, ένα περιφερειακό σηματοδοσίας ρολογιού και αυτοσχέδιο περιφερειακό για σειριακή επικοινωνία υψηλών ταχυτήτων.
9. Σχεδίαση των βιβλιοθηκών και της εφαρμογής λογισμικού.
10. Εκτέλεση επαναλαμβανόμενων δοκιμών, έλεγχος ακραίων καταστάσεων και τελειοποίηση της σχεδίασης.

Σε αυτήν την εργασία πραγματοποιήθηκε:  
*σε υλικό*

- Υλοποίηση συστήματος που περιλαμβάνει έναν επεξεργαστή και τη μνήμη εντολών και δεδομένων του, ένα περιφερειακό σηματοδοσίας ρολογιού κι ένα περιφερειακό για σειριακή επικοινωνία υψηλών ταχυτήτων.
- Υλοποίηση πανομοιότυπου συστήματος με το παραπάνω με την προσθήκη περιφερειακού για την ακριβή μέτρηση των υψηλών ταχυτήτων της σειριακής επικοινωνίας.
- Σύνδεση μίας πλακέτας μέσω καλωδίου SATA, κατασκευή εξωτερικής αναδρομής (loopback) στην ίδια πλακέτα και εκτέλεση μεταφοράς δεδομένων σειριακά σε ταχύτητες της τάξης των Gbps.
- Σύνδεση δύο πλακετών μέσω καλωδίου SATA κι εκτέλεση μεταφοράς δεδομένων σειριακά σε πολύ υψηλές ταχύτητες.

*σε λογισμικό*

- Υλοποίηση βιβλιοθηκών που χρησιμεύουν ως API για την κλήση συναρτήσεων μεταφοράς δεδομένων.
- Υλοποίηση εφαρμογής σε γλώσσα προγραμματισμού C για την εκτέλεση δοκιμών μεταφοράς δεδομένων με διάφορες παραλλαγές όπως
  - Μελέτη αξιόπιστης μεταφοράς δεδομένων
  - Μελέτη ταχύτατης μεταφοράς δεδομένων
  - Μελέτη μεταφοράς πολύ μεγάλου όγκου δεδομένων

## ΚΕΦΑΛΑΙΟ 2

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

#### 2.1 Γνωριμία με τα χαρακτηριστικά του αναπτυξιακού συστήματος

##### 2.1.1 Επισκόπηση

Στόχος του κεφαλαίου αυτού είναι η γνωριμία με τα χαρακτηριστικά του αναπτυξιακού συστήματος που αποτελεί την πλατφόρμα πάνω στην οποία υλοποιούνται σε υλικό οι σχεδιάσεις και εκτελούνται οι δοκιμές «στον πάγκο». Έτσι γίνεται εισαγωγή στην ιδέα του πανεπιστημιακού προγράμματος της Xilinx του οποίου μέρος αποτελεί και το αναπτυξιακό σύστημα που χρησιμοποιείται. Στη συνέχεια γίνεται αναφορά στο ολοκληρωμένο αναδιατασσόμενη λογική (FPGA) που διαθέτει το εν λόγω σύστημα και τις παραμέτρους του. Τέλος επισημαίνονται τα χαρακτηριστικά του αναπτυξιακού συστήματος και της FPGA που θα χρειαστούν στη σχεδίαση της εργασίας αυτής.

##### 2.1.2 Λίγα λόγια για την XUP Virtex II Pro FPGA

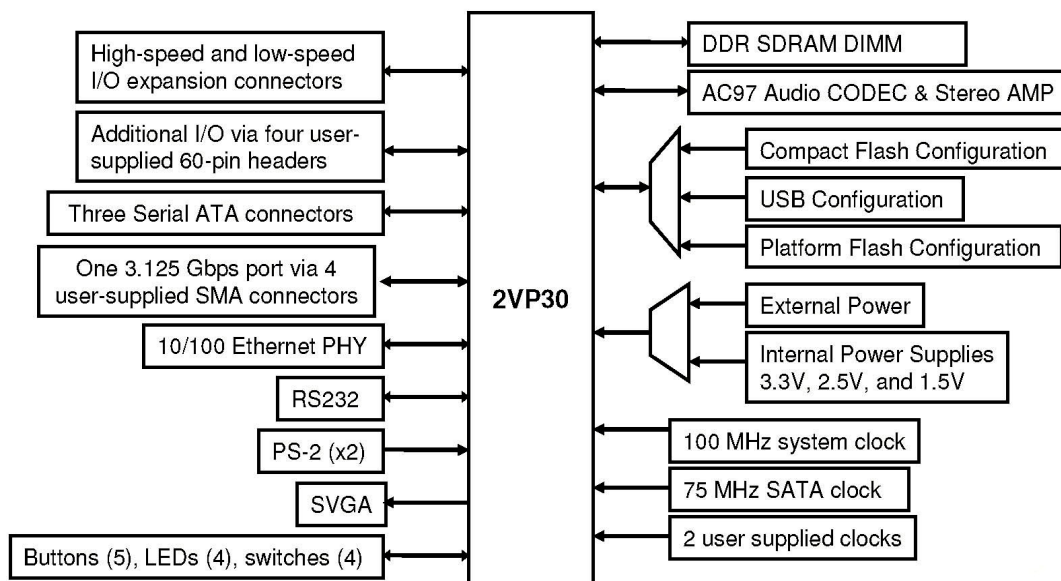
###### Γενική περιγραφή

Η XUP Virtex™-II Pro είναι μια αναπτυξιακή πλακέτα FPGA που μπορεί να χρησιμοποιηθεί για την υλοποίηση ενός ευρέους φάσματος σχεδιάσεων υλικού. Το XUP αποτελεί τα αρχικά Xilinx University Program. Είναι συνεπώς αντιληπτό ότι η XUP Virtex™-II Pro FPGA πλακέτα χρησιμεύει ως εκπαιδευτικό βοήθημα για την ψηφιακή σχεδίαση, ως ένα αναπτυξιακό σύστημα μικροεπεξεργαστή, ή για τη φιλοξενία πυρήνων ενσωματωμένων επεξεργαστών και πολύπλοκων ψηφιακών συστημάτων. Το εν λόγω σύστημα παρέχει μια προηγμένη πλατφόρμα υλικού που αποτελείται από μια υψηλών επιδόσεων FPGA Virtex-II Pro πλαισιωμένη από μια ευκολονόητη συλλογή από περιφερειακές συσκευές που μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός πολύπλοκου συστήματος καθώς και για την επίδειξη των ικανοτήτων μιας FPGA Virtex-II Pro Platform. Είναι αρκετά ισχυρό ώστε να υποστηρίζει προχωρημένα ερευνητικά προγράμματα, αλλά σε τιμή προσιτή ώστε να τοποθετηθεί σε κάθε σταθμό εργασίας. Διαθέτει υποδοχές επέκτασης που μπορούν να φιλοξενήσουν γενικής χρήσης κυκλώματα και συστήματα στο μέλλον. Μπορεί να παραμείνει στο πυρήνα ενός εκπαιδευτικού προγράμματος για απεριόριστο χρόνο και υποστηρίζεται από εργαλεία σχεδίασης υψηλής κλάσης (ISE™, Εργαλείο ChipScope™ Pro, Embedded Developers Kit (EDK), Core generator για DSP (Επεξεργασία ψηφιακών συστημάτων)

###### Χαρακτηριστικά XUP Xilinx Πανεπιστημιακού Προγράμματος Virtex-II Pro Αναπτυξιακών Συστημάτων

Στη συνέχεια, παραθέτονται εν συντομία, τα χαρακτηριστικά ενός XUP. Πρώτον διατίθεται η FPGA Virtex™-II Pro με πυρήνες PowerPC™ 405. Παρέχεται η δυνατότητα προσθήκης μέχρι 2 GB Double Data Rate (DDR) SDRAM. Υπάρχει ακόμα ελεγκτής

System ACE™ κι υποδοχή διασύνδεσης (connector) CompactFlash™ τύπου II για τη ρύθμιση της FPGA κι αποθήκευση δεδομένων. Περιλαμβάνεται θύρα ρύθμισης Embedded Platform Cable USB. Επίσης, υψηλών ταχυτήτων SelectMAP ρύθμιση της FPGA από Platform Flash In-System. PROM προγραμματιζόμενης διαμόρφωσης. Υπάρχει υποστήριξη δύο διαμορφώσεων της FPGA, «Χρυσή» και «Χρήστη», με βάση ακολουθία bit (bitstream). Βρίσκεται στην πλακέτα, συσκευή Ethernet 10/100 PHY. Διαθέτει σειριακό αριθμό πυριτίου για μοναδική αναγνώριση της πλακέτας. Σειριακή θύρα RS-232 DB9, όπως επίσης και δύο σειριακές θύρες PS-2. Ακόμα, τέσσερις φωτοδιόδους (LEDs) συνδεδεμένες σε ακροδέκτες (pins) της Virtex-II Pro I/O, τέσσερις διακόπτες συνδεδεμένους σε ακροδέκτες (pins) της Virtex-II Pro I/O, πέντε κουμπιά συνδεδεμένα σε ακροδέκτες (pins) της Virtex-II Pro I/O. Παρέχονται έξι υποδοχές επέκτασης ενωμένες σε 80 ακροδέκτες (pins) της Virtex-II Pro I/O με προστασία υπέρτασης καθώς και υποδοχές διασύνδεσης υψηλών ταχυτήτων ενωμένες σε 40 ακροδέκτες (pins) της Virtex-II Pro I/O που μπορούν να χρησιμοποιηθούν διαφορεικά ή μόνες τους. Υπάρχει AC-97 CODEC ήχου με ενισχυτή ήχου με έξοδο ηχείου/ακουστικού και έξοδο επιπέδου γραμμής καθώς και είσοδος ήχου επιπέδου γραμμής και μικροφώνου. Περιλαμβάνεται στην πλακέτα έξοδος XSGA , μέχρι 1200 x 1600 με 70 Hz ρυθμό ανανέωσης. Επίσης, τρεις Serial ATA θύρες, δύο Host θύρες και μια Target θύρα. Διατίθεται υποδοχή επέκτασης MGT εκτός πλακέτας, με ρολόι που παρέχεται από το χρήστη. Κρύσταλλος 100 MHz για ρολόι συστήματος, 75 MHz ρολόι SATA αλλά και παροχή για ρολόι που προμηθεύει ο χρήστης. Τέλος τροφοδοτικό ενσωματωμένο στην πλακέτα, όπως επίσης ομάδα κυκλωμάτων Power-on και reset καθώς και PowerPC 405 reset ομάδα κυκλωμάτων. Παρουσιάζεται παρακάτω ένα block Διάγραμμα της XUP Virtex-II Pro αναπτυξιακής πλακέτας.



Σχήμα 2-1 : Block Διάγραμμα Αναπτυξιακού Συστήματος XUP Virtex-II Pro

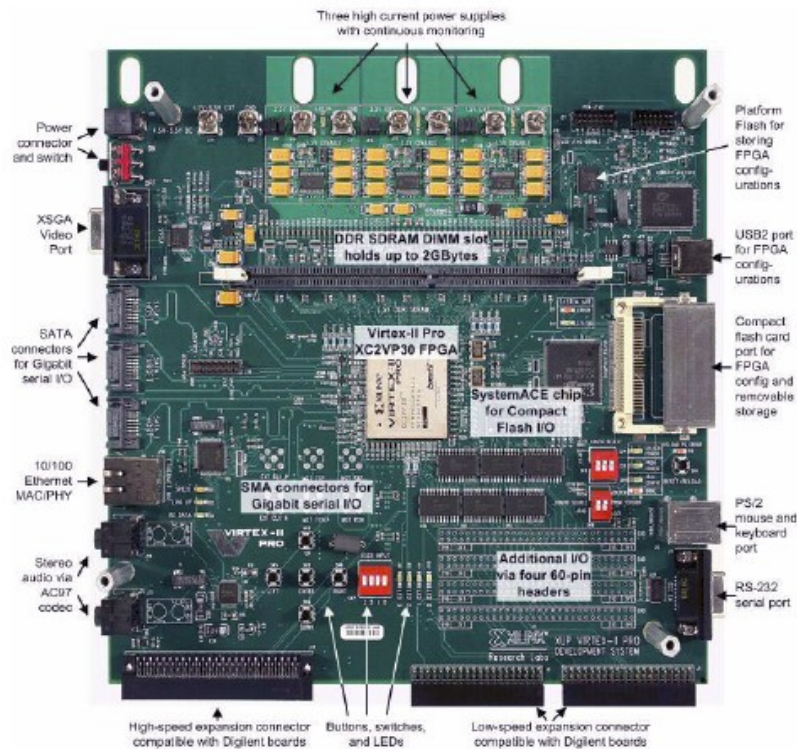
## Virtex-II Pro FPGA



Πρόκειται για μια Virtex-II Pro συσκευή FPGA πακεταρισμένη σε ένα flip-chip-fine-pitch FF896 BGA πακέτο. Δύο διαφορετικής χωρητικότητας FPGAs μπορούν να χρησιμοποιηθούν στο αναπτυξιακό σύστημα XUP Virtex-II Pro χωρίς αλλαγή στη λειτουργικότητα. Ο Πίνακας 2-1 αναγράφει τα γνωρίσματα της συσκευής Virtex-II Pro .

Γνωρίσματα	XC2VP20	XC2VP30
Slices	9280	13969
Array Size	56 x 46	80 x 46
Distributed RAM	290 Kb	428 Kb
Multiplier Blocks	88	136
Block RAMs	1584 Kb	2448 Kb
DCMs	8	8
PowerPC RISC Cores	2	2
Multi-Gigabit Transceivers	8	8

Πίνακας 2-1: Γνωρίσματα Συσκευής XC2VP20 και XC2VP30



Σχήμα 2-2 : Φωτογραφία Πλακέτας Αναπτυξιακού Συστήματος XUP Virtex-II Pro

### Τροφοδοσία και ρύθμιση της FPGA

Η XUP Virtex-II Pro αναπτυξιακή πλακέτα τροφοδοτείται από τροφοδοτικό ρυθμισμένο στα 5V. On-board εναλλακτικά τροφοδοτικά γεννούν ρεύμα τάσης 3.3V, 2.5V, και 1.5V για την FPGA, και περιφερειακά εξαρτήματα και γραμμικοί ρυθμιστές τροφοδοτούν τους MGTs. Η πλακέτα έχει παροχή για μέτρηση όλων των ψηφιακών τροφοδοσιών της FPGA, όπως επίσης δυνατότητα εφαρμογής εξωτερικής τροφοδοσίας αν η χωρητικότητα των εναλλακτικών on-board τροφοδοτικών ξεπεραστεί.

Το XUP Virtex-II Pro αναπτυξιακό σύστημα παρέχει αρκετές μεθόδους για τη ρύθμιση της FPGA Virtex-II Pro. Τα δεδομένα ρύθμισης μπορούν να προέλθουν από την εσωτερική Platform Flash PROM (δύο δυνατές ρυθμίσεις), το εσωτερικό CompactFlash μέσω αποθήκευσης (οκτώ δυνατές ρυθμίσεις), και εξωτερικές ρυθμίσεις που παραδίδονται από το Platform Cable USB ενσωματωμένων ή από τη διεπαφή παράλληλης θύρας.

### Multi-Gigabit Transceivers

Τέσσερις από τους οκτώ Multi-Gigabit Transceivers (MGTs) που είναι παρόντες στην FPGA Virtex-II Pro βγαίνουν έξω σε υποδοχές διασύνδεσης (connectors) και μπορούν να χρησιμοποιηθούν από το χρήστη. Τρία από τα αμφίδρομα MGT κανάλια τερματίζουν σε υποδοχές διασύνδεσης (connectors) Serial Advanced Technology Attachment (SATA) και το τέταρτο κανάλι τερματίζει σε παρεχόμενες από το χρήστη υποδοχές διασύνδεσης (connectors) Sub-Miniature A (SMA). Οι MGT πομποδέκτες είναι εφοδιασμένοι με πηγή ρολογιού 75 MHz ανεξάρτητη από το ρολόι συστήματος για την υποστήριξη βασικών επικοινωνιών SATA. Μια πρόσθετη πηγή ρολογιού MGT είναι διαθέσιμη μέσω υποδοχών διασύνδεσης (connectors) διαφορικού ζεύγους (SMA) παρεχόμενου από το χρήστη. Δύο από τις θύρες με υποδοχές διασύνδεσης (connectors) SATA είναι διαμορφωμένες ως Host θύρες και η τρίτη SATA θύρα είναι διαμορφωμένη ως Target θύρα ώστε να επιτρέπει απλή δικτύωση πλακέτας-προς-πλακέτα.

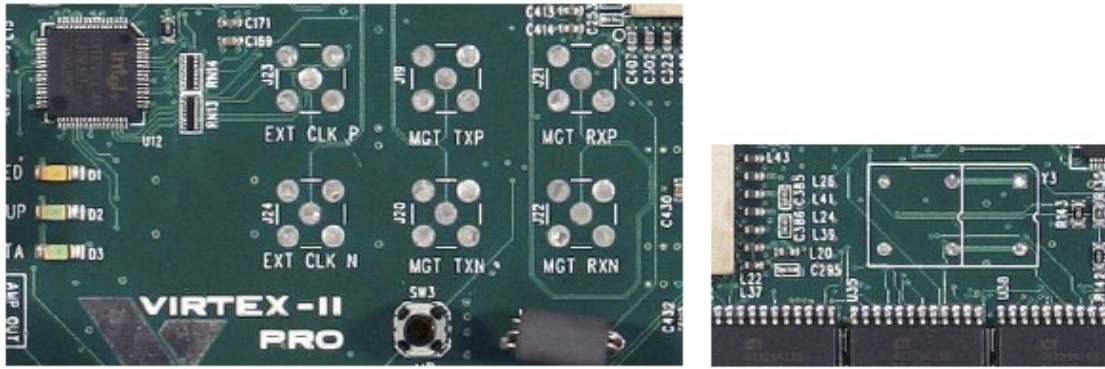
### Δημιουργία και κατανομή του ρολογιού

Το XUP Virtex-II Pro αναπτυξιακό σύστημα υποστηρίζει έξι πηγές ρολογιού:

- Ένα ρολόι συστήματος 100 MHz (Y2),
- Ένα ρολόι 75 MHz για τους MGTs που λειτουργούν τις Serial Advanced Technology Attachment (SATA) θύρες,
- Ένα διπλού ίχνους μέσω-οπής παρεχόμενο-από-χρήστη εναλλακτικό ρολόι (Y3),
- Ένα εξωτερικό ρολόι για τους MGTs (J23-J24),
- Ένα ρολόι 32 MHz (Y4) για τις διεπαφές System ACE, και
- Ένα ρολόι από το υψηλών-ταχυτήτων μέλος επέκτασης της Digilent.

Το ρολόι 75 MHz του SATA αποκτιέται από έναν υψηλής σταθερότητας (20 ppm) 3.3V LVDSL ταλαντωτή διαφορικής εξόδου, και το εξωτερικό ρολόι των MGT αποκτιέται από δύο παρεχόμενες-από-χρήστη υποδοχές διασύνδεσης (connectors) SMA. Οι υπόλοιποι τρεις ταλαντωτές είναι όλοι μονοτερματικές πηγές 3.3V LVTTTL. Κάθε ταλαντωτής είναι εφοδιασμένος με φίλτρο τροφοδοσίας για να μειώσει το θόρυβο στις εξόδους των ρολογιών. Για να επωφεληθεί ο χρήστης από τις εξωτερικές διαφορικές εισόδους ρολογιού, δύο υποδοχές διασύνδεσης (connectors) SMA πρέπει να εγκατασταθούν στα J23 και J24. Το Σχήμα 2-3 επισημαίνει την τοποθεσία των εξωτερικών διαφορικών εισόδων ρολογιού. Η εναλλακτική είσοδος ρολογιού αποκτιέται από τον παρεχόμενα-από-χρήστη ταλαντωτή 3.3V. Η χάραξη στην πλακέτα του τυπωμένου κυκλώματος υποστηρίζει είτε έναν πλήρους μεγέθους (21mm x 13mm) είτε μισού μεγέθους (13mm x 13mm) διαμέσης-οπής ταλαντωτή. Το Σχήμα 2-3 επισημαίνει την τοποθεσία της εισόδου του εναλλακτικού ταλαντωτή ρολογιού.





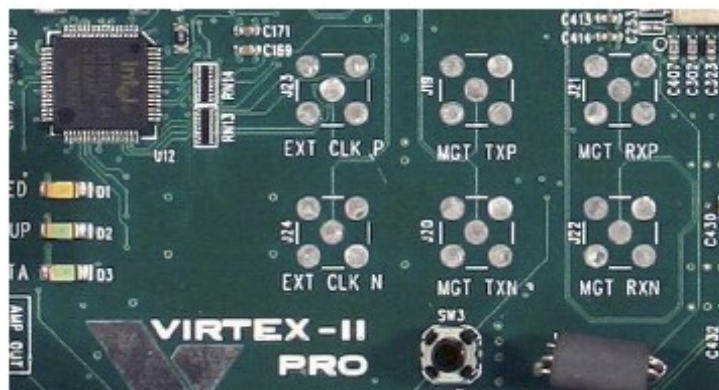
Σχήμα 2-3 : Εξωτερικοί Διαφορικοί Είσοδοι Ρολογιού κι Εναλλακτικός Ταλαντωτής Ρολογιού Εισόδου

Χρησιμοποιώντας τους Multi-Gigabit Πομποδέκτες (Transceivers)

Ο ενσωματωμένος πυρήνας RocketIO™ multi-gigabit πομποδέκτη είναι βασισμένος στην τεχνολογία Mindspeed SkyRail™. Οκτώ πυρήνες πομποδεκτών είναι διαθέσιμοι σε κάθε FPGA που μπορεί να χρησιμοποιηθεί στο XUP Virtex-II Pro Αναπτυξιακό Σύστημα. Ο πυρήνας του πομποδέκτη είναι σχεδιασμένος να δουλεύει σε κάθε ρυθμό baud rate στο εύρος από 622 Mb/s έως 3.125 Gb/s ανά κανάλι. Μόνο τέσσερα από τα διαθέσιμα οκτώ κανάλια χρησιμοποιούνται στο XUP Virtex-II Pro Αναπτυξιακό Σύστημα. Τρία κανάλια είναι εξοπλισμένα με χαμηλού-κόστους SATA υποδοχές διασύνδεσης (connectors) και το τέταρτο κανάλι τερματίζει σε παρεχόμενες-από-χρήστη Sub-Miniature A (SMA) υποδοχές διασύνδεσης (connectors). Τα κανάλια SATA είναι χωρισμένα σε δύο μορφές διεπαφών, δύο θύρες HOST (J16, J18), και μια θύρα TARGET (J17). Η θύρα TARGET εναλλάσσει τα διαφορικά ζεύγη αποστολής και λήψης ώστε να επιτρέψει τη σύνδεση δύο XUP Virtex-II Pro Αναπτυξιακών Συστημάτων ως ένα απλό δίκτυο, ή πολλαπλών XUP Virtex-II Pro Αναπτυξιακών Συστημάτων σε δίκτυο τοπολογίας δακτυλίου. Οι προδιαγραφές του SATA προϋποθέτουν μια κατάσταση σήματος εκτός ζώνης για να χρησιμοποιείται όταν το κανάλι θα είναι αδρανές (idle). Αυτή η ικανότητα δεν παρέχεται άμεσα από τους MGT. Δύο αντιστάσεις, ένα τρανζίστορ FET, και δύο συζευγμένοι πυκνωτές εναλασσόμενου, μαζί με τα ειδικά σήματα ελέγχου της κατάστασης αδράνειας προσθέτουν την ικανότητα εκτός ζώνης σηματοδότησης της κατάστασης IDLE στους MTG. Πρόσθετο εκτός πλακέτας υλικό μπορεί να χρειαστεί για την κατάλληλη διεπαφή με κανονικούς δίσκους SATA.

Το τέταρτο ζεύγος καναλιού MGT καταλήγει σε παρεχόμενες-από-χρήστη υποδοχές διασύνδεσης (connectors) SMA (J19-22) και μπορεί να οδηγηθεί από παρεχόμενο-από-χρήστη διαφορικό ζεύγος εισόδου ρολογιού, το EXTERNAL\_CLOCK\_P και EXTERNAL\_CLOCK\_N που παρέχονται σε υποδοχές διασύνδεσης (connectors) SMA (J23-24). Αυτό το EXTERNAL\_CLOCK μπορεί να χρησιμοποιηθεί για να δώσει ρολόι στις θύρες SATA αν χρειάζονται πέραν των συνηθισμένων ρυθμοί σηματοδότησης. Οι υποδοχές διασύνδεσης (connectors) MGT φαίνονται στο Παράρτημα Πίνακας A-2.

Για να επωφεληθεί ο χρήστης από το τέταρτο κανάλι MGT, τέσσερις υποδοχές διασύνδεσης (connectors) SMA πρέπει να εγκατασταθούν στα J19-J22. Το Σχήμα 2-4 επισημαίνει την θέση των εισόδων εξωτερικού διαφορικού ρολογιού.



Σχήμα 2-4 : Συνδέσεις MGT βασισμένες σε SMA

Υπάρχουν οκτώ εισοδοί ρολογιού μέσα σε κάθε στιγμιότυπο πομποδέκτη RocketIO™. Τα REFCLK και BREFCLK είναι ρολόγια αναφοράς που έχουν δημιουργηθεί από μια εξωτερική πηγή και έχουν εισαχθεί στην FPGA ως διαφορικές εισοδοί. Τα ρολόγια αναφοράς συνδέονται στις θύρες REFCLK ή BREFCLK πάνω στον RocketIO MGT. Ενώ μόνο ένα από αυτά τα ρολόγια αναφοράς χρειάζεται για να οδηγήσει τον MGT, το BREFCLK ή BREFCLK2 χρησιμοποιούνται για σειριακές ταχύτητες από 2.5 Gb/s και πάνω. Για ταχύτητες από 2.5 Gb/s και πάνω, η διαμόρφωση του REFCLK εισάγει περισσότερο από το μέγιστο αποδεκτό θόρυβο (παράσιτα) στον πομποδέκτη RocketIO. Αυτό γιατί για αυτές τις υψηλότερες ταχύτητες, χρειάζεται η διαμόρφωση του BREFCLK. Η διαμόρφωση του BREFCLK χρησιμοποιεί πόρους αποκλειστικής δρομολόγησης που ελαττώνουν το θόρυβο. Το BREFCLK εισάγεται στην FPGA μέσω αποκλειστικού buffer εισόδου ρολογιού. Το BREFCLK μπορεί να συνδεθεί στις εισόδους του BREFCLK του MGT και στην είσοδο CLKIN ενός DCM για τη δημιουργία ρολογιών χρηστών. Περισσότερα στην ενότητα “Ρολοί και σηματοδοσία ρολογιού” σελ. 28.

Ο ρυθμός δεδομένων του SATA είναι λιγότερο από 2.5 Gb/s οπότε τα ρολόγια των 75 MHz θα μπορούσαν να συνδέονται στις εισόδους του REFCLK, αλλά για λόγους συνέπειας οι εισοδοί ρολογιού του BREFCLK και του BREFCLK2 χρησιμοποιούνται για τα ενσωματωμένα και παρεχόμενα-από-χρήστη ρολόγια των MGTs.

Οι RocketIO MGTs χρησιμοποιούν διαφορική σηματοδοσία μεταξύ των θυρών αποστολής και λήψης δεδομένων ώστε να ελαχιστοποιήσουν τα αποτελέσματα του θορύβου κοινής λειτουργίας και το crosstalk σήματος. Με τη χρήση σειριακών πομποδεκτών υψηλών-ταχυτήτων, τα μέσα μετάδοσης, όπως είναι τα καλώδια, προκαλούν υποβάθμιση του σήματος που προορίζεται για τον παραλήπτη. Παράγονται, λοιπόν, φαινόμενα όπως το inter-symbol interference (ISI) ή data dependent jitter. Αυτή η απώλεια μπορεί να είναι αρκετά μεγάλη ώστε να υποβαθμίσει το πρότυπο ανοίγματος ματιού στον παραλήπτη πέρα από το σημείο που επιτρέπει την αξιόπιστη αναμετάδοση δεδομένων. Οι RocketIO MGTs επιτρέπουν στον χρήστη να θέσει την αρχική διαφορική τάση ταλάντωσης και προ-έμφαση του σήματος για να αντιστραφεί ένα μέρος της υποβάθμισης του σήματος ώστε να αυξηθεί η αξιοπιστία της μετάδοσης δεδομένων. Στην προ-έμφαση, η αρχική διαφορική τάσης ταλάντωσης ενισχύεται ώστε να δημιουργηθεί μια ισχυρότερα αυξανόμενη ή φθίνουσα κυματομορφή. Αυτή η μέθοδος αντισταθμίζει την απώλεια στις υψηλές συχνότητες στο μέσο μετάδοσης, το οποίο αλλιώς θα περιόριζε το πλάτος της ληφθείσας κυματομορφής. Η αρχική διαφορική ταλάντωση τάσης και προ-έμφαση σήματος ορίζονται από δύο παρεχόμενα-από-χρήστη γνωρίσματα του RocketIO.

Το γνώρισμα TX\_DIFF\_CTRL θέτει τη διαφορά τάσης μεταξύ των διαφορικών γραμμών, και το γνώρισμα TX\_PREAMPHASIS θέτει την προ-έμφαση του οδηγού εξόδου. Περισσότερα στην ενότητα “Τεχνική Προ-έμφασης”, σελ. 40.

### Αρχείο Περιορισμών Χρήστη (User Constraint Files UCF)

Το αρχείο περιορισμών χρήστη (User Constraint File UCF), με απλά λόγια, αντιστοιχίζει τη δικτύωση (nets) της σχεδίασης του χρήστη με ακροδέκτες (pins) του υλικού της πλακέτας του αναπτυξιακού συστήματος. Αυτό έχει σαν αποτέλεσμα τον ακριβή προσδιορισμό της δρομολόγησης των σημάτων, τον κατάλληλο προσδιορισμό των τοποθεσιών των ακροδεκτών εισόδου/εξόδου για τα σήματα της FPGA Virtex-II Pro, όπως επίσης τη ρύθμιση των επιπέδων διακόπτη εισόδου-εξόδου, την ομαδοποίηση σημάτων της πλακέτας και την αναφορά τους σε συγκεκριμένα τμήματα (slices) της πλακέτας καθώς και τον ορισμό των slew rates για κάθε σήμα. Οι πληροφορίες του αρχείου UCF είναι σπασμένες κατά συνάρτηση και μόνο οι τομείς που χρειάζονται για τη σχεδίαση του χρήστη πρέπει να συμπεριληφθούν στο αρχείο UCF για την πραγματική σχεδίαση. Αξίζει να σημειωθεί ότι μέσω αυτού του αρχείου ο χρήστης επεμβαίνει άμεσα στο υλικό συνεπώς πρέπει να ορίζει με σαφήνεια και σιγουριά τις διάφορες παραμέτρους γιατί είναι δυνατή η καταστροφή κυκλωμάτων της πλακέτας σε περίπτωση σφάλματος του χρήστη.

## 2.2 Ανάλυση των χαρακτηριστικών και λειτουργιών του RocketIO

### 2.2.1 Επισκόπηση

Στην ενότητα αυτή αναλύεται το πιο κρίσιμο κύκλωμα του συστήματος αυτού, που υλοποιεί τη σειριοποίηση κι αποσειριοποίηση δεδομένων κι εκτελεί τις κυρίως λειτουργίες της εγκαθίδρυσης καναλιού καθώς κι αποστολής και λήψης δεδομένων. Επίσης, μελετούνται τα κυκλώματα σηματοδοσίας ρολογιού και προ-έμφασης (pre-emphasis) που περιλαμβάνει.

### 2.2.2 Λίγα λόγια για το Rocket I/O

Οι πομποδέκτες Rocket I/O διαθέτουν την τεχνολογία ανάπτυξης οποιουδήποτε προτύπου σειριακής Εισόδου/Εξόδου. Υπάρχει δυνατότητα ένωσης πολλαπλών Rocket I/O blocks (συγχρονισμένα) σε δεσμό ώστε να διαμορφωθεί κανάλι δεδομένων υψηλότερου εύρους ζώνης.

### Λειτουργίες RocketIO

Το κύκλωμα RocketIO έχει ευέλικτες, προγραμματιζόμενες λειτουργίες που επιτρέπουν σε ένα σειριακό πομποδέκτη multi-gigabit να ενσωματωθεί εύκολα σε όποια σχεδίαση σε Virtex-II Pro. Αυτές είναι:

- Ποικιλία-ταχύτητας, full-duplex πομποδέκτης, επιτρέπει από 600 Mb/s μέχρι 3.125 Gb/s baud ρυθμό μεταφοράς
- Λιθογραφικό σύστημα, αποκλειστικής δρομολόγησης πάνω στην πλακέτα, σύνθεσης κι ανάκτησης ρολογιού (clock recovery), που εξαλείφει την ανάγκη για εξωτερικά κομμάτια
- Αυτόματη συνάρτηση lock-to-reference για τα σήματα ρολογιού
- Πέντε επίπεδα προγραμματιζόμενης σειριακής εξόδου διαφορικής ταλάντωσης (800 mV ως 1600 mV peak-peak), επιτρέποντας συμβατότητα με επίπεδα τάσης άλλων σειριακών συστημάτων
- Τέσσερα επίπεδα προγραμματιζόμενης προ-έμφασης
- Σύζευξη AC και DC
- Προγραμματιζόμενος 50Ω/75Ω τερματισμός στο chip, που εξαλείφει την ανάγκη για εξωτερικές αντιστάσεις τερματισμού
- Σειριακό και παράλληλο TX-σε-RX με λειτουργία βρόχου εσωτερικής αναδρομής για έλεγχο ορθής λειτουργίας υλικού
- Προγραμματιζόμενος εντοπισμός κόμματος που επιτρέπει για κάθε πρωτόκολλο εντοπισμό κάθε χαρακτήρα 10-bit.

### RocketIO Πομποδέκτης Βασική Αρχιτεκτονική και Γνωρίσματα

Ο πομποδέκτης RocketIO βασίζεται στην τεχνολογία Mindspeed's SkyRail™. Στο Σχήμα 2-5, εικονίζεται ένα ολικό block Διάγραμμα του πομποδέκτη. Μέχρι 20 λειτουργικά τμήματα μεταδοτών είναι διαθέσιμα σε μια FPGA Virtex-II Pro, εξαρτάται από το κομμάτι που χρησιμοποιείται.

Το λειτουργικό τμήμα του πομποδέκτη είναι σχεδιασμένο να λειτουργεί σε κάθε σειριακό ρυθμό bit στο εύρος από 600 Mb/s ως 3.125 Gb/s ανά κανάλι, συμπεριλαμβανομένων των συγκεκριμένων ρυθμών bit που χρησιμοποιούνται από τα πρότυπα επικοινωνιών στον Πίνακα 2-2.

Λειτουργία	Κανάλια (Γραμμές) <sup>(1)</sup>	Ρυθμός Bit Εισόδου/Εξόδου (Gb/s)
Κανάλι οπτικής ίνας	1	1.06 2.12
Gbit Ethernet	1	1.25
PCI Express <sup>(2)</sup>	1	2.5
XAUI (10-Gbit Ethernet)	4	3.125
XAUI (10-Gbit Κανάλι Οπτικής Ίνας) <sup>(3)</sup>	4	3.1875 <sup>(4)</sup>
Infiniband	1,4,12	2.5
Aurora (Πρωτόκολλο Xilinx)	1,2,3,4,...	0.600 – 3.125
Αυτοσχέδια Λειτουργία	1,2,3,4,...	0.600 – 3.125

Πίνακας 2-2 : Πρότυπα Επικοινωνίας που Υποστηρίζονται από τον Πομποδέκτη RocketIO

Σημείωση:

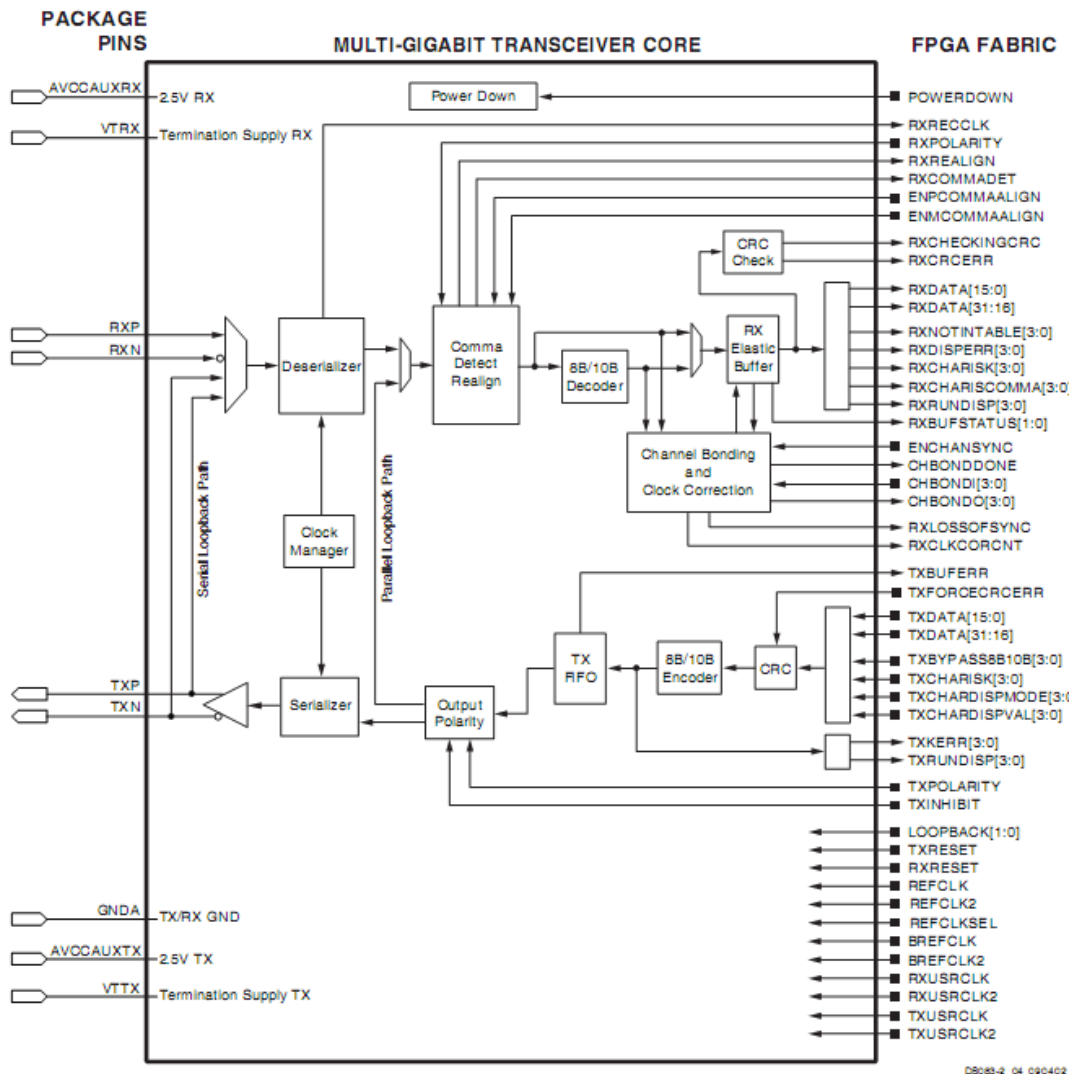
1. Ένα κανάλι θεωρείται ένας πομποδέκτης.
2. Εκτός-ζώνης (Out-of-Band OOB) σήματα δεν υποστηρίζονται με τον πομποδέκτη.
3. Υποστηρίζεται με το GT\_CUSTOM primitive. Ορισμένα χαρακτηριστικά πρέπει να τροποποιηθούν ώστε να εναρμονίζονται με τις προδιαγραφές XAUI 10GFC, συμπεριλαμβανομένου και όχι μόνο του CLK\_COR\_SEQ και CHAN\_BOND\_SEQ.

4. Ρυθμός Bit rate είναι πιθανός με τις ακόλουθες προδιαγραφές τοπολογίας: μέγιστο 6" FR4 και μια υποδοχή διασύνδεσης (connector) Molex 74441.
5. Πλάτος δεδομένων ενός, δύο και τεσσάρων bytes μπορούν να επιλεγθούν για κάθε πρωτόκολλο

Ο σειριακός ρυθμός bit δεν χρειάζεται να διαμορφωθεί στο πομποδέκτη, αφού η συχνότητα λειτουργίας υπαγορεύεται από τα ληφθέντα δεδομένα, το ρολόι αναφοράς που εφαρμόζεται, και το χαρακτηριστικό SERDES\_10B (βλέπε Πίνακα 2-3).

SERDES_10B	Serial Baud Rate
FALSE	1.0 Gb / s – 3.125 Gb / s
TRUE	600 Mb / s – 1.0 Gb / s

Πίνακας 2-3: Σειριακοί Ρυθμοί Baud και η Αντίστοιχη Παράμετρος Ελέγχου, SERDES\_10B



Σχήμα 2-5 : Block Διάγραμμα του πομποδέκτη RocketIO

Υπάρχουν δυο τρόποι τροποποίησης του πομποδέκτη RocketIO:

- Στατικές ιδιότητες μπορούν να ρυθμιστούν μέσω χαρακτηριστικών σε κώδικα HDL.



- Δυναμικές αλλαγές μπορούν να γίνουν από τις θύρες των θεμελιωδών δομών (primitives).

Ο πομποδέκτης RocketIO αποτελείται από τα Physical Media Attachment (PMA) και Physical Coding Sublayer (PCS). Το PMA περιέχει τον serializer/deserializer (σειριοποιητή/αποσειριοποιητή) (SERDES), TX και RX buffers, γεννήτρια ρολογιού, και κύκλωμα ανάκτησης ρολογιού. Το PCS περιέχει τον κωδικοποιητή/αποκωδικοποιητή 8B/10B και τη δημιουργία δεσμού καναλιών που υποστηρίζει ελαστικό buffer και τη διόρθωση ρολογιού. Το PCS επίσης χειρίζεται το Cyclic Redundancy Check (CRC). Αναφέρεται ξανά στο Σχήμα 2-5, που δείχνει το block diagram επιπέδου top-level και τα σήματα διεπαφής της FPGA του πομποδέκτη RocketIO.

### Στιγμιότυπα Πομποδέκτη RocketIO

Για διαφορετικά πλάνα παροχής ρολογιού, αρκετά πράγματα πρέπει να αλλάξουν, περιλαμβανομένης της συχνότητας ρολογιού για USRCLK και USRCLK2. Οι θύρες δεδομένων και ελέγχου για το GT\_CUSTOM πρέπει επίσης να αντικατοπτρίζουν αυτήν την αλλαγή σε πλάτος δεδομένων συγχωνεύοντας μηδενικά πάνω σε εισόδους και καλώδια για εξόδους για σχεδιάσεις σε Verilog, και θέτοντας εξόδους ανοικτές και συγχωνεύοντας μηδενικά στα χρησιμοποιήσιμα bits εισόδου για σχεδιάσεις σε VHDL.

### 2.2.3 Θέματα Ψηφιακού Σχεδιασμού (Digital Design Considerations)

#### Μεθοδολογία Κωδικοποίησης Γραμμής Μετάδοσης στο Rocket I/O

Οι μεθοδολογίες κωδικοποίησης γραμμής μετάδοσης μετατρέπουν τα ακατέργαστα δεδομένα σε μορφή αποδεκτή από το δέκτη. Πιο συγκεκριμένα, η μεθοδολογία κωδικοποίησης γραμμής μετάδοσης εξασφαλίζει ότι διαδραματίζονται αρκετές μεταβάσεις στο σήμα για την εύρυθμη λειτουργία του κυκλώματος ανάκτησης ρολογιού. Επίσης, μέσω της κωδικοποίησης παρέχεται μια μέθοδος ευθυγράμμισης των δεδομένων σε λέξεις με ισορροπημένη παροχή DC ρεύματος στη γραμμή μετάδοσης. Προαιρετικά, η τεχνική αυτή προσφέρεται για τη διαδικασία της διόρθωσης ρολογιού, συγχρονισμό των πυρήνων και δημιουργία δεσμών καναλιών, καθώς και διαίρεση του εύρους ζώνης σε υπο-κανάλια. Υπάρχουν δυο κυρίως τεχνικές κωδικοποίησης γραμμής μετάδοσης, αυτή της αναζήτησης τιμής και αυτή της μετατροπής των ροών αυτής με χρήση κρυπτογράφων (scramblers).

#### 8B/10B Κωδικοποίηση/Αποκωδικοποίηση

##### Γενικά

Ο πομποδέκτης RocketIO έχει την ικανότητα κωδικοποίησης οκτώ bits μέσα σε σειριακή ακολουθία δέκα-bit, χρησιμοποιώντας το πρότυπο κωδικοποίησης 8B/10B. Αυτό με απλά λόγια σημαίνει ότι 8 bits δεδομένων συμβολίζονται με 10 bits δεδομένων με σκοπό τη μεταφορά τους μέσω της γραμμής μετάδοσης. Η κωδικοποίηση αυτή προσφέρει επιπλέον σύμβολα πέραν τον αποδεκτών ascii χαρακτήρων που χρησιμοποιεί ο χρήστης τα οποία θα χρησιμοποιεί μόνο η σχεδίαση για εσωτερικές λειτουργίες όπως η αποστολή χαρακτήρων idle, χαρακτήρων αρχής και τέλους πακέτου, χαρακτήρων pad. Επίσης η κωδικοποίηση αυτή εγγυάται ομαλή ηλεκτρομαγνητική ένταση κατά τη διάρκεια της μεταφοράς δεδομένων. Αυτό το επιτυγχάνει αυξάνοντας τον αριθμό των

μονάδων 1, έτσι ώστε ο αριθμός τους σε κάθε πακέτο να είναι ίσος με αυτόν των 0, ή να διαφέρει κατά δύο. Ακόμα πραγματοποιείται η επιτυχής ανίχνευση του σήματος από τη μεριά του δέκτη λόγω της ιδιότητας που έχει να αλλάζει συνεχώς το σήμα. Έτσι το σήμα είναι εύκολα ανιχνεύσιμο, αφού δεν είναι δυνατόν το ίδιο σύμβολο (1 ή 0) να επαναλαμβάνεται για πάνω από 5 χρονικά διαστήματα. Εν κατακλείδι, η κωδικοποίηση αυτή εγγυάται μια ισορροπημένη κατά DC, πλούσια σε ακμές σειριακή ροή, διευκολύνει την σύζευξη DC ή AC και την ανάκτηση ρολογιού. Τέλος αξίζει να αναφερθεί ότι υπάρχει δυνατότητα χρήσης και της κωδικοποίησης 64B/66B όπως και της 4B/5B. Η κωδικοποίηση 4B/5B είναι παρόμοια με την κωδικοποίηση 8B/10B όσον αφορά την επιπλέον πληροφορία (overhead). Όμως παρόλο που παρέχει απλούστερους κωδικοποιητές και αποκωδικοποιητές, περιλαμβάνει περιορισμένο αριθμό χαρακτήρων ελέγχου και δεν προσφέρει ισορροπία του ρεύματος DC ούτε κι ασχολείται με το θέμα του disparity (ανομοιότητα). Συνεπώς έχει μειωμένη λειτουργικότητα. Η κωδικοποίηση 64B/66B εισάγει μειωμένη επιπλέον πληροφορία (overhead) σε σχέση με την 8B/10B. Όμως αυτό έχει ως τίμημα μεγαλύτερο χρόνο ευθυγράμμισης, την πιθανότητα εμφάνισης μιας μικρής DC συνιστώσας, καθώς και πιο πολύπλοκους κωδικοποιητές και αποκωδικοποιητές. Ακόμα υπάρχει και κόστος πολυπλοκότητας όσον αφορά τη ρύθμιση και χρήση των αντίστοιχων κωδικοποιητών. Με βάση την παραπάνω τεκμηρίωση χρησιμοποιήθηκε η κωδικοποίηση 8B/10B.

### *Running Disparity*

Η ισορροπία του ρεύματος DC στην κωδικοποίηση 8B/10B επιτυγχάνεται μέσω μιας τεχνικής που καλείται running disparity. Ο ευκολότερος τρόπος για την επίτευξη ισορροπίας DC θα ήταν να επιτρέπονται μόνο σύμβολα που διαθέτουν τον ίδιο αριθμό άσσων και μηδενικών, αλλά κάτι τέτοιο θα περιόριζε τον αριθμό των συμβόλων. Αντί αυτού, η 8B/10B χρησιμοποιεί δύο διαφορετικά σύμβολα που ανατίθενται σε κάθε τιμή των δεδομένων. Στις περισσότερες των περιπτώσεων, ένα από τα σύμβολα έχει τέσσερα μηδενικά και έξι άσσους. Ο συνολικός αριθμός μηδενικών και άσσων παρακολουθείται και έτσι το επόμενο σύμβολο επιλέγεται με βάση του τι χρειάζεται ώστε να επανέλθει η ισορροπία DC ξανά στη γραμμή μετάδοσης. Τα δύο σύμβολα αποκαλούνται κοινώς, θετικά (+) κι αρνητικά (-) σύμβολα. Για καλύτερη κατανόηση παρατίθεται σχετικό παράδειγμα.

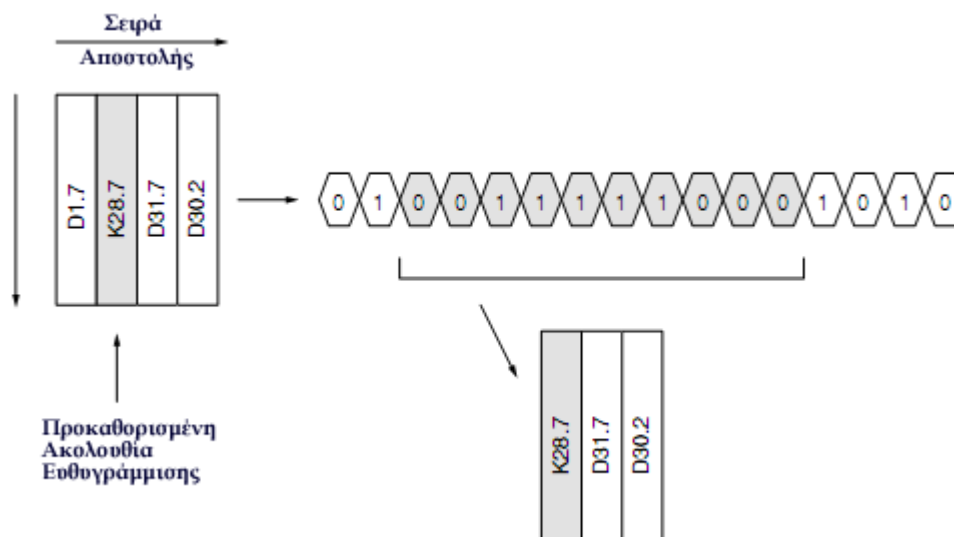
Ονομασία	Hex	8 Bits	RD -	RD +
D10.7	EA	11101010	0101011110	0101010001
D31.7	FF	11111111	1010110001	0101001110
D4.5	A4	10100100	1101011010	0010101010
D0.0	00	00000000	1001110100	0110001011
D23.0	17	00010111	1110100100	0001011011

Πίνακας 2-4 : Αντιστοιχία των Ακολουθιών Δεδομένων 8 bit σε Δύο Σύμβολα 10 bit

Ένα επιπλέον κέρδος της running disparity είναι το γεγονός ότι ο δέκτης μπορεί να παρακολουθήσει την running disparity και να εντοπίσει σφάλμα στην ροή εισερχόμενων δεδομένων γιατί οι κανόνες της μεθόδου θα έχουν παραβιαστεί. Ως απόρροια των συμφραζόμενων η κωδικοποίηση 8B/10B περιλαμβάνει σύνολο από χαρακτήρες δεδομένων και χαρακτήρες-K. Περισσότερα αναφέρονται παρακάτω και στο Παράρτημα, Πίνακας A-3.

### Εντοπισμός Κομμάτων

Τα σύμβολα διαχωρίζονται μεταξύ τους με ένα κόμμα που σηματοδοτεί και τα όριά τους. Συγκεκριμένα, το κόμμα είναι ένα ή δύο σύμβολα που καλούνται και ακολουθία ευθυγράμμισης καθώς χρησιμεύουν στην ευθυγράμμιση των δεδομένων. Αυτή η ακολουθία συνήθως ορίζεται στον πομποδέκτη, αλλά σε κάποιες περιπτώσεις μπορεί να είναι προκαθορισμένη. Ο δέκτης ανιχνεύει τη ροή εισερχόμενων δεδομένων για τη καθορισμένη ακολουθία bit. Αν εντοπιστεί η ακολουθία, ο αποσειριοποιητής επανατοποθετεί τα όρια της λέξης ώστε να ταιριάζουν στην εντοπισμένη ακολουθία κόμμα. Πρόκειται για συνεχόμενη σάρωση. Μόλις έχει πραγματοποιηθεί η ευθυγράμμιση, όλα τα επόμενα κόμματα που εντοπίστηκαν θα βρουν την ευθυγράμμιση να έχει ήδη γίνει. Φυσικά, η ακολουθία κόμμα πρέπει να είναι μοναδική σαν ακολουθία. Για παράδειγμα, αν χρησιμοποιείται ένα σύμβολο σήματος c για το κόμμα, τότε θα πρέπει να εξασφαλίζεται ότι κανένα διατεταγμένο σύνολο δεδομένων xy δεν περιέχει την ακολουθία c.



Σχήμα 2-6 : Έγκυρα Δεδομένα Σειριακής Ροής 8B/10B

Επιπλέον, η χρήση προκαθορισμένου πρωτοκόλλου δεν αποτελεί πρόβλημα καθώς οι χαρακτήρες κόμμα έχουν ήδη οριστεί. Χρησιμοποιούνται κυρίως ένας οι περισσότεροι ανάμεσα σε ένα ειδικό σύνολο χαρακτήρων-K που αποτελείται από τους K28.1, K28.5, και K28.7, όλοι εκ των οποίων διαθέτουν ως τα πρώτα επτά bits τα 1100000. Αυτό το μοτίβο βρίσκεται μόνο στους συγκεκριμένους χαρακτήρες και κανένα άλλο διατεταγμένο σύνολο δεδομένων ή ακόμα και άλλοι χαρακτήρες-K δεν θα περιείχαν ποτέ αυτή την ακολουθία. Συνεπώς, είναι ιδανική για το έργο της ευθυγράμμισης. Σε περιπτώσεις κατασκευής ενός αυτοσχέδιου πρωτοκόλλου, η πιο ενδεδειγμένη λύση είναι ο δανεισμός μιας ακολουθίας κόμματος ενός δημοφιλούς πρωτοκόλλου. Το Gigabit Ethernet χρησιμοποιεί τον K28.5 ως κόμμα και για αυτό έχει επικρατήσει να αναφέρεται ως σύμβολο κόμματος. Τα ονόματα που χρησιμοποιούνται προκύπτουν από τις ανάγκες των κωδικοποιητών και αποκωδικοποιητών.

### Κρυπτογράφηση (Scrambling)



Η κρυπτογράφηση είναι μια μέθοδος αναδιάταξης ή κωδικοποίησης των δεδομένων έτσι ώστε να φαίνεται σα να είναι τυχαία, ωστόσο να μπορούν επίσης να αποκρυπτογραφηθούν. Χρειάζονται randomizers οι οποίοι σπάνε μεγάλες σειρές από μηδενικά και άσσους. Προφανώς, ο στόχος είναι η αποκρυπτογράφηση των bits να μην χρειάζεται επιπλέον πληροφορίες ευθυγράμμισης και αυτό το γνώρισμα αποτελεί την κωδικοποίηση που συγχρονίζεται από τον εαυτό της.

### *8B/10B Κωδικοποιητής*

Ένας κωδικοποιητής 8B/10B συμπεριλαμβάνεται στο πομποδέκτη και παρέχεται επίσης η δυνατότητα παράκαμψής του. Έχει ως λειτουργία την εφαρμογή του running disparity και της κωδικοποίησης χρησιμοποιώντας αν χρειαστεί και κρυπτογράφηση (scrambling). Ο πομποδέκτης χρησιμοποιεί τους ίδιους 256 χαρακτήρες δεδομένων και 12 χαρακτήρες ελέγχου (παρουσιάζονται στο Παράρτημα, “8B/10B Χαρακτήρες Ελέγχου”) που χρησιμοποιούνται για Gigabit Ethernet, XAUI, Κανάλι Οπτικής Ίνας, και InfiniBand.

Ο κωδικοποιητής δέχεται 8 bits δεδομένων μαζί με ένα σήμα ελέγχου χαρακτήρα-K. Αν το σήμα χαρακτήρα-K είναι High, τα δεδομένα κωδικοποιούνται σε έναν από τους δώδεκα πιθανούς χαρακτήρες-K που είναι διαθέσιμοι στην κωδικοποίηση 8B/10B. Αν η είσοδος χαρακτήρα-K είναι Low, τα 8 bits κωδικοποιούνται ως κανονικά δεδομένα. Αν η είσοδος χαρακτήρα-K είναι High και ένας χρήστης εφαρμόσει κάποιον άλλον από τους δώδεκα πιθανούς συνδυασμούς, το σήμα TXKERR υποδεικνύει το λάθος.

### *8B/10B Αποκωδικοποιητής*

Ένας αποκωδικοποιητής 8B/10B, συμπεριλαμβάνεται στο δέκτη και παρέχεται επίσης η δυνατότητα παράκαμψής του. Μια προγραμματιζόμενη λογική επιτρέπει την παράκαμψη του αποκωδικοποιητή. Ο αποκωδικοποιητής χρησιμοποιεί την ίδια μεθοδολογία αποκωδικοποίησης που χρησιμοποιείται για Gigabit Ethernet, Κανάλι Οπτικής Ίνας, και InfiniBand.

Ο αποκωδικοποιητής ξεχωριστά εντοπίζει δύο ειδών λάθη, τα «λάθη στο disparity» (disparity errors) και «λάθη εκτός-ζώνης» (out-of-band errors). Ένα λάθος disparity συμβαίνει όταν παραλαμβάνεται ένας 10-bit χαρακτήρας που υφίσταται στον πίνακα 8B/10B, αλλά έχει μια λάθος αντιστοιχία. Ένα εκτός-ζώνης λάθος συμβαίνει όταν παραλαμβάνεται ένας 10-bit χαρακτήρας που δεν υφίσταται μέσα στον πίνακα 8B/10B. Είναι δυνατόν να υπάρχει ένα εκτός-ζώνης σφάλμα χωρίς να υπάρχει σφάλμα ανομοιότητας. Η κατάλληλη αντιστοιχία υπολογίζεται πάντα και για τους δύο, τους νόμιμους και τους παράνομους χαρακτήρες. Η τρέχουσα ανομοιότητα είναι διαθέσιμη στο σήμα RXRUNDISP. Ο αποκωδικοποιητής 8B/10B εκτελεί μια μοναδική λειτουργία εφόσον εντοπιστεί ένα εκτός-ζώνης σφάλμα. Όταν αυτό συμβεί, ο αποκωδικοποιητής σηματοδοτεί το σφάλμα, περνάει μέσα τα παράνομα 10 bits, και τα τοποθετεί στις εξόδους. Αυτό μπορεί να χρησιμοποιηθεί για λόγους αποσφαλμάτωσης αν είναι επιθυμητό. Ο αποκωδικοποιητής επίσης σηματοδοτεί τη λήψη ενός από τους δώδεκα έγκυρους χαρακτήρες-K προς τη θύρα RXCHARISK.

Επιπλέον, συμπεριλαμβάνεται προγραμματιζόμενος εντοπισμός κόμματος. Το σήμα εντοπισμού κόμματος RXCOMMADET καταχωρεί ένα κόμμα με την παραλαβή όποιου εκ των θετικών (+), αρνητικών (-) ή και των δύο συμβόλων κόμματος. Εφόσον το κόμμα προσδιορίζεται ως ένας χαρακτήρας 7-bit, αυτό περιλαμβάνει αρκετούς χαρακτήρες εκτός ζώνης. Το RXCHARISCOMMA επιτρέπει στον κωδικοποιητή να εντοπίσει μόνο τα τρία διαφορετικά κόμματα (K28.1, K28.5, και K28.7) θετικά (+) κι αρνητικά (-)

σύμβολα κόμματος, ή και τα δυο. Συνολικά, υπάρχουν έξι πιθανές επιλογές, τρεις για έγκυρα κόμματα και τρεις για “οποιοδήποτε κόμμα”.

Σημειώτέον ότι όλα τα bytes (1, 2, or 4) στην διεπαφή της RX της FPGA έχει το καθένα τα δικά του ανεξάρτητα 8B/10B σήματα (χαρακτήρα-K, σφάλμα ανομοιότητας, εκτός-ζώνης σφάλμα, τρέχουσα αντιστοιχία, και εντοπισμός κόμμα).

Επανεκκίνηση/Τερματισμός

Ο παραλήπτης και ο αποστολέας έχουν τις δικές τους σύγχρονες εισόδους επανεκκίνησης. Η επανεκκίνηση του πομποδέκτη ξανακεντράει τη FIFO αποστολής, και επανεκκινεί όλους τους καταχωρητές αποστολής και τον κωδικοποιητή 8B/10B. Η επανεκκίνηση του παραλήπτη ξανακεντράει το ελαστικό buffer του παραλήπτη, κι επανεκκινεί όλους του καταχωρητές λήψης και τον αποκωδικοποιητή 8B/10B. Κανένα από τα σήματα επανεκκίνησης δεν έχει επιρροή στα PLL.

Αφού ενεργοποιηθεί το σήμα κλειδώματος του DCM, μπορούν να ενεργοποιηθούν τα σήματα επανεκκίνησης. Τα σήματα επανεκκίνησης πρέπει να ενεργοποιηθούν για δυο κύκλους του USRCLK2 ώστε να εξασφαλιστεί σωστή αρχικοποίηση των FIFO. Παρ’ όλο που και τα δύο σήματα επανεκκίνησης, τόσο της αποστολής όσο και της λήψης μπορούν να προσαρτηθούν στο ίδιο σήμα, προτιμούνται ξεχωριστά σήματα. Αυτό επιτρέπει στο ελαστικό buffer πληροφοριών να καθαριστεί σε περίπτωση υπερ/υποχείλισης χωρίς να επηρεάζεται η μετάδοση TX σε εξέλιξη. Επιπλέον περιγραφές για έλεγχο επανεκκίνησης και τροφοδοσίας στον Πίνακα 2-5 και Πίνακα 2-6.

Θύρες	Περιγραφή
RXRESET	Το σύγχρονο σήμα επανεκκίνησης του συστήματος λήψης κεντράει ξανά το ελαστικό buffer παραλήπτη, επανεκκινεί τον αποκωδικοποιητή 8B/10B, τον εντοπισμό κόμματος, τη δημιουργία δεσμού καναλιού, τη λογική διόρθωσης ρολογιού και άλλους καταχωρητές του παραλήπτη. Το PLL παραμένει αμετάβλητο.
TXRESET	Το σύγχρονο σήμα επανεκκίνησης του συστήματος αποστολής κεντράει ξανά τη FIFO αποστολής κι επανεκκινεί τον κωδικοποιητή 8B/10B και άλλους καταχωρητές του αποστολέα. Το PLL παραμένει αμετάβλητο.
POWERDOWN	Τερματίζει τη λειτουργία του πομποδέκτη (και στις δύο πλευρές RX και TX). Στην κατάσταση λειτουργίας POWERDOWN, οι ακροδέκτες εξόδου αποστολής TXP/TXN δεν οδηγούνται, αλλά πολώνονται από την τροφοδοσία VTTX της κατάστασης τερματισμού αποστολής. Οι ακροδέκτες εισόδου λήψης RXP/RXN αποκρίνονται παρόμοια στην τροφοδοσία VTRX της κατάστασης τερματισμού λήψης.

Πίνακας 2-5: Περιγραφή ελέγχου Επανεκκίνησης και Λειτουργίας

POWERDOWN	Κατάσταση Πομποδέκτη
0	Ο πομποδέκτης σε λειτουργία
1	Ο πομποδέκτης προσωρινά σταματημένος

Πίνακας 2-6: Περιγραφή ελέγχου Λειτουργίας

Σημειώσεις:

1. Μη-χρησιμοποιούμενοι πομποδέκτες διαμορφώνονται αυτόματα ως ανενεργοί από τα εργαλεία υλοποίησης.

Ευθυγράμμιση SERDES

### *Serializer*

Ο multi-gigabit πομποδέκτης πολλαπλασιάζει τη συχνότητα αναφοράς που παρέχεται στην είσοδο του ρολογιού αναφοράς (REFCLK) με 20, ή με 10 εάν είναι επιλεγμένη η λειτουργία μισού-ρυθμού. Τα δεδομένα μετατρέπονται από παράλληλη σε σειριακή διαμόρφωση και μεταδίδονται στις διαφορικές εξόδους TXP και TXN.

Η ηλεκτρική πολικότητα των TXP και TXN μπορεί να ελεγχθεί μέσω της θύρας TXPOLARITY της διεπαφής TX του πυρήνα Aurora. Αυτό διευκολύνει την ανάνηψη από περιπτώσεις όπου τα ίχνη της πλακέτας του τυπωμένου κυκλώματος έχουν αντιστραφεί.

### *Deserializer*

Ο πυρήνας του πομποδέκτη του RocketIO δέχεται σειριακά διαφορικά δεδομένα στις RXP και RXN εισόδους του. Το κύκλωμα ανάκτησης ρολογιού/δεδομένων εξάγει την φάση και τη συχνότητα ρολογιού από την εισερχόμενη ακολουθία δεδομένων και συγχρονίζει τα εισερχόμενα δεδομένα σε αυτό το ρολόι. Το ανακτημένο ρολόι παρουσιάζεται στην έξοδο RXRECCLK στο 1/20 του ρυθμού των εισερχόμενων σειριακών δεδομένων.

Ο παραλήπτης είναι ικανός να χειρίζεται είτε πλούσιες σε μεταβάσεις ακολουθίες 8B/10B είτε κρυπτογραφημένες ακολουθίες, και μπορεί να αντέξει μια συμβολοσειρά μέχρι 75 μη-μεταβατικά bits χωρίς σφάλμα.

Η ευθυγράμμιση λέξης εξαρτάται από την κατάσταση των bits εντοπισμού κόμματος. Αν ο εντοπισμός κόμματος είναι ενεργοποιημένος, ο πομποδέκτης αναγνωρίζει μέχρι δύο 10-bit προγραμματισμένους από πριν χαρακτήρες. Με τον εντοπισμό του χαρακτήρα ή των χαρακτήρων, το RXCOMMADET οδηγείται σε High και συγχρόνως τα δεδομένα ευθυγραμμίζονται. Εάν εντοπιστεί ένα κόμμα και τα δεδομένα είναι ευθυγραμμισμένα, δεν διαδραματίζονται περεταίρω αλλαγές ευθυγράμμισης. Εάν παραληφθεί κόμμα και η ευθυγράμμιση είναι αναγκαία, τα δεδομένα ευθυγραμμίζονται ξανά και το RXREALIGN ενεργοποιείται. Το σήμα επανευθυγράμμισης είναι μια διακριτή έξοδος. Ο πομποδέκτης συνεχώς παρακολουθεί τα δεδομένα για την παρουσία χαρακτήρων 10-bit. Με την εμφάνιση του χαρακτήρα 10-bit, τα δεδομένα ελέγχονται για ευθυγράμμιση λέξης. Εάν ο εντοπισμός κόμματος είναι απενεργοποιημένος, τα δεδομένα δεν είναι ευθυγραμμισμένα με κανένα συγκεκριμένο πρότυπο. Η επιλογή προγραμματισμού επιτρέπει στο χρήστη την ευθυγράμμιση δεδομένων σε κόμμα θετικών συμβόλων, αρνητικών συμβόλων, ή και τα δύο, ή μια μοναδικά ορισμένη από το χρήστη ακολουθία.

Η ηλεκτρική πολικότητα των RXP και RXN μπορεί να εναλλαχθεί μέσω της θύρας RXPOLARITY. Αυτό μπορεί να είναι χρήσιμο αν συμβεί τα ίχνη της πλακέτας τυπωμένου κυκλώματος να έχουν αντιστραφεί.

### Ρολόι και σηματοδότηση ρολογιού

Τα ρολόγια εισόδου ή τα ρολόγια αναφοράς ενός Multi-Gigabit πομποδέκτη (MGT) έχουν πολύ αυστηρές προδιαγραφές. Περιλαμβάνουν πολύ ακριβή προσδιορισμό συχνότητας η οποία μετράται σε parts per million (PPM) επιτρεπόμενων σφαλμάτων συχνότητας. Ορίζεται, λοιπόν, η μονάδα parts per million (0.0001%) για την οποία αναφέρεται χαρακτηριστικά ότι 12 ppm αντιστοιχούν μόλις σε ένα δευτερόλεπτο ανά μέρα. Για παράδειγμα 500 ppm θα σήμαινε ότι ένα ρολόι θα είχε απόκλιση κατά 43

δευτερόλεπτα ανά μέρα, πράγμα που θα μπορούσε να συμβεί μόνο σε παλιά κακής ποιότητας μηχανικά ρολόγια χειρός! Ακόμα διαθέτει στενά όρια για το jitter (παρέχεται σαφής ορισμός του στην παράγραφο “Θέματα Αναλογικού Σχεδιασμού”, σελ. 39), που ορίζεται με βάση μονάδες χρόνου σε picoseconds ή σε μοναδιαία διαστήματα (Unit Intervals UI) που ταυτίζεται με τη διάρκεια του χρόνου ενός συμβόλου. Παραδείγματος χάριν, 0.2 UI ισούται με το 20% του χρόνου συμβόλου.

Υπάρχουν οκτώ είσοδοι ρολογιού μέσα στο στιγμιότυπο κάθε πομποδέκτη RocketIO (βλέπε Πίνακα 2-7). REFCLK και BREFCLK είναι ρολόγια αναφοράς παραγόμενα από μια εξωτερική πηγή και παρουσιάζονται στην FPGA ως διαφορικές είσοδοι. Τα ρολόγια αναφοράς συνδέονται στις θύρες REFCLK ή BREFCLK του multi-gigabit πομποδέκτη (MGT) του RocketIO. Μόνο ένα από αυτά τα ρολόγια αναφοράς χρειάζεται για να οδηγήσει τον MGT, ωστόσο το BREFCLK ή το BREFCLK2 πρέπει να χρησιμοποιηθεί για σειριακές ταχύτητες της τάξεως των 2.5 Gb/s ή παραπάνω (βλέπε Πίνακα 2-8).

Σημείωση: Το ρολόι αναφοράς πρέπει να είναι τουλάχιστον 50 MHz (για την λειτουργία πλήρους-ρυθμού μόνο; 60 MHz για λειτουργία μισού-ρυθμού) με κύκλο εργασίας ανάμεσα σε 45% και 55%, και θα πρέπει να έχει σταθερότητα συχνότητας της τάξεως των  $\pm 100$  ppm ή καλύτερη, με το θόρυβο όσο χαμηλότερο γίνεται. Το λειτουργικό τμήμα 3 του data sheet του Virtex-II Pro δίνει λεπτομέρειες.

Ρολόι	Είσοδοι/Εξοδοι	Περιγραφή
BREFCLK	Είσοδος	Το ρολόι αναφοράς που χρησιμοποιείται για την ανάγνωση της TX FIFO και πολλαπλασιάζεται με 20 για τη μετατροπή από παράλληλο-σε σειριακό (20x)
BREFCLK2	Είσοδος	Εναλλακτικό στο BREFCLK
RXRECCLK	Έξοδος	Ρολόι ανάκτησης (από τη σειριακή ροή δεδομένων) διαιρεμένο με 20. Παρέχει το χρονισμό για τα δεδομένα μέσα στο ελαστικό buffer.
REFCLK	Είσοδος	Το ρολόι αναφοράς που χρησιμοποιείται για την ανάγνωση της TX FIFO και πολλαπλασιάζεται με 20 για τη μετατροπή από παράλληλο-σε σειριακό (20x)
REFCLK2	Είσοδος	Εναλλακτικό στο REFCLK
REFCLKSEL	Είσοδος	Επιλέγει πιο ρολόι αναφοράς χρησιμοποιείται. Η τιμή 0 επιλέγει το REFCLK και η τιμή 1 επιλέγει το REFCLK2
RXUSRCLK	Είσοδος	Ρολόι από την FPGA που χρησιμοποιείται για την ανάγνωση του RX ελαστικού buffer. Χρονίζει τα σήματα CHBONDI και CHBONDO μέσα κι έξω από τον πομποδέκτη. Αυτό το ρολόι είναι τυπικά το ίδιο με το TXUSRCLK.
TXUSRCLK <sup>(1)</sup>	Είσοδος	Ρολόι από την FPGA που χρησιμοποιείται για την εγγραφή του TX buffer. Αυτό το ρολόι πρέπει να είναι σύμφωνο κατά συχνότητα στο REFCLK για την ενδεχόμενη λειτουργία.
RXUSRCLK2	Είσοδος	Ρολόι από την FPGA που χρησιμοποιείται για να χρονίζει δεδομένα και την κατάσταση του RX μεταξύ του πομποδέκτη και του υλικού της FPGA. Η σχέση μεταξύ RXUSRCLK2 και RXUSRCLK εξαρτάται από το πλάτος του datapath του παραλήπτη. Το RXUSRCLK2 είναι τυπικά το ίδιο με το TXUSRCLK2.
TXUSRCLK2 <sup>(1)</sup>	Είσοδος	Ρολόι από την FPGA που χρησιμοποιείται για να χρονίζει δεδομένα και την κατάσταση του TX μεταξύ του πομποδέκτη και του υλικού της FPGA. Η σχέση μεταξύ TXUSRCLK2 και TXUSRCLK εξαρτάται από το πλάτος του datapath του

		ΑΠΟΣΤΟΛΕΑ.
--	--	------------

Πίνακας 2-7: Θύρες Ρολογιών

Σημειώσεις:

1. TXUSRCLK και TXUSRCLK2 πρέπει να οδηγούνται από πηγές ρολογιού, ακόμα κι αν χρησιμοποιείται μόνο ο παραλήπτης του MGT.

	Ρυθμός Δεδομένων		Δρομολόγηση	
	600 Mb/s - 2.499 Gb/s	2.500 Gb/s – 3.125 Gb/s	Μπορεί να δρομολογηθεί κατά μήκος του chip ;	Μπορεί να δρομολογηθεί μέσω του BUFG ;
REFCLK	✓	x	✓ <sup>(2)</sup>	✓ <sup>(2)</sup>
BREFCLK	✓	✓	x <sup>(1)</sup>	x <sup>(1)</sup>

Πίνακας 2-8: Χρήση Ρολογιού Αναφοράς

Σημειώσεις:

1. Λόγω αποκλειστικής δρομολόγησης ώστε να μειωθεί ο θόρυβος, το BREFCLK δεν μπορεί να δρομολογηθεί μέσα από το υλικό.

2. Ενώ αυτή η επιλογή είναι διαθέσιμη στο πυρίτιο, αυτή η τοπογραφία προσθέτει επιπλέον θόρυβο στο ρολόι αναφοράς ο οποίος μπορεί να επηρεάσει τη συνολική απόδοση του πομποδέκτη.

Για την παροχή ρολογιού στα σειριακά δεδομένα, η αρχιτεκτονική PLL για το πομποδέκτη χρησιμοποιεί το ρολόι αναφοράς ως την πηγή υπερπλήρωσης. Αφαιρώντας τα ρολόγια αναφοράς σταματούν να δουλεύουν το RX και το TX PLLs. Για αυτό, πρέπει να παρέχεται πάντα ένα ρολόι αναφοράς. Αυτό είναι σημαντικό ειδικά στο τέλος της διαμόρφωσης όταν το τμήμα PMA του MGT χρειάζεται ένα ρολόι αναφοράς για να αρχικοποιηθεί κατάλληλα. Αν σε αυτό το σημείο δεν είναι διαθέσιμο ρολόι αναφοράς, ο χρήστης πρέπει να αλλάξει τον ακροδέκτη POWERDOWN όταν γίνει διαθέσιμο το ρολόι για να διασφαλίσει ότι το PMA είναι αρχικοποιημένο κατάλληλα.

Το ρολόι αναφοράς επίσης τροφοδοτεί ένα Digital Clock Manager (DCM) ή ένα BUFG για να δημιουργήσει όλα τα άλλα ρολόγια για τον MGT. Πότε δεν πρέπει να δημιουργείται ρολόι αναφοράς από ένα DCM, λόγω της εισαγωγής ανεπιθύμητου θορύβου. Όποιος επιπρόσθετος θόρυβος στο ρολόι αναφοράς θα μεταφερθεί στη σειριακή Είσοδο/Εξόδο RX και TX του πομποδέκτη.

Είναι προτεινόμενο όλα τα ρολόγια αναφοράς μέσα στην FPGA να είναι LVDS ή LVPECL IBUFGDS. Το χαρακτηριστικό DCI ή το DT του LVDS είναι προαιρετικά. Γίνεται αναφορά στο Εγχειρίδιο χρήσης του Virtex-II Pro Platform FPGA (“Chapter 3, “Design Considerations”) για μια ολοκληρωμένη παράθεση και αναφορά στο IBUFGDS και άλλων διαθέσιμων θεμελιωδών δομών (primitives) Εισόδου/Εξόδου. Ακόμα δείτε την ενότητα “Reference Clock” στο “Chapter 3” αυτού του οδηγού.

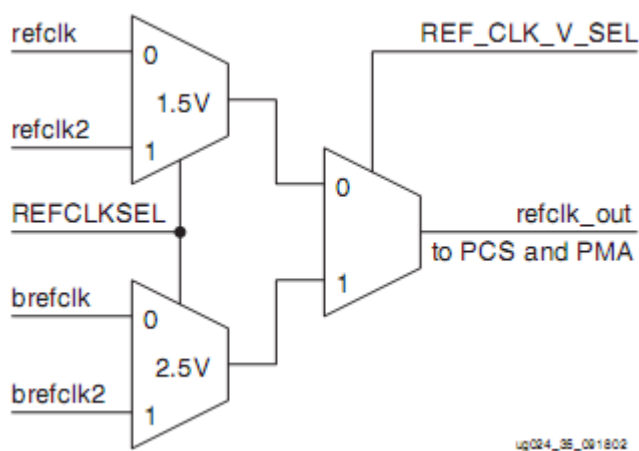
Τυπικά, TXUSRCLK = RXUSRCLK και TXUSRCLK2 = RXUSRCLK2. Ο πομποδέκτης χρησιμοποιεί ένα ή δύο ρολόγια που έχουν δημιουργηθεί από το DCM. Παραδείγματος χάριν, τα ρολόγια USRCLK και USRCLK2 τρέχουν στην ίδια ταχύτητα αν χρησιμοποιείται το μονοπάτι δεδομένων πλάτους 2-byte. Το USRCLK πρέπει πάντα να είναι κλειδωμένο ως προς τη συχνότητα στο ρολόι αναφοράς του πομποδέκτη RocketIO όταν SERDES\_10B = FALSE (λειτουργία πλήρους-ρυθμού full-rate operation).

BREFCLK

Όπως αναφέρθηκε σε προηγούμενη παράγραφο, σε ταχύτητες της τάξεως των 2.5 Gb/s ή μεγαλύτερες, η REFCLK διαμόρφωση εισάγει περισσότερο από το μέγιστο αποδεκτό θόρυβο στο πομποδέκτη του RocketIO. Για αυτές τις υψηλότερες ταχύτητες, χρειάζεται η BREFCLK διαμόρφωση. Η διαμόρφωση BREFCLK χρησιμοποιεί πόρους αποκλειστικής δρομολόγησης που ελαττώνουν το θόρυβο.

Το BREFCLK πρέπει να εισαχθεί στην FPGA μέσω αποκλειστικής Εισόδου/Εξόδου ρολογιού. Το BREFCLK μπορεί να συνδεθεί στις εισόδους BREFCLK του πομποδέκτη και στην είσοδο CLKIN του DCM για τη δημιουργία των USRCLK. Αν είναι επιθυμητό να χρησιμοποιηθούν όλοι οι πομποδέκτες σε μια Virtex-II Pro FPGA, πρέπει να δημιουργηθούν δύο BREFCLK, ένα για το πάνω μέρος του chip και ένα για το κάτω μέρος. Αυτά τα αποκλειστικά ρολόγια χρησιμοποιούν τις ίδιες εισόδους ρολογιού για κάθε συσκευασία:

Ένα χαρακτηριστικό (REF\_CLK\_V\_SEL) και μια θύρα (REFCLKSEL) προσδιορίζουν ποιο ρολόι αναφοράς χρησιμοποιείται για το τμήμα PMA του MGT. Σχήμα 2-7 δείχνει πως το REFCLK και το BREFCLK επιλέγονται με χρήση του REFCLKSEL και REF\_CLK\_V\_SEL.



Σχήμα 2-7 : Λογική Επιλογής REFCLK/BREFCLK

### Λόγος ρολογιού

Το USRCLK2 δίνει ρολόι στα data buffers. Η ικανότητα αποστολής/λήψης παράλληλων δεδομένων προς/από το πομποδέκτη σε τρία διαφορετικά πλάτη προϋποθέτει ο χρήστης να αλλάξει τη συχνότητα του USRCLK2. Αυτό δημιουργεί ένα λόγο συχνότητας μεταξύ του USRCLK και του USRCLK2. Οι καθοδικές ακμές των ρολογιών πρέπει να ευθυγραμμίζονται. Ο Πίνακας 2-9 δείχνει τους λόγους για κάθε ένα από τα τρία πλάτη δεδομένων.

Πλάτος Δεδομένων	Λόγος Συχνότητας του USRCLK/USRCLK2
1 byte	1:2 <sup>(1)</sup>
2 byte	1:1
4 byte	2:1 <sup>(1)</sup>

Πίνακας 2-9: Λόγοι Ρολογιών για κάθε Πλάτος Δεδομένων

Σημειώσεις:

1. Κάθε ακμή του πιο αργού ρολογιού πρέπει να ευθυγραμμίζεται με την καθοδική ακμή του ταχύτερου ρολογιού

### Ανάκτηση Ρολογιού και Δεδομένων

Ψηφιακές σειριακές ροές δεδομένων υψηλών-ταχυτήτων αποστέλλονται χωρίς ένα συνοδευτικό ρολόι. Ο παραλήπτης γεννά ένα ρολόι από προσεγγιστική αναφορά πηγής, και μετά ευθυγραμμίζεται κατά φάση στις μεταβάσεις της ροής δεδομένων με ένα phase-locked loop (PLL). Αυτή η διαδικασία είναι γνωστή ως **ανάκτηση ρολογιού και δεδομένων (CDR)**.

Στον πομποδέκτη RocketIO η σύγχρονη λήψη σειριακών δεδομένων διευκολύνεται από ένα κύκλωμα ανάκτησης ρολογιού και δεδομένων. Αυτό το κύκλωμα χρησιμοποιεί ένα πλήρως λιθογραφικό Phase-Locked Loop (PLL), το οποίο δεν χρειάζεται κανένα εξωτερικό μέλος. Το κύκλωμα ανάκτησης ρολογιού/δεδομένων εξάγει και τα δύο, τη φάση και τη συχνότητα από την εισερχόμενη ακολουθία δεδομένων. Το ανακτημένο ρολόι παρουσιάζεται στην έξοδο RXRECCLK στο 1/20 του ρυθμού των ληφθέντων σειριακών δεδομένων. Η ροή δεδομένων πρέπει να περιέχει έναν επαρκή αριθμό μεταβάσεων για την ομαλή λειτουργία της CDR. Το όριο, μέχρι το οποίο, μια μονάδα ανάκτησης ρολογιού μπορεί να λειτουργήσει χωρίς μετάβαση είναι γνωστό ως προδιαγραφή μεγίστου πλήθους διαδοχικών ίδιων ψηφίων (CID). Ο πομποδέκτης πολλών-gigabit έχει ανοχή 75 χαμένων μεταβάσεων πριν την απώλεια κλειδώματος στην εισερχόμενη ροή δεδομένων. Ενώ, κατά την έναρξη λειτουργίας του, χρειάζεται περίπου 5000 μεταβάσεις ώστε να εγγυηθεί κλείδωμα στο ρυθμό εισερχόμενων δεδομένων. Για να δουλέψει, λοιπόν, αυτή η τεχνική, μια ροή δεδομένων πρέπει να μεταβαίνει αρκετά συχνά ώστε να διορθώνεται όποια απόκλιση στον ταλαντωτή του PLL. Έτσι ώστε να εξασφαλίζονται συχνές μεταβάσεις, χρησιμοποιείται κάποιας μορφής κωδικοποίηση, εδώ η κωδικοποίηση 8B/10B.

### Σύνθεση Ρολογιού

Ο πομπός gigabit πολλαπλασιάζει τη συχνότητα αναφοράς που παρέχεται στην είσοδο του ρολογιού αναφοράς (REFCLK) με 20.

Δεν θεωρείται καμιά σχέση σταθερής φάσης μεταξύ REFCLK, RXRECCLK, και/ή όποιου άλλου ρολογιού που δεν συνδέεται με κάποιο από αυτά τα ρολόγια. Όταν τα 4-byte ή 1-byte μονοπάτια δεδομένων λήψης χρησιμοποιούνται, τα RXUSRCLK και RXUSRCLK2 έχουν διαφορετικές συχνότητες (1:2), και κάθε ακμή του πιο αργού ρολογιού είναι ευθυγραμμισμένη σε μια καθοδική ακμή του ταχύτερου ρολογιού. Η ίδια σχέση ισχύει στα TXUSRCLK και TXUSRCLK2. Βλέπε Πίνακα 2-9, σελίδα 32, για λεπτομέρειες.

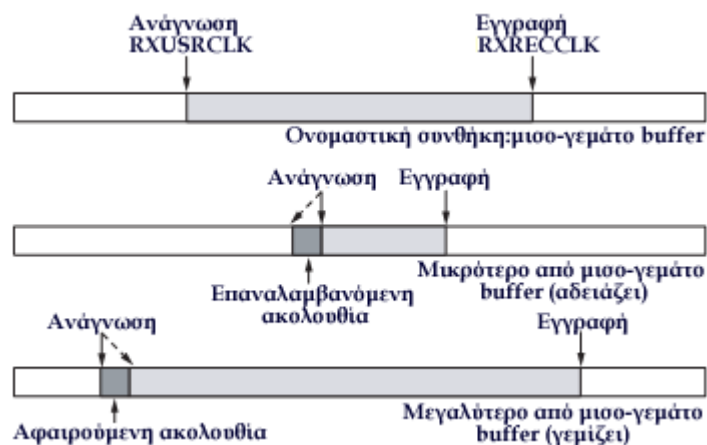
### Ανάκτηση ρολογιού και δεδομένων

Τα κυκλώματα ανάκτησης ρολογιού/δεδομένων (CDR) κλειδώνουν αυτόματα στο ρολόι αναφοράς αν δεν υπάρχουν δεδομένα. Για την ενδεδειγμένη λειτουργία, το σήμα ανάκτησης ρολογιού αποστολής TXUSRCLK πρέπει να έχει την ίδια ακριβώς συχνότητα με τα σήματα REFCLK ή BREFCLK. Τα σήματα REFCLK ή BREFCLK, και το σήμα ανάκτησης ρολογιού λήψης RXUSRCLK, καθώς και η εισερχόμενη ροή (RXRECCLK) δεν πρέπει να ξεπερνούν στη διακύμανση συχνότητας τα  $\pm 100$  ppm.

Είναι, τέλος, κρίσιμο να διατηρηθεί ο θόρυβος από την τροφοδοσία χαμηλός ώστε να ελαχιστοποιηθούν οι καταστάσεις λειτουργίας κοινού και διαφορικού θορύβου μέσα στο κύκλωμα ανάκτησης ρολογιού/δεδομένων.

### Διόρθωση Ρολογιού

Το ρολόι RXRECCLK (το ανακτημένο ρολόι) αντικατοπτρίζει το ρυθμό δεδομένων των εισερχόμενων δεδομένων. Το ρολόι RXUSRCLK προσδιορίζει το ρυθμό στον οποίο ο πυρήνας της FPGA καταναλώνει τα δεδομένα. Ιδανικά, αυτοί οι ρυθμοί είναι πανομοιότυποι. Παρ' όλ' αυτά, λόγω ότι τα ρολόγια τυπικά έχουν διαφορετικές πηγές, ένα από τα ρολόγια είναι ταχύτερο από το άλλο. Το buffer πληροφοριών του παραλήπτη φανερώνει αυτή τη διαφορά μεταξύ των ρυθμών ρολογιού. Βλέπε Σχήμα 2-8.



Σχήμα 2-8 : Διόρθωση Ρολογιού στο Δέκτη

Ονομαστικά, το buffer (απομονωτής) πληροφοριών είναι πάντα μισό-γεμάτο. Αυτό φαίνεται στο buffer πληροφοριών στην κορυφή, όπου η σκιασμένη περιοχή συμβολίζει αποθηκευμένα προσωρινά δεδομένα που δεν έχουν αναγνωστεί ακόμα. Τα ληφθέντα δεδομένα εισάγονται μέσω του δείκτη εγγραφής υπό τον έλεγχο του RXRECCLK. Ο πυρήνας της FPGA κάνει ανάγνωση δεδομένων μέσω του δείκτη ανάγνωσης υπό τον έλεγχο του RXUSRCLK. Η μισό-γεμάτο/μισό-άδειο συνθήκη του buffer παρέχει προστασία για τους ρυθμούς ρολογιού που διαφέρουν. Αυτή η λειτουργία συνεχίζει επ' αόριστον, άσχετα αν λαμβάνονται δεδομένα “με νόημα”. Όταν δεν υπάρχουν δεδομένα με νόημα για λήψη, τα εισερχόμενα δεδομένα αποτελούνται από χαρακτήρες IDLE ή αλλά “γεμίσματα” (padding).

Εάν το RXUSRCLK είναι ταχύτερο από το RXRECCLK, το buffer γίνεται πιο άδειο με το χρόνο. Η λογική διόρθωσης ρολογιού διορθώνει για αυτό ελαττώνοντας το δείκτη ανάγνωσης ώστε να ξαναδιαβάσει μια επαναλαμβανόμενη ακολουθία byte. Αυτό φαίνεται στο μεσαίο buffer, Σχήμα 2-8, όπου ο δείκτης ανάγνωσης με τη συνεχόμενη γραμμή μειώνεται στη τιμή που συμβολίζει ο δείκτης με τη διακεκομμένη γραμμή. Ελαττώνοντας τον δείκτη ανάγνωσης αντί να αυξάνεται με το σύνηθες τρόπο, το buffer είναι μερικώς γεμάτο. Ο πομποδέκτης εισάγει μια απλή επαναλαμβανόμενη ακολουθία byte όποτε χρειάζεται ώστε να ξαναγεμίσει ένα buffer. Αν το μήκος της ακολουθίας byte είναι μεγαλύτερο από ένα, και αν το χαρακτηριστικό CLK\_COR\_REPEAT\_WAIT είναι 0, τότε ο πομποδέκτης μπορεί να επαναλάβει την ίδια ακολουθία πολλαπλές φορές μέχρι να ξαναγεμιστεί το buffer στην κατάσταση μισό-γεμάτο.



Ομοίως, αν το RXUSRCLK είναι πιο αργό από το RXRECCLK, το buffer γεμίζει με το χρόνο. Η λογική διόρθωσης ρολογιού το διορθώνει αυτό αυξάνοντας το δείκτη ανάγνωσης ώστε να περάσει πάνω από μια κινούμενη ακολουθία byte που δε χρειάζεται να εμφανιστεί στην τελική ροή του πυρήνα στην FPGA. Αυτό φαίνεται στο buffer στο κάτω μέρος, Σχήμα 2-8, όπου ο δείκτης ανάγνωσης με τη συνεχόμενη γραμμή αυξάνεται μέχρι την τιμή που συμβολίζεται από το δείκτη διακεκομμένης γραμμής. Αυτό επιταχύνει το άδειασμα του buffer, εμποδίζοντας την υπερχείλισή του. Η σχεδίαση του πομποδέκτη προσπερνάει μια μεμονωμένη ακολουθία byte, όταν χρειάζεται, για να αδειάσει μερικώς ένα buffer. Εάν το χαρακτηριστικό CLK\_COR\_REPEAT\_WAIT είναι 0, ο πομποδέκτης μπορεί επίσης να προσπεράσει τέσσερις διαδοχικές αφαιρούμενες ακολουθίες byte σε ένα βήμα, ώστε να αδειάσει περεταίρω το buffer όταν είναι αναγκαίο.

Αυτές οι λειτουργίες προϋποθέτουν η λογική διόρθωσης ρολογιού να αναγνωρίσει μια ακολουθία byte που μπορεί να επαναλαμβάνεται ή να παραλείπεται ελεύθερα στην εισερχόμενη ροή δεδομένων. Αυτή η ακολουθία είναι γενικά μια ακολουθία IDLE, ή άλλη ακολουθία που απαρτίζεται από ειδικές τιμές που υπάρχουν στα κενά που διαχωρίζουν τα πακέτα με χρήσιμα δεδομένα. Αυτά τα κενά χρειάζεται να διαδραματίζονται αρκετά συχνά ώστε να διευκολύνουν την έγκαιρη εκτέλεση της διόρθωσης ρολογιού.

Η λογική διόρθωσης ρολογιού έχει την ικανότητα να απομακρύνει μέχρι τέσσερις ακολουθίες IDLE κατά τη διάρκεια μιας διόρθωσης ρολογιού. Πόσα IDLE απομακρύνονται εξαρτάται από αρκετούς παράγοντες, συμπεριλαμβανομένου πόσα IDLEs έχουν παραληφθεί και αν το CLK\_COR\_KEEP\_IDLE είναι TRUE ή FALSE. Για παράδειγμα, αν λαμβάνονται τρία IDLE και το CLK\_COR\_KEEP\_IDLE είναι στο TRUE, τουλάχιστον μια ακολουθία IDLE πρέπει να παραμείνει αφού ολοκληρωθεί η διόρθωση ρολογιού. Αυτό περιορίζει τη λογική διόρθωσης ρολογιού να απομακρύνει μόνο δύο από τις τρεις ακολουθίες IDLE. Αν το CLK\_COR\_KEEP\_IDLE είναι FALSE, τότε όλα τα τρία IDLE μπορούν να απομακρυνθούν. Ο Πίνακας 2-10 απεικονίζει τη σχέση μεταξύ του αριθμού των ακολουθιών IDLE που απομακρύνονται, την κληρονομική σταθερότητα του REFCLK, και τον αριθμό bytes που επιτρέπονται μεταξύ των ακολουθιών διόρθωσης ρολογιού.

	Επιτρεπόμενα Bytes Μεταξύ των Ακολουθιών Διόρθωσης Ρολογιού <sup>(1)</sup>			
Σταθερότητα REFCLK	Αφαιρείται 1 Ακολουθία IDLE <sup>(2)</sup>	Αφαιρούνται 2 Ακολουθίες IDLE	Αφαιρούνται 3 Ακολουθίες IDLE	Αφαιρούνται 4 Ακολουθίες IDLE
100 ppm <sup>(3)</sup>	5,000	10,000	15,000	20,000
50 ppm	10,000	20,000	30,000	40,000
20 ppm	25,000	50,000	75,000	100,000

Πίνακας 2-10 : Bytes Δεδομένων που Επιτρέπονται μεταξύ των Διορθώσεων Ρολογιού ως Συνάρτηση της σταθερότητας του REFCLK και των Ακολουθιών IDLE που αφαιρούνται.

Σημειώσεις:

1. Όλα τα νούμερα είναι προσεγγιστικά.
2. IDLE = η προσδιορισμένη ακολουθία διόρθωσης ρολογιού.
3. PPM = parts per million (0.0001%).

Επισκόπηση Διεπαφής Υλικού με Buffers Αποστολέα και Ελαστικά Buffers(Παραλήπτη)

Και οι δύο, ο αποστολέας και ο παραλήπτης περιλαμβάνουν buffers (FIFOs) στο μονοπάτι δεδομένων. Ο σκοπός του buffer αποστολέα είναι η αποθήκευση της διαφοράς

φάσης μεταξύ των σημάτων TXUSRCLK και REFCLK. Απαραίτητη συνθήκη για αυτό είναι το ρολόι της FPGA TXUSRCLK, που χρησιμεύει ως δείκτης εγγραφής του buffer αποστολέα, να είναι κλειδωμένο-σε-συχνότητα με το ρολόι αναφοράς REFCLK, που χρησιμεύει ως δείκτης ανάγνωσης του buffer αποστολέα. Αυτό συνεπάγεται ότι δεν χρειάζεται διόρθωση ρολογιού και δεσμός καναλιού για το buffer αυτό. Επιτρέπονται διακυμάνσεις φάσης μέχρι ένα κύκλο ρολογιού. Μια απλή FIFO είναι αρκετή για την υλοποίηση του buffer αποστολέα. Ο παραλήπτης χρησιμοποιεί ένα “ελαστικό” buffer, όπου ο όρος “ελαστικό” αναφέρεται στην ικανότητα αλλαγής του δείκτη ανάγνωσης για τη διόρθωση ρολογιού και το δεσμό καναλιού. Το buffer παραλήπτη (ελαστικό buffer) επιτελεί δύο στόχους. Πρώτον, αποθήκευση της μικρής διαφοράς στη συχνότητα ανάμεσα στο ανακτημένο ρολόι RXRECCLK και το εσωτερικό ρολόι του πυρήνα της FPGA RXUSRCLK για διόρθωση ρολογιού. Δεύτερον, να επιτρέψει την επανευθυγράμμιση της ροής εισόδου ώστε να διασφαλίσει την κατάλληλη ευθυγράμμιση των δεδομένων που διαβάζονται μέσω πολλαπλών πομποδεκτών, δηλαδή του δεσμού καναλιού.

#### Δεσμός Καναλιού ή αλλιώς Ευθυγράμμιση Καναλιού. Γενικά

Ο δεσμός καναλιού είναι η τεχνική δεσίματος πολλών σειριακών καναλιών ή γραμμών μαζί ώστε να δημιουργηθεί ένα αθροιστικό κανάλι. Πολλά κανάλια/γραμμές τροφοδοτούνται στην πλευρά αποστολής από ένα παράλληλο δίαυλο και αναπαράγονται στην πλευρά λήψης ως ένας πανομοιότυπος παράλληλος δίαυλος. Ο μέγιστος αριθμός σειριακών διαφορικών ζευγών που μπορεί να κάνουν δεσμό είναι 24.

Ο δεσμός καναλιού επιτρέπει σε αυτές τις θεμελιώδεις δομές (primitives) που τον υποστηρίζουν να αποστέλλουν δεδομένα πάνω σε πολλαπλές γραμμές. Ανάμεσα σε αυτές τις θεμελιώδεις δομές (primitives) είναι το GT\_CUSTOM, GT\_INFINIBAND, GT\_XAUI, και GT\_AURORA. Για το “δεσμό” μεταξύ γραμμών, πάντα θα υπάρχει ένας “Master”. Οι άλλες γραμμές μπορεί είτε να είναι SLAVE\_1\_HOP ή SLAVE\_2\_HOPS. Το SLAVE\_1\_HOP είναι Slave σε έναν Master που μπορεί επίσης να είναι αλυσιδωτά συνδεδεμένος σε ένα SLAVE\_2\_HOPS. Ένας SLAVE\_2\_HOPS μπορεί να είναι Slave μόνο σε έναν SLAVE\_1\_HOP και το CHBONDO αυτού να μην συνδέεται σε άλλο πομποδέκτη. Για να οριστεί ένας πομποδέκτης ως Master ή Slave, το χαρακτηριστικό CHAN\_BOND\_MODE πρέπει να οριστεί σε μια από τις τρεις αναθέσεις: Master, SLAVE\_1\_HOP, ή SLAVE\_2\_HOPS. Για να απενεργοποιηθεί ο δεσμός καναλιού, ορίζεται το χαρακτηριστικό του πομποδέκτη σε “OFF”. Οι πιθανές τιμές που μπορεί να χρησιμοποιηθούν φαίνονται στον Πίνακα 2-11.

Λειτουργία	CHBONDI	CHBONDO
OFF	NA	AN
MASTER	NA	Slave 1 CHBONDI
SLAVE_1_HOP	Master CHBONDO	Slave 2 CHBONDI
SLAVE_2_HOPS	Slave 1 CHBONDO	NA

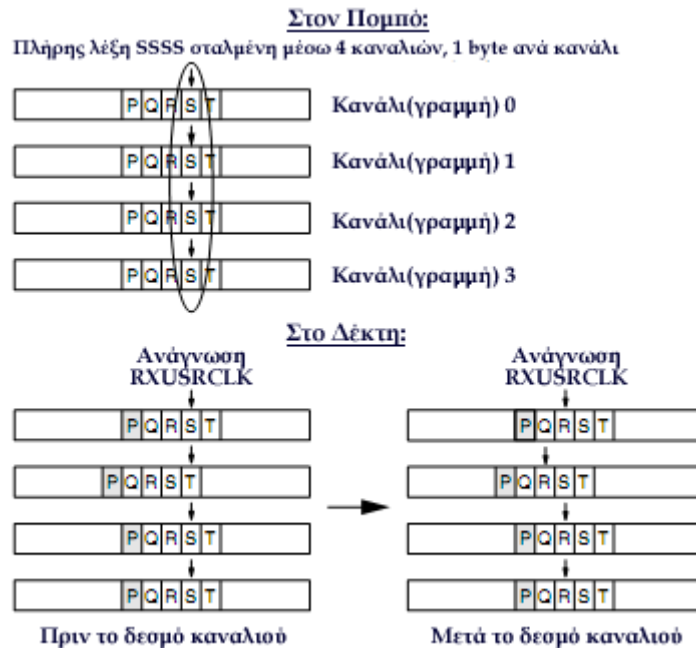
Πίνακας 2-11: Συνδέσεις Δεσμού Καναλιού

Σημείωση: Όλα τα πρότυπα που χρησιμοποιούν και τα δύο, τη διόρθωση ρολογιού και το δεσμό καναλιού χρειάζονται ένα κενό μεγαλύτερο από ή ίσο με 4 bytes μεταξύ των ακολουθιών διόρθωσης ρολογιού και δεσμού καναλιού. Εάν ένας χρήστης δημιουργήσει το δικό του/της πρωτόκολλο που χρησιμοποιεί διόρθωση ρολογιού και δεσμό καναλιού, τότε πρέπει να εξασφαλίσει ότι υπάρχει τουλάχιστον κενό 4 byte μεταξύ των ακολουθιών.

Η ακολουθία δεσμού καναλιού είναι παρόμοια σε διαμόρφωση με την ακολουθία διόρθωσης ρολογιού. Αυτή η ακολουθία έχει οριστεί κατάλληλα για τις θεμελιώδεις δομές (primitives) που υποστηρίζουν δεσμό καναλιού. Ο GT\_CUSTOM είναι η μόνη θεμελιώδης δομή που επιτρέπει τροποποίηση στην ακολουθία. Αυτές οι ακολουθίες απαρτίζονται από μία ή δύο ακολουθίες μήκους μέχρι 4 bytes η καθεμία, όπως ορίζεται από το CHAN\_BOND\_SEQ\_LEN και το CHAN\_BOND\_SEQ\_2\_USE. Για επιπλέον λεπτομέρειες βλέπε Παράρτημα, παράγραφο “Σήματα Ελέγχου Δεσμού Καναλιού”.

#### Λειτουργία Δεσμού Καναλιού (Ευθυγράμμιση)

Κάποια πρότυπα Εισόδου/Εξόδου gigabit όπως το XAUI προσδιορίζουν τη χρήση πολλαπλών μεταδοτών εν παραλλήλω για υψηλότερους ρυθμούς δεδομένων. Λέξεις δεδομένων χωρίζονται σε bytes, με αποστολή κάθε byte από ξεχωριστό κανάλι, ή καλύτερα γραμμή, πομποδέκτη. Βλέπε Σχήμα 2-9.



Σχήμα 2-9 : Δημιουργία Δεσμού Καναλιού (Ευθυγράμμιση)

Το πάνω μισό του διαγράμματος δείχνει τη μετάδοση λέξεων διαχωρισμένων ανάμεσα σε τέσσερις πομποδέκτες (κανάλια ή γραμμές). PPPP, QQQQ, RRRR, SSSS, και TTTT συμβολίζουν λέξεις που αποστέλλονται πάνω στις τέσσερις γραμμές.

Το κάτω-αριστερά τμήμα του διαγράμματος δείχνει την αρχική κατάσταση στους παραλήπτες της FPGA στην άλλη άκρη των τεσσάρων γραμμών. Λόγω ποικιλίας στην καθυστέρηση μετάδοσης — ειδικά αν οι γραμμές είναι δρομολογημένες διαμέσου επαναληπτών — ο πυρήνας της FPGA μπορεί να μην συναρμολογήσει τα bytes σε ολοκληρωμένες λέξεις. Η κάτω-αριστερά απεικόνιση δείχνει την λάθος συναρμολόγηση των λέξεων δεδομένων PQPP, QRQQ, RSRR, κτλ.

Για την υποστήριξη διόρθωσης αυτής της λάθος ευθυγράμμισης, η ροή δεδομένων περιλαμβάνει ειδικές ακολουθίες byte που προσδιορίζουν αντίστοιχα σημεία σε πολλές γραμμές. Στο κάτω-αριστερά μισό του Σχήματος 2-9, τα σκιασμένα “P” bytes

συμβολίζουν αυτούς τους ειδικούς χαρακτήρες. Κάθε παραλήπτης αναγνωρίζει το χαρακτήρα δεσμού καναλιού “P”, και θυμάται την θέση του στο buffer. Σε κάποιο σημείο, ένας πομποδέκτης που έχει τεθεί Master δίνει εντολή σε όλους τους πομποδέκτες να ευθυγραμμιστούν στο χαρακτήρα δεσμού καναλιού “P” (ή σε μια θέση σχετική με το χαρακτήρα δεσμού καναλιού). Μετά από αυτή τη λειτουργία, οι λέξεις που έχουν μεταδοθεί στον πυρήνα της FPGA είναι ευθυγραμμισμένες κατάλληλα: RRRR, SSSS, TTTT, κτλ., όπως φαίνεται στο κάτω-δεξιά τμήμα του Σχήματος 2-9. Για να διασφαλιστεί ότι οι γραμμές μένουν κατάλληλα ευθυγραμμισμένες ακολουθώντας τη λειτουργία δεσμού καναλιού, ο Master πομποδέκτης πρέπει επίσης να ελέγχει τις λειτουργίες διόρθωσης ρολογιού που περιγράφηκαν στην προηγούμενη ενότητα για όλους τους πομποδέκτες με δεσμό καναλιού.

### Γενικά για CRC (Cyclic Redundancy Check)

Ο Cyclic Redundancy Check (CRC) είναι μια διαδικασία εντοπισμού σφαλμάτων στα ληφθέντα δεδομένα. Η λογική CRC του πομποδέκτη του RocketIO υποστηρίζει 32-bit αμετάβλητο CRC υπολογισμό που χρησιμοποιείται από Infiniband, Fibre Channel, και Gigabit Ethernet.

### Λειτουργία CRC

Από την πλευρά του πομποδέκτη, η λογική του CRC αναγνωρίζει που πρέπει να εισαχθούν τα CRC bytes και αντικαθιστά τέσσερα αρμόδια bytes στην ουρά ενός πακέτου δεδομένων με το υπολογισμένο CRC. Για Gigabit Ethernet και Fibre Channel, ο CRC του πομποδέκτη μπορεί να αλλάξει μερικά bytes μονοπατιού για να δημιουργήσει την απαιτούμενη τρέχουσα αντιστοιχία στο τέλος του πακέτου. Από την πλευρά του παραλήπτη, η λογική CRC επαληθεύει τη ληφθείσα τιμή CRC, υποστηρίζοντας τα ίδια πρότυπα όπως παραπάνω. Με την επανεκκίνηση, η λογική CRC ξεκινάει με αρχική τιμή όλο άσσους.

### Δημιουργία CRC

Οι πομποδέκτες του RocketIO υποστηρίζουν 32-bit αμετάβλητο CRC (σταθερό πολυώνυμο 32-bit που φαίνεται παρακάτω) για Gigabit Ethernet, Fibre Channel, Infiniband, και καταστάσεις λειτουργίας προσδιορισμένες από το χρήστη.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

Σχήμα 2-10 : Σταθερό Πολυώνυμο 32-bit για CRC

Ο CRC αναγνωρίζει την Αρχή του Πακέτου (SOP), Τέλος του Πακέτου (EOP), και άλλα γνωρίσματα πακέτων για την αναγνώριση της αρχής και του τέλους των δεδομένων. Τα SOP και EOP προσδιορίζονται από CRC\_FORMAT για ETHERNET, INFINIBAND, και FIBRE\_CHAN, και σε αυτές τις περιπτώσεις ο χρήστης δε χρειάζεται να ορίσει το CRC\_START\_OF\_PKT και CRC\_END\_OF\_PKT. Όπου το

CRC\_FORMAT είναι USER\_MODE (προσδιορισμένο από το χρήστη), το CRC\_START\_OF\_PKT και το CRC\_END\_OF\_PKT χρησιμοποιούνται για να προσδιοριστεί το SOP και το EOP.



Σχήμα 2-11 : Μορφοποίηση πακέτων CRC

Ο πομποδέκτης υπολογίζει 4-byte CRC στα δεδομένα των πακέτων μεταξύ του SOP και του EOP (εξαιρώντας τα αρμόδια bytes του CRC). Ο πομποδέκτης εισάγει τον υπολογισμένο CRC ακριβώς πριν το EOP. Ο πομποδέκτης τροποποιεί Idles για ανεύρεση μονοπατιού ή EOP αν χρειάζεται ώστε να δημιουργήσει σωστή τρέχουσα αντιστοιχία για Gigabit Ethernet και Fibre Channel. Ο παραλήπτης επαναυπολογίζει τον CRC και το επαληθεύει απέναντι στον εισαχθέντα CRC. Το Σχήμα 2-11 δείχνει τη διαμόρφωση του πακέτου για τη δημιουργία του CRC. Τα άδεια κουτάκια χρησιμοποιούνται μόνο σε συγκεκριμένα πρωτόκολλα (Ethernet). Η λογική χρήστη πρέπει να δημιουργήσει ένα τεσσάρων-byte αντιπρόσωπο για τον CRC τοποθετώντας το στο TXDATA. Αλλιώς τα δεδομένα πανωγράφονται.

#### Λανθάνουσα Κατάσταση του CRC

Ενεργοποιώντας τον CRC αυξάνεται ο λανθάνων χρόνος της μετάδοσης από TXDATA στο TXP και TXN. Η ενεργοποίηση του CRC δεν επηρεάζει τον λανθάνων χρόνο από RXP και RXN στο RXDATA. Οι τυπικοί και μέγιστοι λανθάνοντες χρόνοι, εκφρασμένοι σε κύκλους του TXUSRCLK/RXUSRCLK, φαίνονται στον Πίνακα 2-12. Για διαγράμματα χρόνου που εκφράζουν αυτές τις σχέσεις, βλέπε το “Module 3” του “Virtex-II Pro Data Sheet”.

	TXDATA σε TXP και TXN σε κύκλους TXUSRCLK		RXDATA σε RXP και RXN σε κύκλους RXUSRCLK	
	Τυπικό	Μέγιστο	Τυπικό	Μέγιστο
CRC Απενεργοποιημένο	8	11	25	42
CRC Ενεργοποιημένο	14	17	25	42

Πίνακας 2-12 : Επιδράσεις του CRC στον Λανθάνοντα Χρόνο του Πομποδέκτη (transceiver)

#### Περιορισμοί Υποστήριξης του CRC στο RocketIO

Υπάρχουν περιορισμοί στην υποστήριξη του CRC που παρέχεται από τον πυρήνα του πομποδέκτη του RocketIO:

- Η υποστήριξη CRC στο RocketIO είναι υλοποιήσιμη μόνο για χρήση απλού καναλιού. Ο υπολογισμός και το byte-striping του CRC σε κανάλια πολλαπλών δεσμών δεν υποστηρίζεται. Για αυτή τη χρήση, η λογική CRC μπορεί να υλοποιηθεί στο υλικό της FPGA.
- Ο πομποδέκτης RocketIO δεν υπολογίζει την 16-bit μεταβλητή που χρησιμοποιείται από το CRC για Infiniband, και συνεπώς δεν πληροί τις

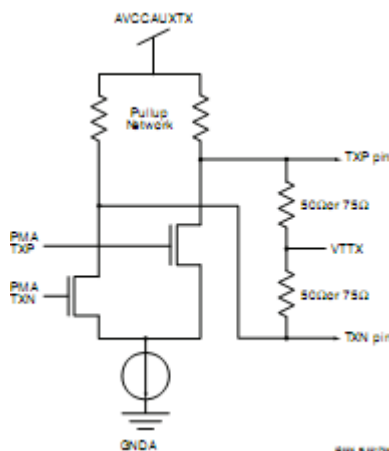
προϋποθέσεις CRC για Infiniband. Το Infiniband CRC μπορεί να υπολογιστεί στο υλικό της FPGA.

- Όλες οι διαμορφώσεις CRC έχουν ελάχιστο επιτρεπτό μέγεθος πακέτου. Αυτά τα όρια είναι μεγαλύτερα από αυτά που τίθενται από την κατάσταση λειτουργίας χρήστη, και ορίζονται από το συγκεκριμένο πρωτόκολλο.

## 2.2.4 Θέματα Αναλογικού Σχεδιασμού (Analog Design Considerations)

### Περιγραφή σειριακής Εισόδου/Εξόδου

Ο πομποδέκτης του RocketIO μεταδίδει και λαμβάνει σειριακά διαφορικά σήματα. Αυτό το γνώρισμα λειτουργεί με ελάχιστη τάση τροφοδοσίας 2.5 V DC. Ένα σειριακό διαφορικό ζεύγος αποτελείται από ένα αληθές (VP) και ένα συμπληρωματικό (VN) σύνολο σημάτων. Η διαφορά τάσης αντιπροσωπεύει τα μεταφερθέντα δεδομένα. Οπότε:  $V_P - V_N = V_{DATA}$ . Η διαφορική προσαρμογή εκτελείται με τη διασταύρωση των δυο συμπληρωματικών σημάτων. Γι' αυτό, δεν χρειάζεται ξεχωριστό επίπεδο αναφοράς. Μια γραφική αναπαράσταση αυτής της ιδέας φαίνεται στο Σχήμα 2-12.



Σχήμα 2-12 : Διαφορικός Ενισχυτής

Ο πομποδέκτης RocketIO υλοποιείται σε Current Mode Logic (CML). Μια έξοδος CML αποτελείται από τρανζίστορ που διαμορφώνονται όπως φαίνεται στο Σχήμα 2-12. Η CML χρησιμοποιεί μια θετική τροφοδοσία και προσφέρει εύκολες προδιαγραφές διεπαφής. Σε αυτή την διαμόρφωση, και τα δύο σκέλη του οδηγού, το VP και VN, απορροφούν ρεύμα, με το ένα σκέλος να απορροφά πάντα περισσότερο ρεύμα από το συμπληρωματικό του. Η έξοδος CML αποτελείται από ένα διαφορικό ζεύγος των 50Ω (ή, προαιρετικά, 75Ω) αντιστάτες πηγής. Η ταλάντωση του σήματος δημιουργείται διακόπτοντας το ρεύμα σε ένα διαφορικό ζεύγος κοινής-υποδοχής.

Οι προδιαγραφές του διαφορικού πομποδέκτη φαίνονται στον Πίνακα 2-13

Παράμετρος	Min	Typ	Max	Μονάδες	Συνθήκες
VOUT	Διαφορικό peak to peak σειριακής εξόδου	800	1600	mV	Η τάση διαφορικής εξόδου είναι προγραμματιζόμενη
VTTX	Τροφοδοσία τάσης τερματισμού εξόδου	1.8	2.625	V	
VTCM	Εύρος τάσης εξόδου	1.1	1.5	V	

	κοινής λειτουργίας (χωρίς σύνδεση γραμμής αποστολής)					
VTCM	Εύρος τάσης εξόδου κοινής λειτουργίας (με σύνδεση γραμμής αποστολής)	1.1		2.0	V	Η κοινή λειτουργία εξαρτάται από τη σύζευξη (DC ή AC). VTTX, VTRX και διαφορική ταλάντωση. Η προσομοίωση στο Spice δίνει την ακριβή τάση κοινής λειτουργίας για όποιο δοσμένο σύστημα
VISKEW	Στρέβλωση διαφορικής εξόδου			15	ps	

Πίνακας 2-13 : Παράμετροι Διαφορικού Πομποδέκτη

### Διαφορικός Δέκτης

Ο διαφορικός δέκτης δέχεται τα σήματα VP και VN signals, πραγματοποιώντας τον υπολογισμό της διαφοράς VP –VN ηλεκτρονικά. Όλα τα δεδομένα εισόδου πρέπει να είναι διαφορικά και ονομαστικά πολωμένα σε μια τάση κοινής κατάστασης λειτουργίας των 0.5 V – 2.5 V, ή συζευγμένα σε AC. Εσωτερικές απολήξεις προσφέρουν απλή σύνδεση γραμμής μετάδοσης των 50Ω ή 75Ω. Βλέπε Σχήμα 2-12.

### Τεχνική Προ-έμφασης

Ίσως το πιο σημαντικό χαρακτηριστικό ενός multi-gigabit οδηγητή είναι η ικανότητά του να εφαρμόζει την τεχνική της προ-έμφασης. Ως προ-έμφαση ορίζεται η σκόπιμη υπεροδήγηση, δηλαδή αύξηση του πλάτους, μιας ζώνης συχνοτήτων (συνήθως υψηλών), στην αρχή μιας μετάβασης (transition), ώστε να βελτιωθεί ο λόγος σήματος προς θόρυβο κατά την λήψη της μετάδοσης. Με απλά λόγια, η τεχνική της προ-έμφασης δίνει έμφαση στις υψηλές συχνότητες ενός σήματος. Αυτή η τεχνική χρησιμεύει στην επίλυση του προβλήματος της αλληλοπαρεμβολής συμβόλων (Inter Symbol Interference ISI) που προκαλεί υποβάθμιση της επίδοσης του συστήματος μετάδοσης. Για την καλύτερη κατανόηση του φαινομένου, όταν σε ένα κανάλι ο ρυθμός μετάδοσης δεδομένων είναι πολύ υψηλός και ξεπερνάει το εύρος ζώνης της γραμμής μετάδοσης τότε τα δυαδικά δεδομένα δεν προλαβαίνουν να ολοκληρώσουν τη μετάβαση από ένα σύμβολο σε άλλο μέσα στη διάστημα που μεσολαβεί μεταξύ αυτών, με αποτέλεσμα την εξάπλωσή τους στα γειτονικά σύμβολα.

Στην προ-έμφαση, η αρχική διαφορική ταλάντωση ενισχύεται ώστε να δημιουργηθεί μια ισχυρότερα αύξουσα ή φθίνουσα κυματομορφή. Αυτή η μέθοδος αναπληρώνει για την απώλεια σε υψηλές συχνότητες στο μέσο μετάδοσης που αλλιώς θα περιόριζε το πλάτος αυτής της κυματομορφής. Τα αποτελέσματα της προ-έμφασης φαίνονται σε εικόνα στιγμιότυπου τεσσάρων οπτικών, από το Σχήμα 2-13 μέχρι το Σχήμα 2-15 στην ακόλουθη σελίδα.

Η επισήμανση STRONG στο Σχήμα 2-13 χρησιμοποιείται για να δείξει ότι η κυματομορφή είναι μεγαλύτερη ως προς το πλάτος τάσης, σε αυτό το σημείο, συγκριτικά με το LOGIC ή κανονικό επίπεδο (π.χ. χωρίς προ-έμφαση).



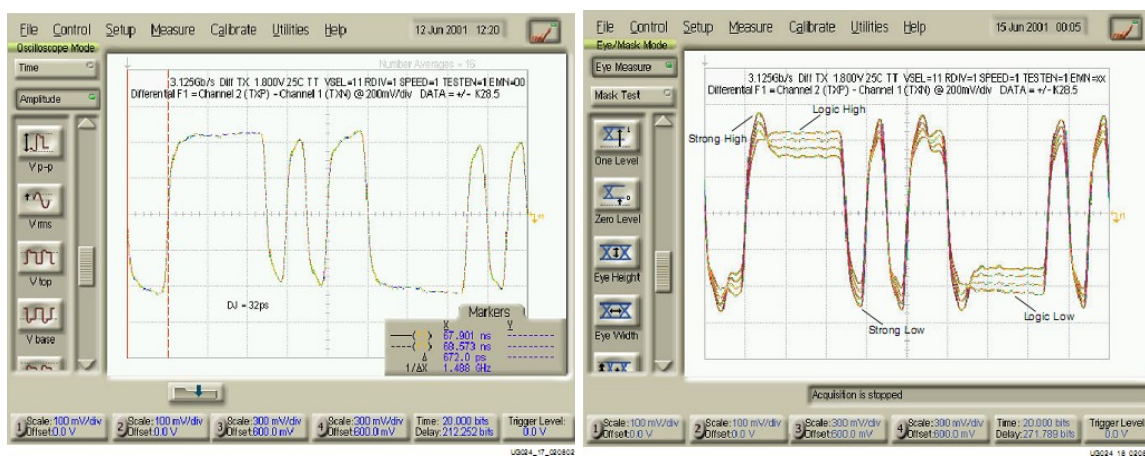
Ένα δεύτερο χαρακτηριστικό της προ-έμφασης του πομποδέκτη RocketIO είναι ότι το επίπεδο STRONG ελαττώνεται μετά από κάποιο χρόνο στο επίπεδο LOGIC, κι έτσι ελαχιστοποιείται η ταλάντωση τάσης που είναι αναγκαία για τη μεταγωγή του διαφορικού ζεύγους στην αντίθετη κατάσταση.

Οι γραμμές μετάδοσης με απώλειες προκαλούν την σπατάλη ηλεκτρικής ενέργειας. Αυτή η τεχνική προ-έμφασης επεκτείνει την απόσταση κατά την οποία μπορούν να οδηγηθούν τα σήματα εντός μέσου γραμμής με απώλειες και αυξάνει το λόγο σήματος προς θόρυβο στο δέκτη.

Πρέπει να σημειωθεί ότι ρυθμίσεις με υψηλή προ-έμφαση δεν είναι κατάλληλες για μικρές διασυνδέσεις (ένα κομμάτι του μέγιστου μήκους των 40 ιντσών του FR4). Η παραπάνω προ-έμφαση μπορεί στην πράξη να υποβαθμίσει το ρυθμό σφαλμάτων bit (BER) μιας σύνδεσης multi-gigabit. Προσεκτική προσομοίωση και/ή εργαστηριακές δοκιμές του συστήματος θα έπρεπε πάντα να χρησιμοποιούνται ώστε να διασφαλίσουν ότι χρησιμοποιείται η βέλτιστη ρύθμιση προ-έμφασης. Συμβουλευθείτε την Περίληψη Χαρακτηρισμού του Virtex-II Pro RocketIO™ Multi-Gigabit Πομποδέκτη για πιο λεπτομερείς πληροφορίες στις αναμενόμενες κυματομορφές στα διάφορα επίπεδα προ-έμφασης. Τα τέσσερα επίπεδα προ-έμφαση φαίνονται στον Πίνακα 2-14.

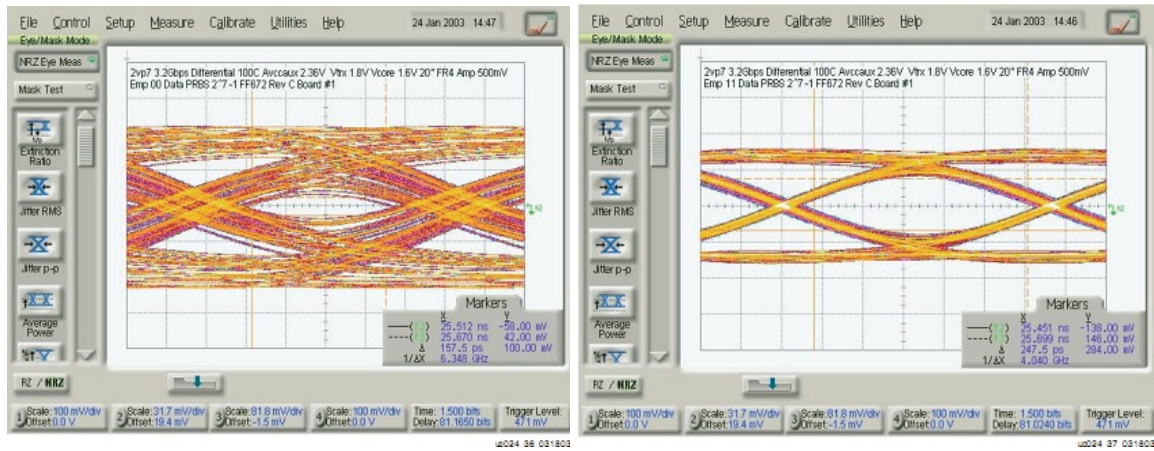
Τιμή Χαρακτηριστικού	Έμφαση (%)
0	10
1	20
2	25
3	33

Πίνακας 2-14 : Τιμές Προ-έμφασης



Σχήμα 2-13 : Εναλλακτικό K28.5+ χωρίς Προ-έμφαση απέναντι K28.5+ με Προ-έμφαση



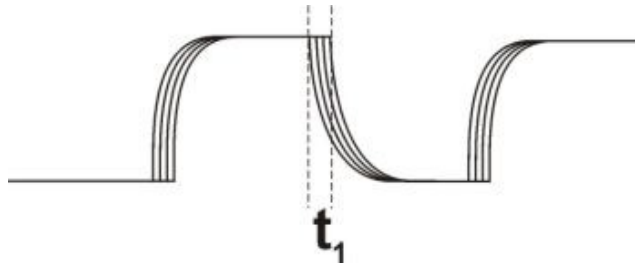


Σχήμα 2-14 : Διάγραμμα Οφθαλμού, 10% Προ-έμφαση, 20" FR4, Συνθήκες Χειρότερης Περίπτωσης

Σχήμα 2-15 : Διάγραμμα Οφθαλμού, 33% Προ-έμφαση, 20" FR4, Συνθήκες Χειρότερης Περίπτωσης

### Απόκλιση σήματος (Jitter)

Ο διεθνής οργανισμός τηλεπικοινωνιών (ITU) περιγράφει το jitter ως "τις στιγμιαίες αποκλίσεις καθοριστικών τμημάτων ενός ψηφιακού σήματος σε σχέση με τις ιδανικές θέσεις τους στον χρόνο".



Σχήμα 2-16 : Σχήμα απόκλισης του παλμού ενός σήματος

Οποιοδήποτε πρόβλημα επαγόμενου θορύβου στο εσωτερικό μίας συσκευής επιδρά αρνητικά στα ψηφιακά κυκλώματα και δημιουργεί μία περιοχή αβεβαιότητας για το πού βρίσκεται το σήμα χρονισμού ( $t_1$ ). Η απόκλιση αυτή του σήματος γνωστή και ως jitter, εκφράζεται τυπικά ως δεκαδικό τμήμα της Unit Interval (UI), π.χ. 0.3 UI. Η συνολική απόκλιση σήματος (total jitter), υπολογίζεται ως εξής:

$$\text{Total Jitter} = \text{Deterministic Jitter (DJ)} + \text{Random Jitter (RJ)}.$$

Deterministic Jitter (DJ) ορίζεται ως η απόκλιση σήματος που εξαρτάται από το συμβολισμό των δεδομένων, που αποδίδεται σε μια μοναδική πηγή (π.χ. Inter Symbol Interference ISI) και προκαλείται από την επίδραση φαινομένων απωλειών του μέσου μετάδοσης. Η DJ είναι γραμμικά προσθετική.

Random Jitter (RJ) υπάρχει λόγω τυχαίων πηγών, όπως το υπόστρωμα, τροφοδοτικό, κτλ. Το RJ είναι προσθετικό όπως το άθροισμα τετραγώνων, και ακολουθεί μια κωδωνωτή καμπύλη.

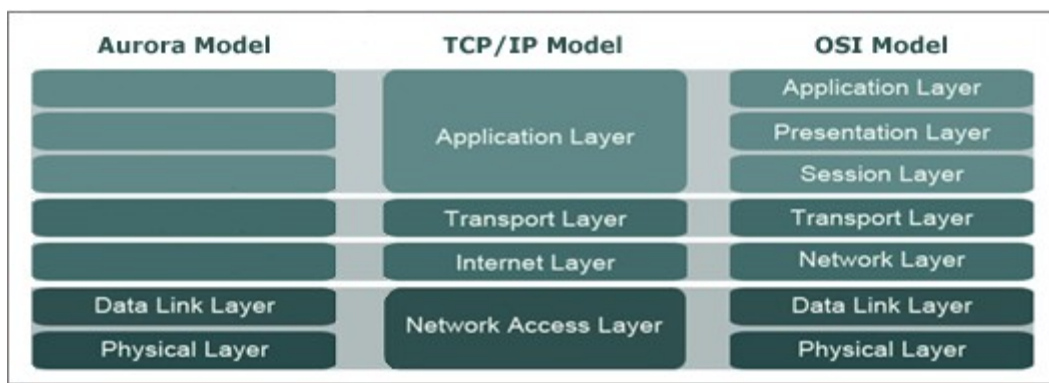
## 2.3 Πρωτόκολλο επιπέδου διασύνδεσης Aurora

### 2.3.1 Επισκόπηση

Σε αυτήν την ενότητα περιγράφεται λεπτομερώς το πρωτόκολλο Aurora που χρησιμοποιεί η σχεδίαση της εργασίας αυτής, το οποίο υλοποιεί μέχρι και το επίπεδο διασύνδεσης δικτύου. Αναλύονται οι προδιαγραφές του πρωτοκόλλου αυτού, οι παράμετροί του και επισημαίνονται όλες οι σημαντικές λεπτομέρειες που χρησίμευσαν στην περαιτέρω ανάπτυξη της σχεδίασης της εργασίας αυτής.

### 2.3.2 Γενικά για τα επίπεδα δικτύου

Ενδεικτικά ένα από τα πιο δημοφιλή πρωτόκολλα μεταφοράς το TCP/IP και τα επίπεδα ανάπτυξής του όσον αφορά το μοντέλο OSI παρατίθενται παρακάτω με σκοπό να γίνει πιο εύκολα αντιληπτό το επίπεδο δίκτυο που υλοποιεί το Aurora και σε πιο επίπεδο υπηρεσιών δικτύου αντιστοιχεί στην πράξη σε σύγκριση με ένα δημοφιλές πρωτόκολλο:



Σχήμα 2-17 : Σύγκριση μοντέλου OSI και επιπέδων δικτύου TCP/IP

Σε αυτήν την εργασία χρησιμοποιείται το πρωτόκολλο που υλοποιεί μέχρι και το επίπεδο διασύνδεσης (Data Link Layer). Ως γνωστόν, το επίπεδο διασύνδεσης δεδομένων παρέχει τα λειτουργικά και διαδικαστικά μέσα για την μεταφορά δεδομένων από την μια συσκευή του δικτύου στην άλλη, και για τον έλεγχο και την πιθανή διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Οι μη ιεραρχημένες διευθύνσεις των συσκευών εδώ είναι οι φυσικές (π.χ. MAC διευθύνσεις), δηλαδή είναι καλωδιωμένες στις κάρτες δικτύου των συσκευών από το εργοστάσιο.

### 2.3.3 Επισκόπηση πρωτοκόλλου επιπέδου διασύνδεσης Aurora

Το Aurora είναι ένα κλιμακωτό κι ελαφρύ πρωτόκολλο επιπέδου διασύνδεσης που χρησιμοποιείται για τη μετακίνηση δεδομένων κατά μήκος σειριακών συνδέσεων σημείο-προς-σημείο. Είναι ένα ανοιχτό πρωτόκολλο, χωρίς χρέωση, και μπορεί να

υλοποιηθεί σε όποια συσκευή/τεχνολογία πυριτίου. Παρέχει μια αόρατη διεπαφή στις φυσικές σειριακές συνδέσεις, επιτρέποντας τα ανώτερα επίπεδα ιδιωτικών ή καθιερωμένων από τη βιομηχανία πρωτοκόλλων, όπως το Ethernet και το TCP/IP, να χρησιμοποιήσουν εύκολα αυτές τις σειριακές συνδέσεις υψηλών-ταχυτήτων. Αυτό οδηγεί σε υψηλότερη απόδοση διασύνδεσης ενώ διατηρεί την επένδυση σε δεδομένες δομές λογισμικού. Πρόκειται για ένα πρωτόκολλο πολύ αποδοτικό και χαμηλού λανθάνοντος χρόνου που χρησιμοποιεί το λιγότερο δυνατό ποσό λογικής. Το Aurora προσφέρει ένα πλούσιο, σε μεγάλο ποσοστό επαναπροσδιοριζόμενο, σύνολο γνωρισμάτων. Το Aurora αυξάνει το εύρος ζώνης μέσω των απεριόριστων δεσμών γραμμών. Χρειάζεται πολύ λίγο χρόνο να ενσωματωθεί σε υπάρχουσες σχεδιάσεις του χρήστη. Όντας ελαφρύ, το Aurora λειτουργεί άψογα για σειριακή διασύνδεση σημείο-προς-σημείο. Είναι αποτελεσματικό λόγω της χαμηλής ποσότητας επιπλέον πληροφορίας (overhead) πρωτοκόλλου και χαμηλού πλήθους πυλών. Αποτελεί πρωτόκολλο κλιμακούμενο γιατί οι διασυνδέσεις του Aurora μπορούν να χρησιμοποιήσουν αυθαίρετο αριθμό μεταδοτών υψηλών ταχυτήτων σε δεσμό τρέχοντας σε ευρύ φάσμα συχνοτήτων. Χαρακτηρίζεται από απλό «εμφωλισμό». Ακόμα υποστηρίζει επικοινωνίες από chip σε chip, από κουτί σε κουτί και backplane.



Σχήμα 2-18 : Διεπαφή Πομποδέκτη

Σχήμα 2-19 : Διεπαφή Aurora και Πομποδέκτη και Χαρακτηριστικά Aurora

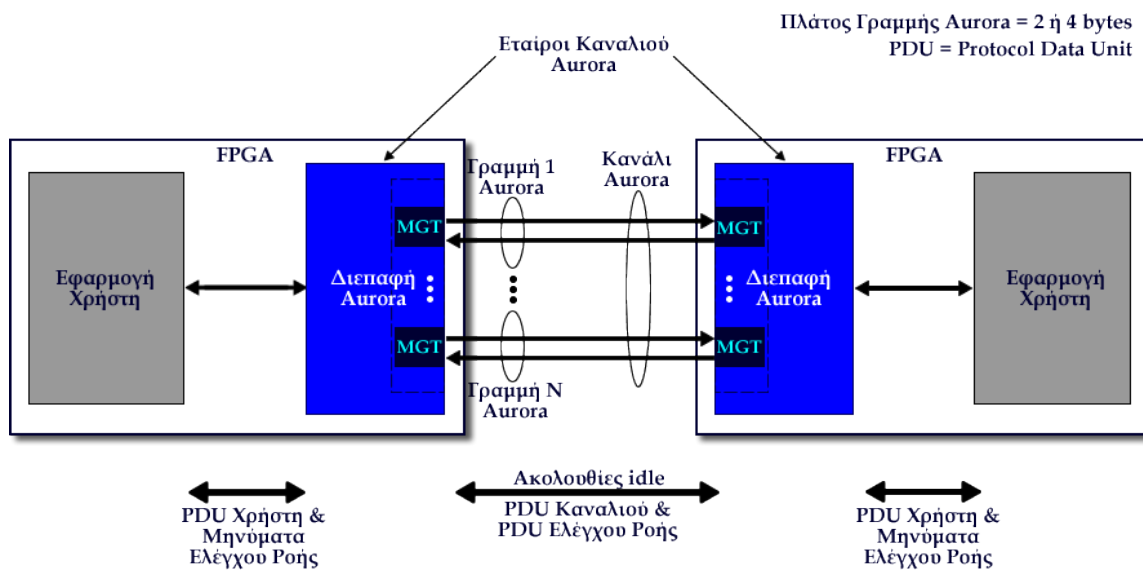
Η Xilinx παρέχει τις σχεδιάσεις αναφοράς του Aurora και τα παρελκόμενα για Virtex-II Pro™ και Virtex-II Pro™X. Η σχεδίαση αναφοράς του Aurora αναπτύχθηκε από την Xilinx, παρέχει μια απλή διεπαφή LocalLink με την απαραίτητη λειτουργικότητα επιπέδου διασύνδεσης και διαχειρίζεται τη διεπαφή ελέγχου του RocketIO™. Οι σχεδιάσεις αναφοράς μπορούν να αδειοδοτηθούν για χρήση ως πυρήνες ASIC. Ενώ είναι διαθέσιμο και Bus functional μοντέλο για την ανάπτυξη του Aurora.

#### Προδιαγραφές του πρωτοκόλλου Aurora

- Οι προδιαγραφές του πρωτοκόλλου Aurora ορίζουν:
  - Φυσικό στρώμα: Ηλεκτρονικά χαρακτηριστικά & φυσική κωδικοποίηση
  - Στρώμα Διασύνδεσης: Τοποθέτηση σε «καλούπι» των δεδομένων χρήστη, αρχικοποίηση καναλιού και αντιμετώπιση σφαλμάτων, διαδικασίες αποστολής & λήψης
  - Διαμοιρασμός Δεδομένων: Πώς τα δεδομένα κατανέμονται μεταξύ πολλαπλών γραμμών
  - Μηχανισμοί ελέγχου ροής: Έλεγχος ρυθμού μεταφοράς και αποστολή μηνυμάτων ελέγχου υψηλής προτεραιότητας

- Το πρωτόκολλο δεν διευθετεί:
  - CRC: οι χρήστες δεν υποχρεούνται να χρησιμοποιήσουν κάποιο συγκεκριμένο μοντέλο διόρθωσης σφαλμάτων
  - Μεταγωγή δεδομένων σε δίκτυο (switching)
- Το πρωτόκολλο Aurora ικανοποιεί τις ανάγκες επιπέδου διασύνδεσης: δεν παραφορτώνει τις εφαρμογές του χρήστη με ακατάλληλες συναρτήσεις επιπέδου εφαρμογών

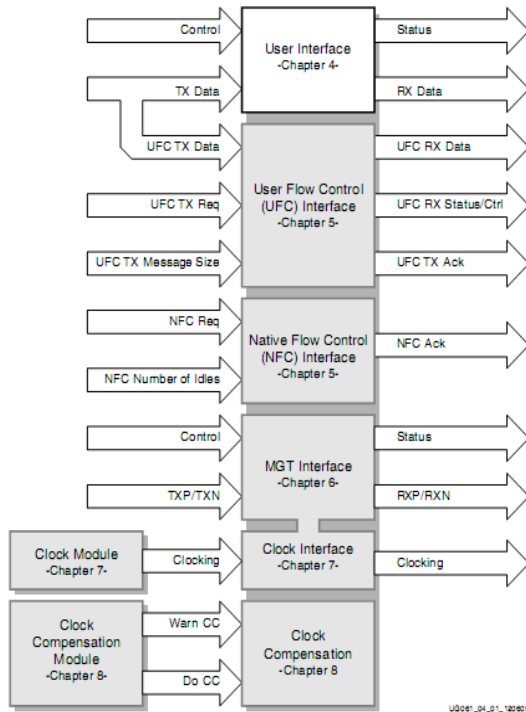
Aurora με λίγα λόγια



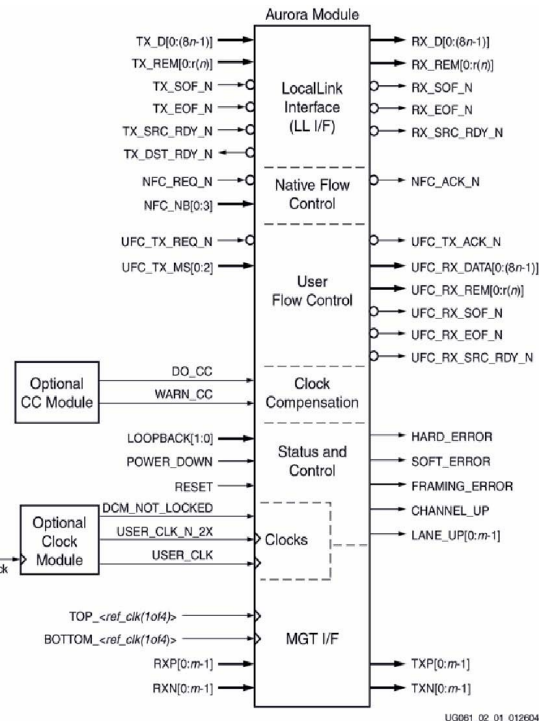
Σχήμα 2-20 : Aurora Συνοπτικά

Όλα τα σύμβολα στο κανάλι είναι κωδικοποιημένα σε 8B/10B και αποστέλλονται ως κωδικοποιημένες ομάδες των 2 συμβόλων. Κάθε υψηλών ταχυτήτων σειριακή σύνδεση μεταξύ MGT καλείται γραμμή (**lane**) και πιο συγκεκριμένα στις συσκευές Virtex-II Pro™, κάθε γραμμή έχει ρυθμό μετάδοσης συμβόλων δεδομένων από 600 Mbps ως 3.125 Gbps ενώ στις συσκευές Virtex-II Pro™X, κάθε γραμμή επιτυγχάνει από 1 Gbps ως 10 Gbps. Οποιοσδήποτε αριθμός γραμμών μπορεί να χρησιμοποιηθεί σε δεσμό ώστε να δημιουργηθεί ένα **κανάλι Aurora**. Στις συσκευές Virtex-II Pro και Virtex-II Pro X πάνω από 20 γραμμές είναι διαθέσιμες για δεσμό. Τα άκρα ενός καναλιού Aurora καλούνται **εταίροι καναλιού**. Τα δεδομένα μεταφέρονται μεταξύ των εταίρων καναλιού και των εφαρμογών του χρήστη σε μονάδες που λέγονται μονάδες δεδομένων πρωτοκόλλου χρήστη (**user Protocol Data Units PDU**). Οι PDU που μεταφέρονται μεταξύ εταίρων καναλιού του Aurora περνάνε από το κανάλι Aurora Channel ως **PDU καναλιού**. Οι PDU καναλιού έχουν ένα ελάχιστο μέγεθος των έξι byte και κανένα μέγιστο μέγεθος ούτε απαιτούμενο κενό (με idles) μεταξύ πλαισίων δικτύου, που λέγεται IFG (InterFrame Gap). Οι PDU καναλιού μοιράζονται το κανάλι Aurora με PDU Ελέγχου Ροής κι ακολουθίες διόρθωσης ρολογιού. Όταν το κανάλι Aurora δεν

χρησιμοποιείται για την αποστολή δεδομένων, συμπληρώνεται με μια τυχαία ακολουθία Idle. Το Aurora χρησιμοποιεί τους ίδιους χαρακτήρες Idle όπως το πρωτόκολλο XAUI, ταξινομημένους τυχαία για χαμηλή EMI. Το Aurora χρησιμοποιεί κωδικοποίηση 8B/10B για ισορροπία DC, εντοπισμό σφάλματος και για την αποδοχή χαρακτήρων ελέγχου στη ροή δεδομένων. Παρουσιάζεται top level διεπαφής χρήστη και Block Diagram σχεδίασης αναφοράς του Aurora



Σχήμα 2-21 : Αφαιρετικό Top-Level Διεπαφής Χρήστη Aurora

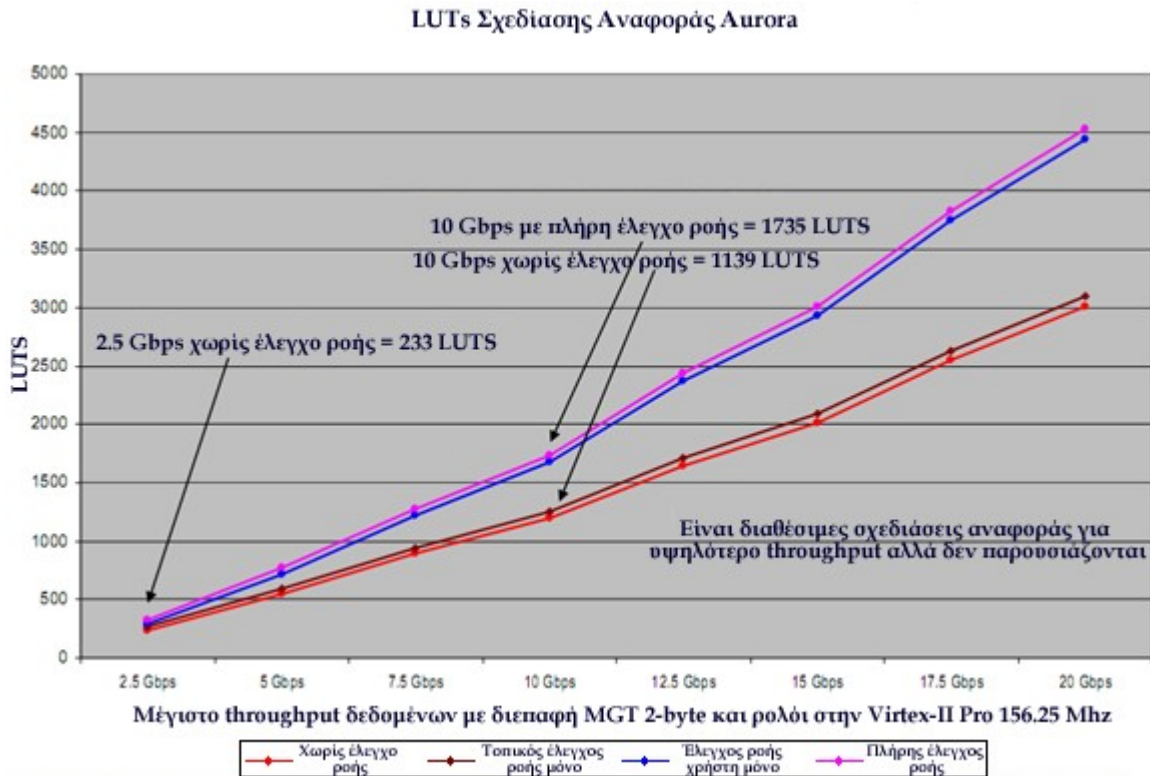


Σχήμα 2-22 : Top-Level Διεπαφής χρήστη Aurora

## Γνωρίσματα της σχεδίασης αναφοράς του Aurora

Η σχεδίαση αναφοράς του Aurora παρέχει εύκολη προς ενσωμάτωση διεπαφή χρήστη. Ενσωματώνει τα Multi-gigabit κανάλια επικοινωνίας στις συσκευές V-II Pro™ και V-II Pro™X. Περιλαμβάνει απλή διεπαφή για τη διαχείριση λειτουργιών και διεπαφή LocalLink για το μονοπάτι δεδομένων. Διαθέτει ακόμα παραμετροποιήσιμα γνωρίσματα. Έτσι η σχεδίαση αναφοράς υποστηρίζει 1 έως 20 γραμμές ανά κανάλι. Οι MGT μπορούν να ρυθμιστούν για διεπαφή 2-byte ή 4-byte. Όλα τα γνωρίσματα Ελέγχου Ροής του Aurora είναι προαιρετικά. Τέλος η σχεδίαση του Aurora παραδίδεται από τον CORE Generator με πηγαίο κώδικα VHDL ή Verilog. Εκτελείται αυτόματη αρχικοποίηση στην εκκίνηση, επανεκκίνηση ή κατόπιν καταστροφικού σφάλματος. Παρουσιάζονται δείγματα μεγεθών της σχεδίασης αναφοράς του Aurora.





Σχήμα 2-23 : Throughput δεδομένων συναρτήσει του αριθμού των LUTs

### Πλάτος Διεπαφής

Το Aurora χρησιμοποιεί ομάδες κωδικοποίησης των 2-byte που ταιριάζει φυσιολογικά στη διεπαφή των 2-byte του MGT. Ο MGT επίσης έχει διεπαφή των 4-byte, όπου το ρολόι χρήστη USER\_CLK πρέπει να «τρέξει» στη μισή ταχύτητα του ρολογιού αναφοράς. Επίσης ο CORE Generator Aurora 2.1 υποστηρίζει τη λειτουργία 4-byte. Σε κάποιες περιπτώσεις, η λειτουργία 4-byte μπορεί να προτιμάται. Η λειτουργία 4-byte ενεργοποιεί τα 5-10 Gbaud ανά γραμμή όταν χρησιμοποιούνται συσκευές Virtex-II Pro™X.

Πλάτος Διεπαφής του MGT (σε Bytes)	2	4
Πλάτος Διεπαφής Χρήστη του LocalLink (σε Bytes) N = # lanes	2 x N	4 x N

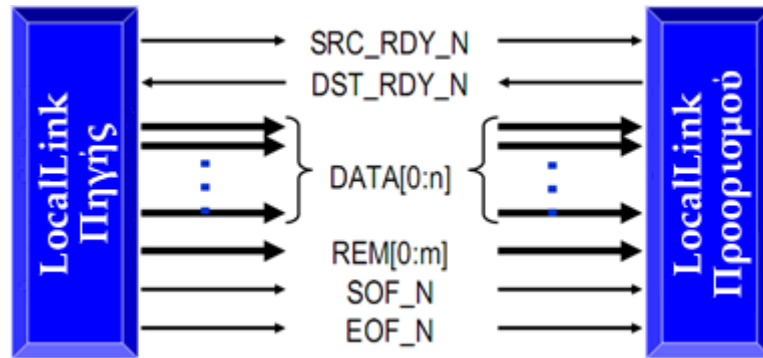
Πίνακας 2-15 : Πλάτος Διεπαφής Χρήστη σε Σχέση με τη Διεπαφή του MGT

### Τροποποιήσεις Επιπέδου Πηγής

Μπορεί να αλλαχθεί ο πηγαίος κώδικας της σχεδίασης αναφοράς ώστε να εκπληρωθούν συγκεκριμένες ανάγκες σχεδίασης. Μερικές συνηθισμένες τροποποιήσεις περιλαμβάνουν αλλαγές στα επίπεδα τάσης του MGT και/ή της προ-έμφαση. Αυτές οι παράμετροι επιτρέπουν τη βελτιστοποίηση της σηματοδοσίας του MGT για την πλακέτα. Υπάρχει δυνατότητα επίσης κι αλλαγών στη Διόρθωση Ρολογιού. Οι προδιαγραφές της διόρθωσης ρολογιού του Aurora επιτρέπουν διαφορά μέχρι 200 ppm των ρολογιών κάθε εταίρου καναλιού. Η συχνότητα διόρθωσης ρολογιού του Aurora και το μήκος ακολουθίας μπορεί να μεταβληθεί ώστε να πληροί τις προϋποθέσεις της διόρθωσης

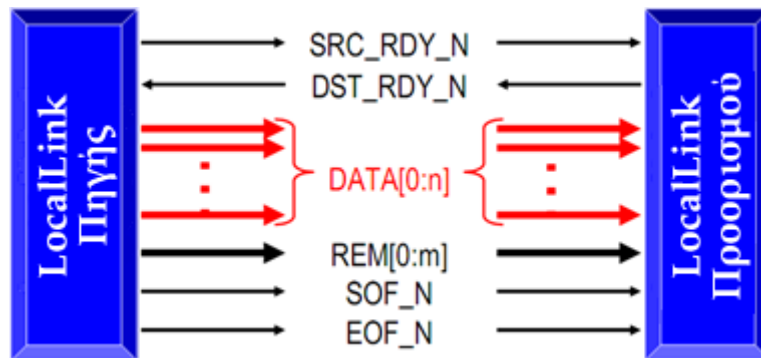
ρολογιού της εφαρμογής. Για εφαρμογές όπου οι PDU δεν μπορούν να αντέξουν διακοπές, η διόρθωση ρολογιού πρέπει να ελεγχθεί προσεκτικά. Η σχεδίαση αναφοράς το ενεργοποιεί αυτό με τη θύρα της για Διόρθωση Ρολογιού.

Διεπαφή LocalLink Πρωτοκόλλου Aurora



Σχήμα 2-24 : Διεπαφή του LocalLink

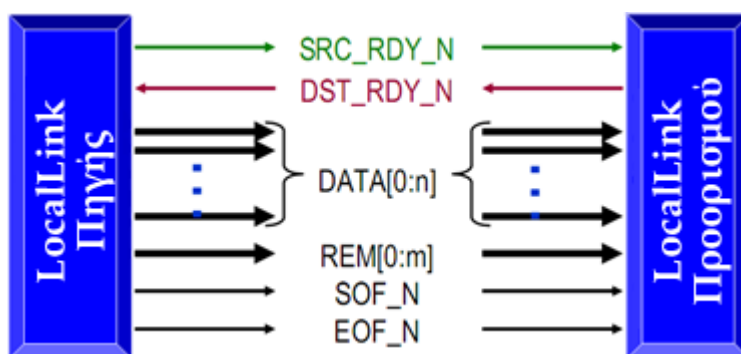
Το LocalLink είναι μια καθιερωμένη διεπαφή IP της Xilinx που χρησιμοποιείται στη σχεδίαση αναφοράς του Aurora για το μονοπάτι δεδομένων των PDU και τη διεπαφή RX για Έλεγχο Ροής Χρήστη. Πρόκειται για σύγχρονη διεπαφή εφόσον τα σήματα δειγματοληπτούνται μόνο στις ακμές ρολογιού.



Σχήμα 2-25 : Διεπαφή Δεδομένων του LocalLink

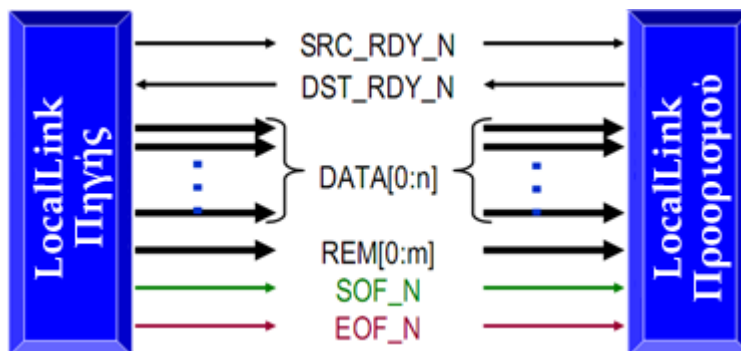
Οι θύρες δεδομένων του LocalLink είναι κλιμακούμενες κι έτσι μια λέξη του LocalLink μπορεί να απαρτίζεται από οποιονδήποτε αριθμό bytes. Στο Aurora ο αριθμός των byte σε κάθε λέξη είναι δύο φορές ο αριθμός των γραμμών στο κανάλι (ή τέσσερις φορές, εάν η επιλογή της διεπαφής 4-byte χρησιμοποιείται), όπως φαίνεται στον Πίνακα 2-15. Τα δεδομένα στο LocalLink ρέουν από την πηγή στο δέκτη.





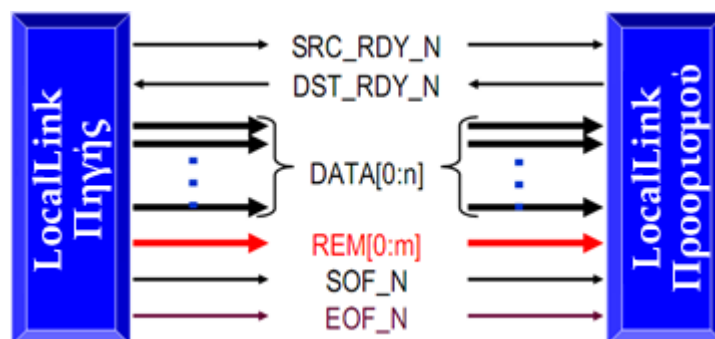
Σχήμα 2-26 : Σήματα Έναρξης και Παύσης της Ροής Δεδομένων

Η πηγή πρέπει να υποδεικνύει ότι τα σήματα μες στην διεπαφή είναι έγκυρα ενεργοποιώντας το σήμα SRC\_RDY\_N. Ο δέκτης δε χρειάζεται να δειγματοληπτεί τη διεπαφή LocalLink σε κάθε κύκλο. Στους κύκλους όπου ο δέκτης δειγματοληπτεί τα σήματα της διεπαφής LocalLink, ενεργοποιεί το DST\_RDY\_N. Οι συνδιαλλαγές του LocalLink διαδραματίζονται μόνο στις ακμές του ρολογιού όπου και τα δύο το SRC\_RDY\_N και το DST\_RDY\_N είναι ενεργοποιημένα



Σχήμα 2-27 : Σήματα Αρχής και Τέλους Πακέτου Δεδομένων

Τα δεδομένα του LocalLink είναι σε πλαίσιο και τα δεδομένα εκτός πλαισίου θεωρούνται άκυρα. Η έναρξη του πλαισίου του LocalLink σηματοδοτείται από την ενεργοποίηση του SOF\_N. Ο τερματισμός ενός πλαισίου του LocalLink γίνεται όταν το EOF\_N ενεργοποιείται. Το SOF\_N και EOF\_N πρέπει να ενεργοποιείται την ίδια στιγμή με μια λέξη (ή μέρος λέξης) δεδομένων. Οι PDU του χρήστη αποστέλλονται ως πλαίσια του LocalLink



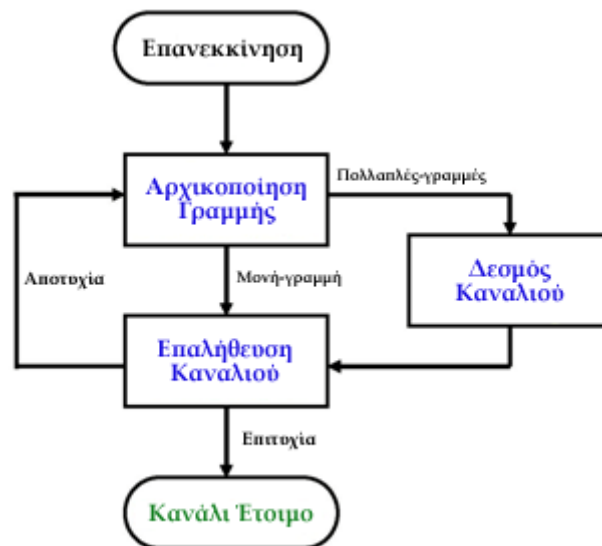
Σχήμα 2-28 : Σήμα Έγκυρων Bytes Τελευταίας Λέξης Πακέτου

Κάποια πλαίσια δε χωράνε σε ακέραιο αριθμό λέξεων του LocalLink. Για να τα φιλοξενήσει, το LocalLink επιτρέπει η τελευταία λέξη του πλαισίου να είναι μερικώς συμπληρωμένη. Το σήμα REM είναι δυαδική κωδικοποίηση του αριθμού των έγκυρων bytes στην τελική λέξη – 1 π.χ., 1 byte έγκυρο, REM= 0 σε δυαδικό. 4 byte έγκυρα, REM= 3 σε δυαδικό. Η θύρα REM είναι αρκετά ευρεία για τη δυαδική κωδικοποίηση του αριθμού των bytes σε μια ολόκληρη λέξη. Το REM είναι έγκυρο μόνο στην τελευταία λέξη ενός πλαισίου – όταν είναι ενεργοποιημένο το EOF\_N. Τα bytes στη λέξη του LocalLink ξεκινούν από τα αριστερά και συμπληρώνονται προς τα δεξιά.

### Αρχικοποίηση Καναλιού

Πριν οι εταίροι καναλιού του Aurora μπορούν να ανταλλάξουν δεδομένα, πρέπει πρώτα να αρχικοποιήσουν ένα κανάλι Aurora. Η αρχικοποίηση καναλιού διαδραματίζεται όταν οποιοσδήποτε από τους εταίρους καναλιού εκκινεί, επανεκκινεί ή ανανίπτει από καταστροφικό σφάλμα. Πρόκειται για λειτουργία τριών φάσεων:

- Αρχικοποίηση κάθε γραμμής
- Δημιουργία δεσμού μεταξύ των γραμμών εφόσον χρειάζεται
- Επικύρωση καναλιού

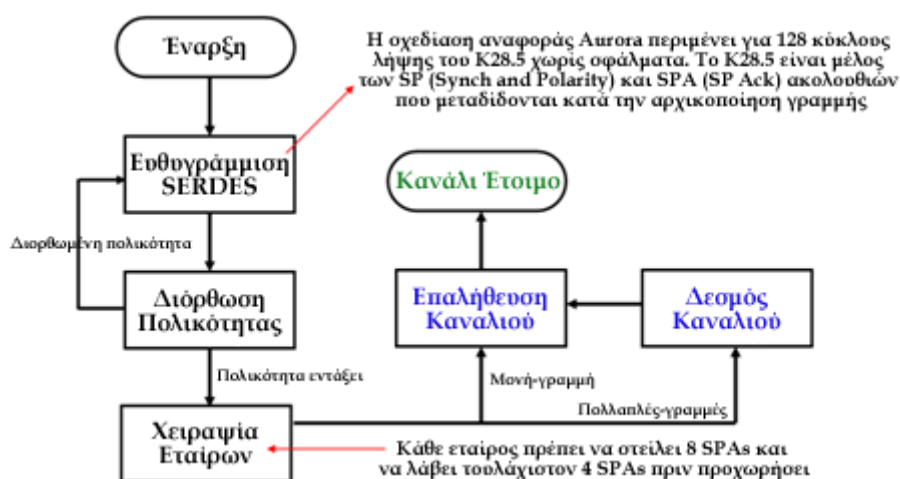


Σχήμα 2-29 : Μηχανή Πεπερασμένων Καταστάσεων για Αρχικοποίηση Καναλιού

Η σχεδίαση αναφοράς του Aurora τυπικά χρειάζεται < 800 κύκλους χρήστη για να ολοκληρώσει την αρχικοποίηση

### Αρχικοποίηση Γραμμής

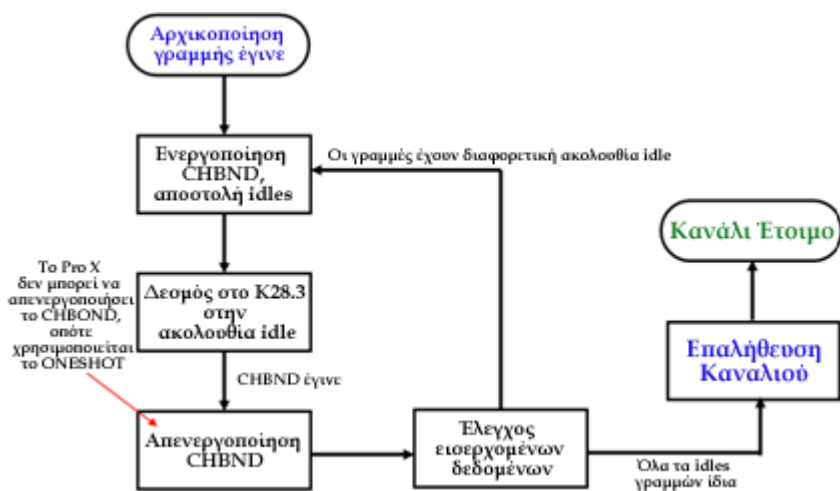
Αρχικοποιούνται οι σειριακές συνδέσεις μεταξύ αντίστοιχων MGTs σε κάθε εταίρο καναλιού



Σχήμα 2-29 : Μηχανή Πεπερασμένων Καταστάσεων για Αρχικοποίηση Γραμμής

### Δημιουργία Δεσμού Καναλιού

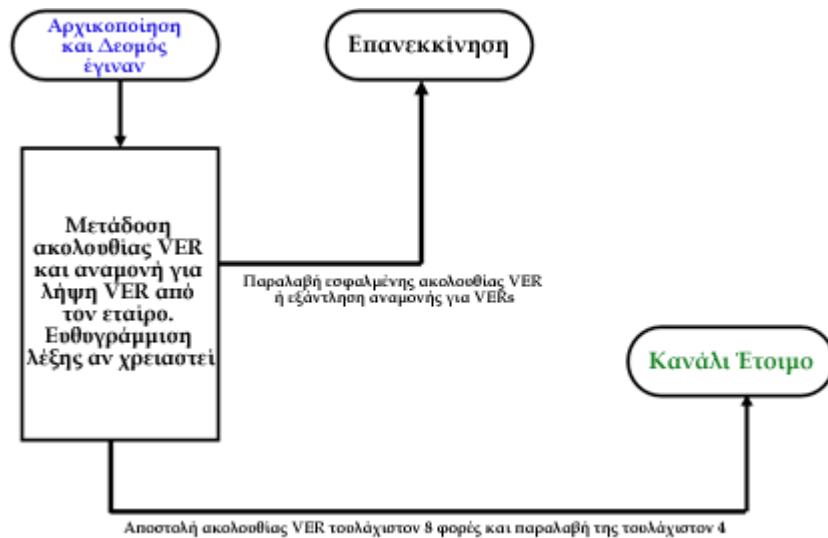
Συνδυάζονται οι γραμμές σε ένα απλό κανάλι Aurora. Η ίδια ακολουθία Idle αναμεταδίδεται σε κάθε γραμμή. Η ακολουθία Idle του Aurora περιέχει, τυχαία διαχωρισμένα με κενό, σύμβολα K28.3. Η σχεδίαση αναφοράς του Aurora έχει ένα χρονομετρητή watchdog που επανεκκινεί το εν λόγω λειτουργικό τμήμα εάν η δημιουργία δεσμού πάρει πολύ χρόνο



Σχήμα 2-29 : Μηχανή Πεπερασμένων Καταστάσεων για Δεσμό Καναλιού

### Επικύρωση Καναλιού

Επαληθεύεται ότι το κανάλι είναι έτοιμο και εκτελείται ευθυγράμμιση λέξης εάν χρειάζεται. Η ακολουθία VER που μεταδίδεται είναι 60 Idle bytes ακολουθούμενα από K28.5, D8.7, D8.7, D8.7 σε κάθε γραμμή. Η ευθυγράμμιση λέξης είναι μερικές φορές απαραίτητη μετά από τη δημιουργία δεσμού καναλιού επειδή οι ευθυγραμμισμένες κατά MSB γραμμές σκλάβοι μπορεί να χρειάζεται να ολισθήσουν την ευθυγράμμισή τους αναλόγως με τη στρέβλωση σήματός (skew) τους.



Σχήμα 2-30 : Μηχανή Πεπερασμένων Καταστάσεων για Επικύρωση Καναλιού

### Κατάσταση και Διεπαφή Συστήματος της Σχεδίασης Αναφοράς του Aurora

Η σχεδίαση αναφοράς του Aurora έχει πολλές θύρες για τη διαχείριση του καναλιού του Aurora, όπως

- **USER\_CLK**: αυτό το ρολόι είναι το κοινόχρηστο ρολόι μεταξύ του Aurora και της εφαρμογής χρήστη
- **RESET**: αυτό το σήμα επανεκκινεί το λειτουργικό τμήμα Aurora, προκαλώντας την επανεκκίνηση των MGT και την επαναρχικοποίηση του καναλιού Aurora
- **POWERDOWN**: αυτό το σήμα κλείνει τους MGT, βάζοντάς τους σε κατάσταση λειτουργίας χαμηλής ενέργειας. Επίσης επανεκκινεί τους MGTs και συνεπώς το κανάλι Aurora.
- **DCM\_NOT\_LOCKED**: εάν χρησιμοποιείται ένα DCM για τη δημιουργία του σήματος USER\_CLK, το σήμα DCM\_NOT\_LOCKED θα πρέπει να χρησιμοποιείται για να λέει στο λειτουργικό τμήμα του Aurora πότε η έξοδος του DCM είναι έγκυρη
- **LOOPBACK**: αυτό το σήμα χρησιμοποιείται για τον έλεγχο των ενσωματωμένων κυκλωμάτων βρόχου στους MGTs. 00 απενεργοποιεί το βρόχο, 10 ενεργοποιεί το σειριακό βρόχο και 01 ενεργοποιεί τον εσωτερικό παράλληλο βρόχο

Τα ακόλουθα σήματα επιτρέπουν στους χρήστες να παρακολουθούν την κατάσταση του καναλιού Aurora

- **LANE\_UP**: κάθε γραμμή στο λειτουργικό τμήμα Aurora έχει ένα σήμα Lane up. Ενεργοποιείται όταν η γραμμή έχει ολοκληρώσει την αρχικοποίηση γραμμής
- **CHANNEL\_UP**: ενεργοποιείται όταν η αρχικοποίηση καναλιού ολοκληρώνεται  
**Σημείωση**: το λειτουργικό τμήμα μπορούν να αρχίσει να λαμβάνει δεδομένα πριν ενεργοποιηθεί το σήμα Channel Up. Αυτό συμβαίνει όταν παραλαμβάνει τέσσερα VER προτού στείλει όλα τα οκτώ
- **SOFT\_ERROR**: ενεργοποιείται όταν ένα προσωρινό σφάλμα bit διαδραματίζεται. Τα σφάλματα bit εντοπίζονται χρησιμοποιώντας τα σήματα RXNOTINTABLE και RXDISPERR από τον MGT

- **HARD\_ERROR** : ενεργοποιείται όταν το κανάλι επανεκκινεί λόγω καταστροφικού σφάλματος όπως υπερχείλιση buffer, υποχείλιση buffer ή επαναλαμβανόμενα ήπια λάθη (soft error).
- **FRAME\_ERROR** : ενεργοποιείται όταν παραλαμβάνονται PDU καναλιού με κακό πλαίσιο. Τα κακά πλαίσια είναι συνήθως παρενέργεια των ήπιων σφαλμάτων (soft error).

#### Διεπαφή Διόρθωσης Ρολογιού της Σχεδίασης Αναφοράς του Aurora

Η διόρθωση ρολογιού (CC) εμποδίζει την υποχείλιση/υπερχείλιση του RX buffer στον απομακρυσμένο εταίρο καναλιού. Το πρωτόκολλο Aurora επιτρέπει 200 ppm διαφορά μεταξύ των ρολογιών των εταίρων καναλιού. Η σχεδίαση αναφοράς του Aurora έρχεται με ένα καθιερωμένο λειτουργικό τμήμα CC που δημιουργεί τα κλασικά 12 byte ανά 10000 bytes ως ακολουθία διόρθωσης ρολογιού. Το καθιερωμένο λειτουργικό τμήμα CC κανονικά οδηγεί τη διεπαφή διόρθωσης ρολογιού. Αντικαθίσταται με λογική χρήστη αν χρειάζεται αυτοσχέδιο πλάνο διόρθωσης ρολογιού. Οι θύρες διόρθωσης ρολογιού ορίζονται ως low εφόσον και οι δυο εταίροι καναλιού χρησιμοποιούν το ίδιο ρολόι. Η διεπαφή διόρθωσης ρολογιού έχει δύο θύρες

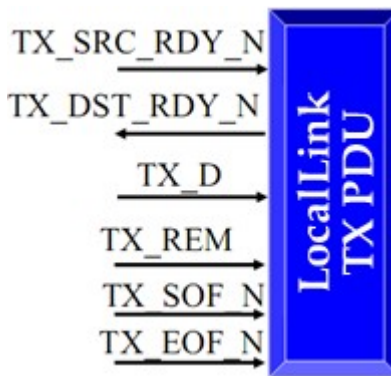
- **DO\_CC**: Η ακολουθία CC μεταδίδεται εφόσον είναι ενεργοποιημένο το DO\_CC
- **WARN\_CC**: Πρέπει να είναι ενεργοποιημένο για να εμποδίσει μηνύματα Ελέγχου Ροής Χρήστη (UFC messages) από το να ξεκινήσουν πολύ κοντά σε ακολουθία CC. Αυτό θα επεξηγηθεί με περισσότερη λεπτομέρεια στην ενότητα Ελέγχου Ροής

#### Τα Ρολόγια Αναφοράς των MGT της Σχεδίασης Αναφοράς του Aurora

Τα ρολόγια αναφοράς είναι υψηλής ποιότητας σήματα ρολογιών που χρησιμοποιούνται από τον MGT. Πολλαπλασιάζονται από τον MGT για να πάρουν το σωστό ρυθμό γραμμής. Πρέπει να είναι χαμηλού θορύβου και χρησιμοποιούνται τα BREFCLK μόνο για ρυθμούς ρολογιού αναφοράς >100 Mhz. Το Aurora μπορεί να χρησιμοποιήσει τις θύρες εισόδου BREFCLK ή REFCLK. Εάν η σχεδίαση χρησιμοποιεί MGT στην πάνω και κάτω άκρη του chip, χρειάζεται να επιβεβαιωθεί ότι το ίδιο φυσικό ρολόι τροφοδοτείται και στις δυο πλευρές για την αποτροπή προβλημάτων TX buffer

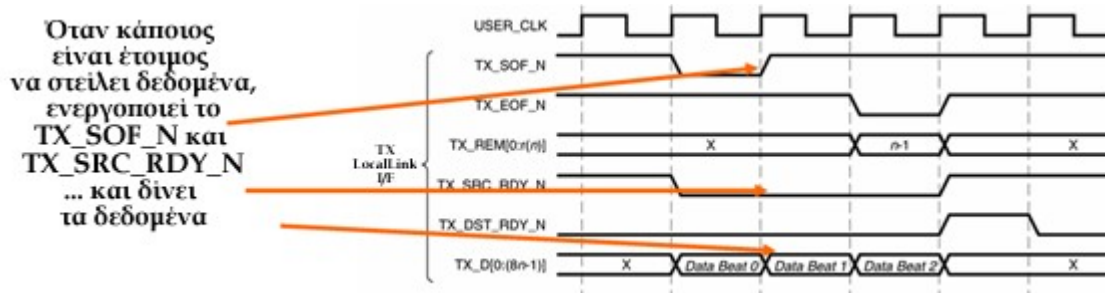
#### Αποστέλλοντας Δεδομένα

Για να αποσταλούν δεδομένα, χρησιμοποιείται η διεπαφή TX PDU του LocalLink



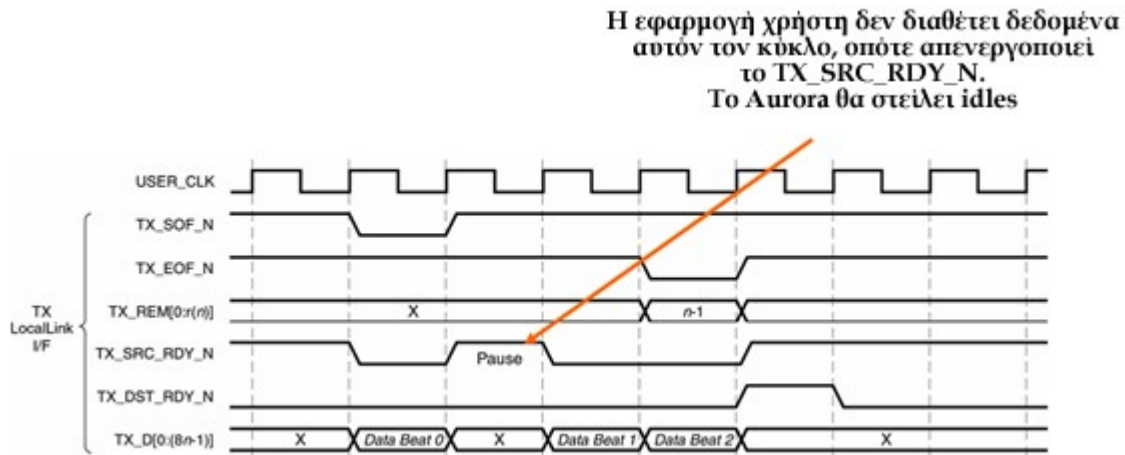
Σχήμα 2-31 : Διεπαφή TX (Αποστολής)

- Οδηγείται η πρώτη λέξη του PDU στην θύρα TX\_D, και ενεργοποιούνται τα σήματα TX\_SRC\_RDY\_N και TX\_SOF\_N
- Εάν το TX\_DST\_RDY\_N ενεργοποιείται στην ανοδική ακμή ρολογιού, τα δεδομένα δειγματοληπτήθηκαν από το λειτουργικό τμήμα Aurora και θα σταλούν στον εταίρο καναλιού
- Συνεχίζεται η τοποθέτηση δεδομένων στο TX\_D με ενεργοποιημένο το TX\_SRC\_RDY\_N. Όταν τα δεδομένα δεν είναι διαθέσιμα, απενεργοποιείται το TX\_SRC\_RDY\_N ώστε να σταλούν Idles αντ' αυτών
- Όταν το TX\_DST\_RDY\_N είναι απενεργοποιημένο, κρατούνται τα δεδομένα για να γίνει απόπειρα να σταλούν ξανά
- Στην τελευταία λέξη του πλαισίου, ενεργοποιείται το TX\_EOF\_N και το TX\_SRC\_RDY\_N. Επιβεβαιώνεται ότι το TX\_REM φέρει τον αριθμό των έγκυρων bytes στην τελευταία λέξη-1



Σχήμα 2-32 : Παράδειγμα απλής μεταφοράς δεδομένων

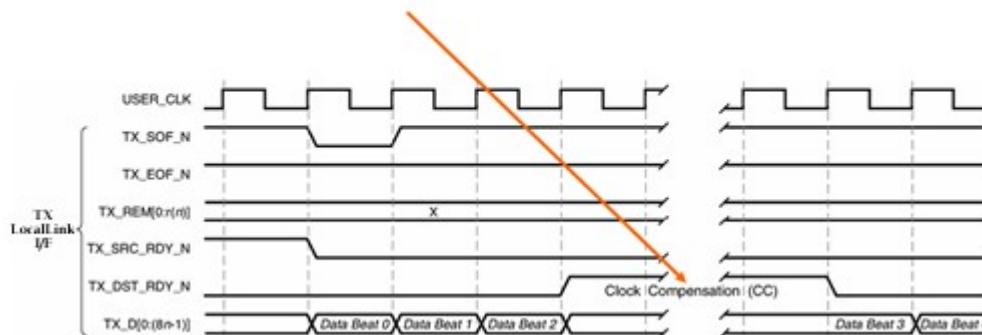
Στο παραπάνω Σχήμα 2-32 παρουσιάζεται η διεπαφή του LocalLink με λέξεις n-byte, τρεις κύκλοι 3n byte μεταφέρονται. Η απενεργοποίηση του SRC\_RDY\_N στη μέση ενός πλαισίου δεν προκαλεί κανένα πρόβλημα στη μετάδοση δεδομένων. Απλά θα προστεθούν, όπως και στο συγκεκριμένο παράδειγμα που παρατίθεται τέσσερις οκτάδες (DWORD) από IDLE μέσα στη μετάδοση όπως στο Σχήμα 2-33.



Σχήμα 2-33 : Παράδειγμα Παύσης Ροής Δεδομένων στη Μέση Πακέτου

Οι λόγοι που απενεργοποιείται το TX\_DST\_RDY\_N είναι ότι το λειτουργικό τμήμα Aurora απενεργοποιεί το TX\_DST\_RDY\_N γιατί **το κανάλι δεν αρχικοποιείται** δηλαδή δεν δίνεται TX\_DST\_RDY\_N μέχρι το κανάλι να είναι έτοιμο. Ακόμα **αποστέλονται ακολουθίες διόρθωσης ρολογιού** και θεωρείται ότι οι ακολουθίες διόρθωσης ρολογιού έχουν υψηλότερη προτεραιότητα από τα δεδομένα χρήστη. Ενδεχομένως να **αποστέλονται μηνύματα ελέγχου ροής** εφόσον και τα μηνύματα ελέγχου ροής επίσης έχουν υψηλότερη προτεραιότητα από τα δεδομένα. Μπορεί επίσης να βρίσκεται **εν ενεργεία τοπικός έλεγχος ροής** έτσι όταν ο απομακρυσμένος εταίρος καναλιού ζητά τοπικό έλεγχο ροής, το TX\_DST\_RDY\_N απενεργοποιείται για να σταματήσει τη ροή δεδομένων. Τέλος **είναι αναγκαίο, εύρος ζώνης για τα byte SCP και ECP**. Οι PDU καναλιού είναι πάντα τέσσερα bytes μεγαλύτερες από τις PDU εξαιτίας του SCP και ECP. Αυτό σημαίνει ότι το TX\_DST\_RDY\_N πρέπει μερικές φορές να απενεργοποιείται κατόπιν του τερματισμού ενός πλαισίου ώστε να δίδεται στο κανάλι Aurora χρόνος να αποστείλει ολόκληρη τη PDU καναλιού. Παρατίθεται παράδειγμα απενεργοποίησης TX\_DST\_RDY\_N για διόρθωση ρολογιού.

Το TX\_DST\_RDY\_N απενεργοποιείται  
για όσο χρειαστεί να μεταδοθεί η διόρθωση ρολογιού

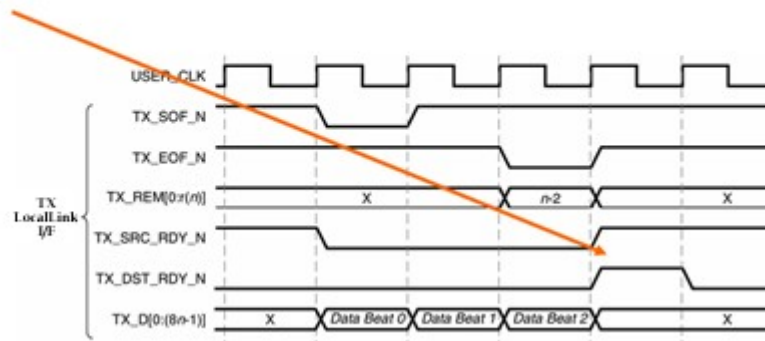


Σχήμα 2-34 : Παράδειγμα Παύσης Ροής Δεδομένων για Διόρθωση Ρολογιού



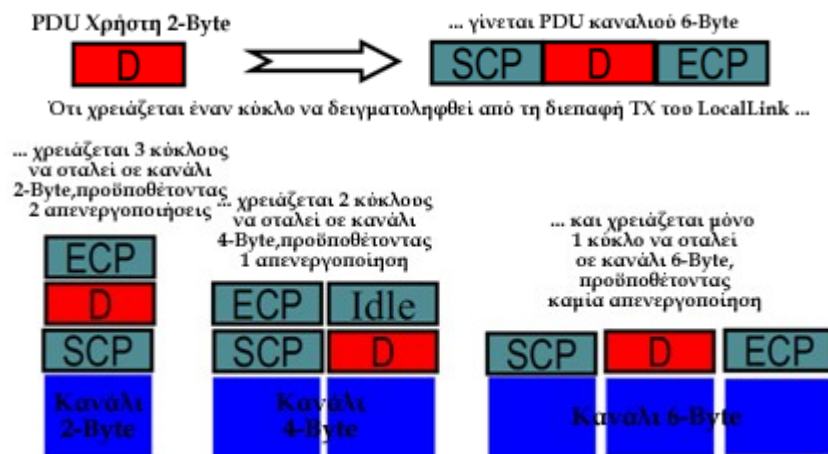
Παρατίθεται παράδειγμα απενεργοποίησης TX\_DST\_RDY\_N μετά το τέλος πλαισίου

Το TX\_DST\_RDY\_N απενεργοποιείται για ένα κύκλο μετά από τη λέξη του EOF του προηγούμενου πλαισίου



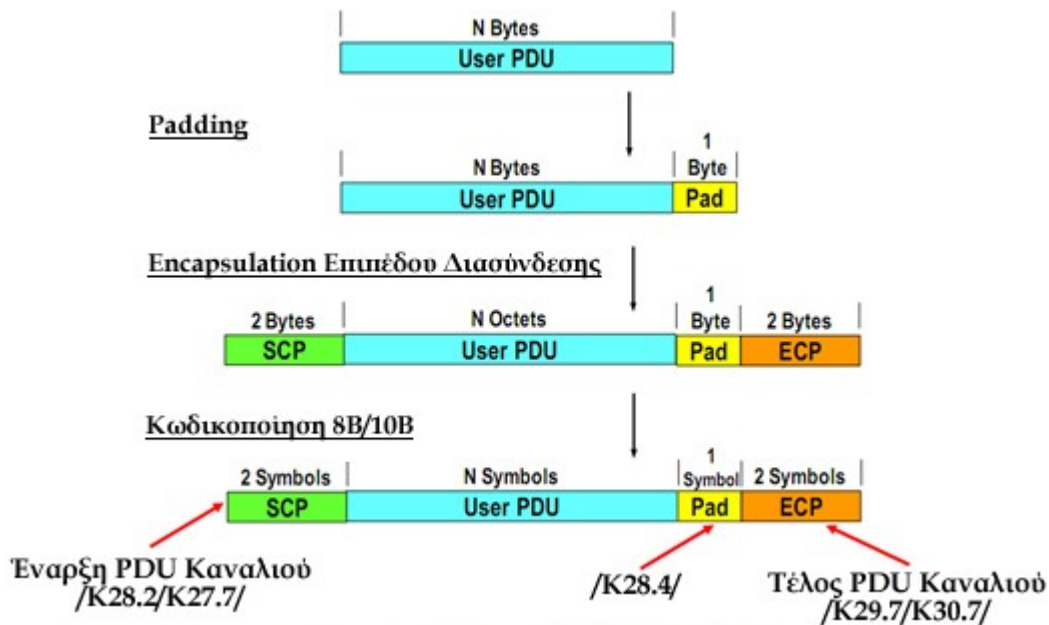
Σχήμα 2-35 : Παράδειγμα Παύσης Ροής Δεδομένων για Ολοκλήρωση Αποστολής Πακέτου

Σχηματικά αποδεικνύεται ο λόγος απενεργοποίησης του TX\_DST\_RDY\_N μετά από κάποιες PDU



Σχήμα 2-36 : Σχέση PDU Χρήστη και PDU Καναλιού

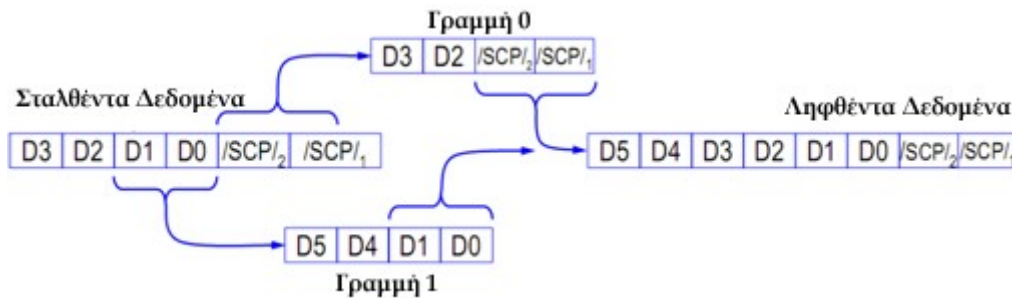
Εμφωλιασμός (Encapsulation) Δεδομένων στο Aurora



Σχήμα 2-37 : Encapsulation Δεδομένων στο Aurora

### Striping Δεδομένων στο Aurora

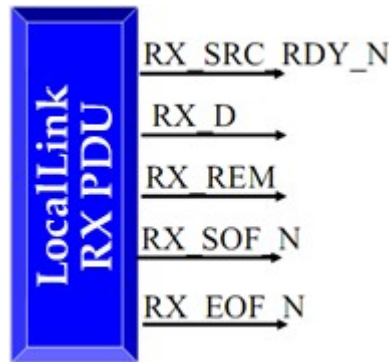
Όταν το κανάλι Aurora αποτελείται από πολλές γραμμές, το Aurora κατανέμει τις PDU στις πολλαπλές γραμμές όπως φαίνεται στο Σχήμα 2-38. Σημειώνεται ότι η striping εξασφαλίζει ότι εταίροι καναλιού ακόμα και με διαφορετικού πλάτους διεπαφές MGT μπορούν να επικοινωνήσουν μεταξύ τους.



Σχήμα 2-38 : Ξετύλιγμα Δεδομένων στο Aurora

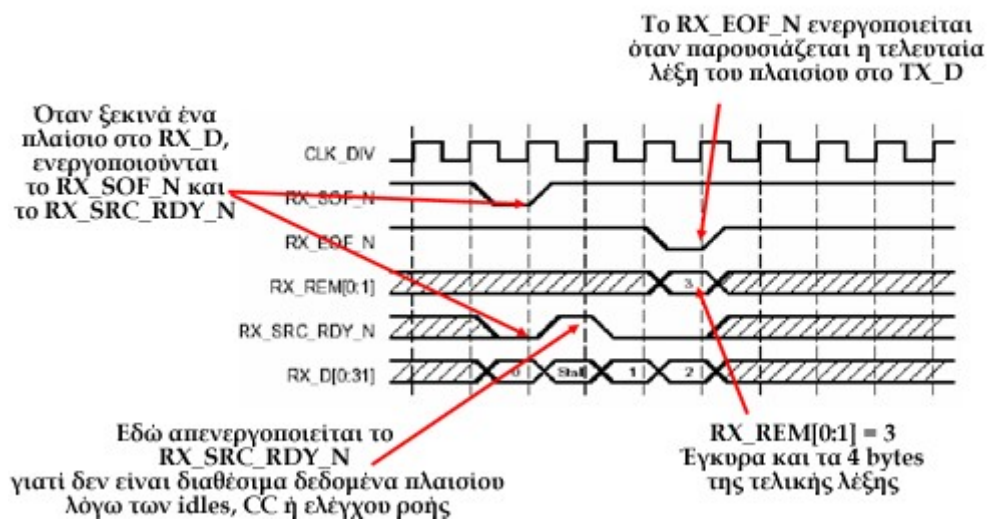
### Λήψη Δεδομένων

Όταν φτάνουν PDU καναλιού, παρουσιάζονται στην εφαρμογή του χρήστη μέσω της διεπαφής PDU του LocalLink RX. Σημειωτέον ότι η διεπαφή RX\_PDU δεν έχει ένα σήμα RX\_DST\_RDY\_N. Αυτό συμβαίνει γιατί το λειτουργικό τμήμα Aurora δεν έχει εσωτερικές FIFO. Το βάθος της FIFO και η συμπεριφορά του ελέγχου ροής εξαρτάται από την εφαρμογή, οπότε λόγω του κινδύνου της αύξησης της κατανάλωσης πόρων και της λανθάνουσας κατάστασης, η σχεδίαση αναφοράς απαιτεί η συμπεριφορά FIFO να υλοποιείται εξωτερικά. Οι FIFO είναι διαθέσιμες μέσω του CORE Generator

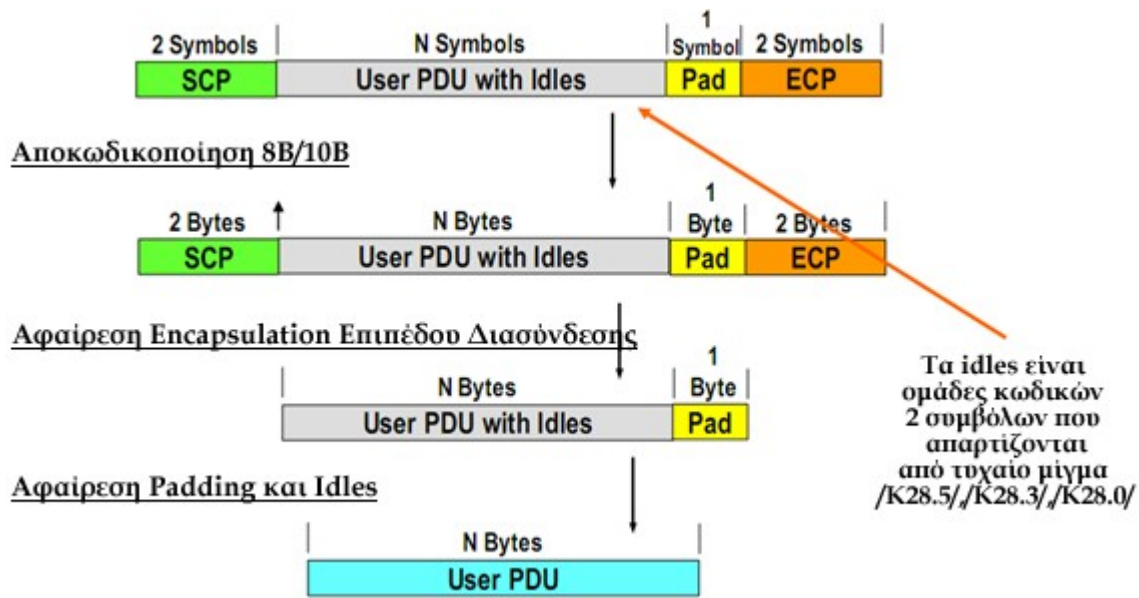


Σχήμα 2-39 : Διεπαφή RX (Λήψης)

Παρατηρούνται τα εισερχόμενα δεδομένα στη διεπαφή RX PDU στο LocalLink στο παρακάτω παράδειγμα:



Σχήμα 2-40 : Παράδειγμα Λήψης Δεδομένων



Σχήμα 2-41 : Decapsulation Δεδομένων

### Έλεγχος Ροής του Aurora

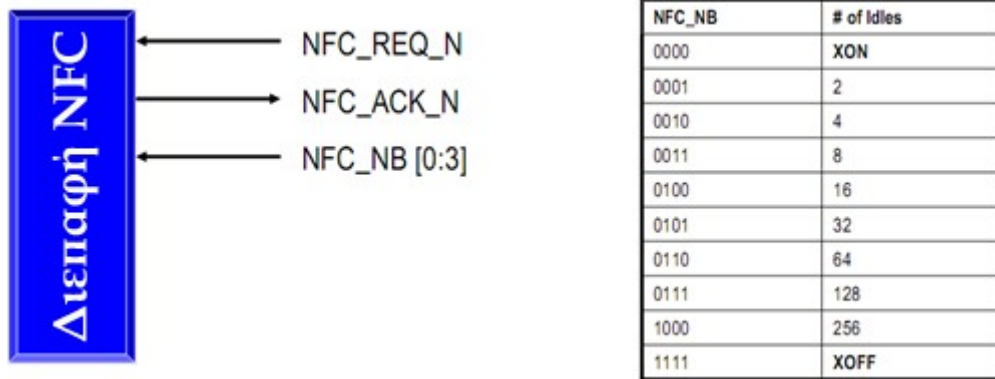
Το πρωτόκολλο Aurora προσδιορίζει δύο διαφορετικών τύπων ελέγχους ροής. Και οι δύο αυτοί υποστηρίζονται στη σχεδίαση αναφοράς του Aurora

- **Τοπικός Έλεγχος Ροής (NFC):** ο τοπικός έλεγχος ροής επιτρέπει σε έναν εταίρο καναλιού να ελέγξει το ρυθμό μετάδοσης των PDU καναλιού από τον απομακρυσμένο εταίρο καναλιού
- **Έλεγχος Ροής Χρήστη (UFC):** ο έλεγχος ροής χρήστη επιτρέπει την αποστολή συντόμων μηνυμάτων ελέγχου μέσω του καναλιού Aurora.
  - Ανόμοια με τις PDU του καναλιού, που μπορούν να διακοπουν και υπόκεινται σε έλεγχο ροής, τα μηνύματα UFC είναι αδιάκοπα και δεν σταματούν από έλεγχο ροής
  - Τα μηνύματα UFC μπορούν να διακόψουν PDU καναλιού επειδή έχουν υψηλότερη προτεραιότητα

### Τοπικός Έλεγχος Ροής

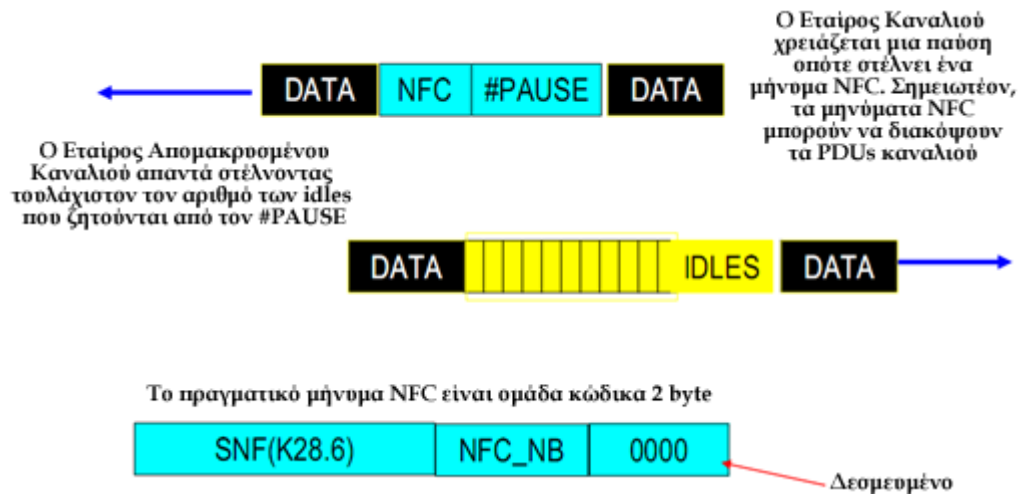
Ο Τοπικός Έλεγχος Ροής (NFC) δουλεύει ως ακολούθως. Ο εταίρος καναλιού αποστέλλει ένα σύντομο μήνυμα ελέγχου ώστε να ζητήσει NFC. Το μήνυμα φέρει ένα κωδικό που υποδεικνύει τον ελάχιστο αριθμό χαρακτήρων Idle που χρειάζονται. Ένας εταίρος καναλιού μπορεί να είναι σε **κατάσταση λειτουργίας Αμεσότητας (Immediate)** ή **κατάσταση λειτουργίας Ολοκλήρωσης (Completion)**. Εάν ληφθεί μήνυμα στην κατάσταση Αμεσότητας (Immediate), οι αιτήσεις NFC πρέπει να εκπληρωθούν αμέσως. Εάν βρίσκεται σε κατάσταση Ολοκλήρωσης (Completion), ο εταίρος καναλιού περιμένει για όποια PDU καναλιού σε εξέλιξη, να ολοκληρωθεί η μεταφορά τους πριν εκπληρωθεί το αίτημα. Ένας εταίρος καναλιού μπορεί επίσης να ζητήσει **XON** ή **XOFF**. XOFF συμβολίζει την παύση αποστολής PDU καναλιού μέχρι νεωτέρας. XON συμβολίζει την παύση εξυπηρέτησης όποιων εκκρεμών NFC και κανονική αποστολή PDU. Σημειωτέον ότι οι νεώτερες αιτήσεις NFC πανωγράφουν τις εκκρεμείς αιτήσεις, ακόμα και την αίτηση XOFF. Εάν ακολουθηθεί μια αίτηση XOFF από μια αίτηση για δυο idle, όταν ο απομακρυσμένος εταίρος καναλιού παραλάβει το δεύτερο μήνυμα θα τοποθετήσει

τουλάχιστον δυο idle στο κανάλι Aurora και μετά θα επιστρέψει στην κανονική λειτουργία.

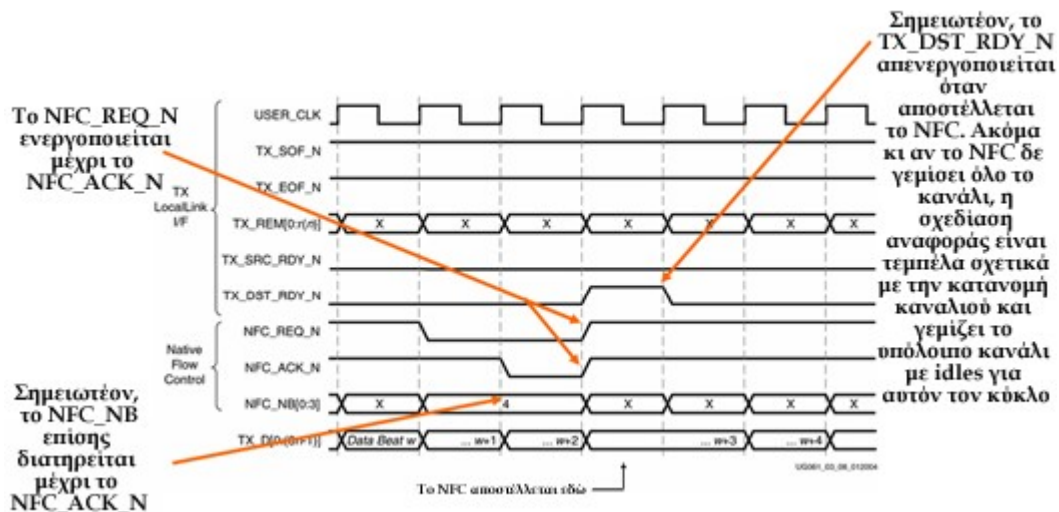


Σχήμα 2-42 : Απεικονίζεται η Διεπαφή NFC

Για την αποστολή ενός μηνύματος NFC, οδηγείται ένας κωδικός NFC στη θύρα NFC\_NB και ενεργοποιείται το NFC\_REQ\_N. Κρατείται το NFC\_REQ\_N μέχρι να φτάσει ακμή ρολογιού όπου το NFC\_ACK\_N είναι ενεργοποιημένο. Αυτό υποδεικνύει ότι το λειτουργικό τμήμα Aurora θα στείλει το μήνυμα. Παράδειγμα τοπικού ελέγχου ροής.



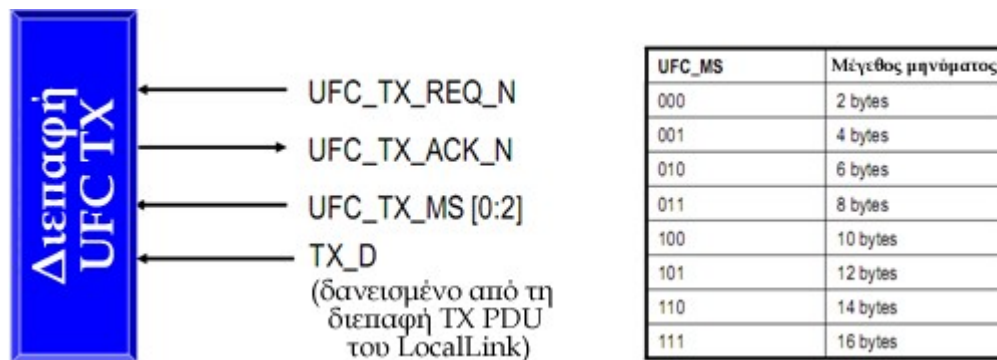
Σχήμα 2-43 : Παράδειγμα Αποστολής Μηνύματος NFC





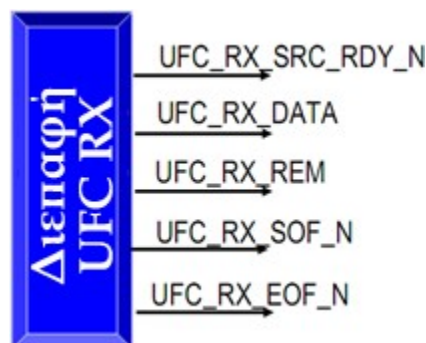
## Έλεγχος Ροής Χρήστη

Ο έλεγχος ροής χρήστη αποτελείται από σύντομα μηνύματα με μια επικεφαλίδα UFC. Έχουν τις ακόλουθες ιδιότητες. Έχουν άρτιο αριθμό bytes και είναι μεταξύ 2 και 16 bytes. Ακόμα δεν επηρεάζονται από τον τοπικό έλεγχο ροής. Και τέλος έχουν χαμηλότερη προτεραιότητα από μηνύματα NFC και ακολουθίες CC, αλλά μόλις αρχίσουν δεν γίνεται να διακοπούν από κανένα από τα δύο. Γι' αυτό η διεπαφή CC έχει τη θύρα WARN\_CC. Το WARN\_CC ενεργοποιείται πολύ πιο μπροστά από την CC τόσο ώστε να είναι εγγυημένο ότι πάντα τελειώνει το μήνυμα UFC πριν να αρχίσει η CC αν το WARN\_CC δεν είναι ενεργοποιημένο. Ενώ το κανάλι είναι ανενεργό, το WARN\_CC ενεργοποιείται, το DO\_CC ενεργοποιείται ή το NFC\_ACK\_N ενεργοποιείται, η σχεδίαση αναφοράς του Aurora δεν θα αρχίσει κανένα μήνυμα UFC. Τα μηνύματα UFC αποστέλλονται στον απομακρυσμένο εταίρο καναλιού όπου περνούν στην εφαρμογή του χρήστη μέσω μιας διεπαφής LocalLink. Το Aurora δεν επεξεργάζεται τα περιεχόμενα των μηνυμάτων UFC



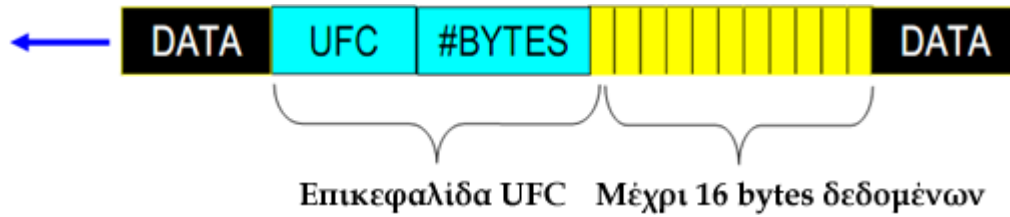
Σχήμα 2-45 : Απεικονίζεται η Διεπαφή UFC TX

Για την αποστολή ενός μηνύματος UFC, οδηγείται ένας κωδικός μεγέθους μηνύματος στη θύρα UFC\_MS και ενεργοποιείται το UFC\_TX\_REQ\_N. Κρατείται το UFC\_TX\_REQ\_N μέχρις ότου να φτάσει μια ακμή ρολογιού όπου το UFC\_TX\_ACK\_N είναι ενεργοποιημένο. Αυτό υποδεικνύει ότι το λειτουργικό τμήμα Aurora είναι έτοιμο για το μήνυμα. Στον κύκλο μετά το UFC\_TX\_ACK\_N, τοποθετούνται τα δεδομένα του μηνύματος στο TX\_D. Συνεχίζεται για όσους κύκλους χρειάζεται για την αποστολή ολόκληρου του μηνύματος. Το λειτουργικό τμήμα θα αγνοήσει το αχρησιμοποίητο κομμάτι των μερικών λέξεων



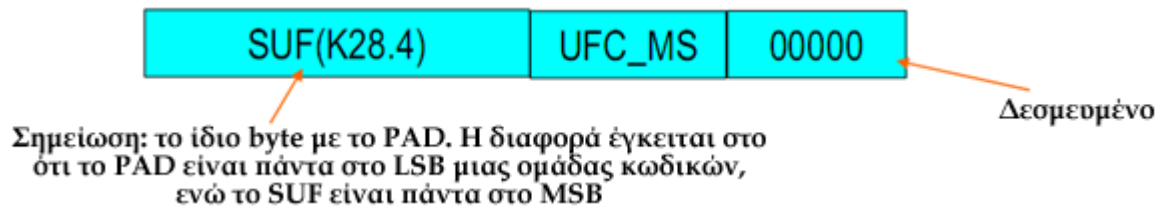
Σχήμα 2-46 : Απεικονίζεται η Διεπαφή UFC RX

Όταν φθάνουν μηνύματα UFC, παρουσιάζονται στην εφαρμογή χρήστη μέσω της διεπαφής RX UFC του LocalLink. Όπως η διεπαφή RX PDU του LocalLink, αυτή η διεπαφή δεν έχει σήμα UFC\_RX\_DST\_RDY\_N. Το μέγιστο πλάτος της θύρας UFC\_RX\_Data σε όποιο λειτουργικό τμήμα είναι 16 bytes πλάτος – δεν χρειάζεται να είναι πλατύτερο επειδή 16 bytes είναι το μεγαλύτερο δυνατό εισερχόμενο μήνυμα.

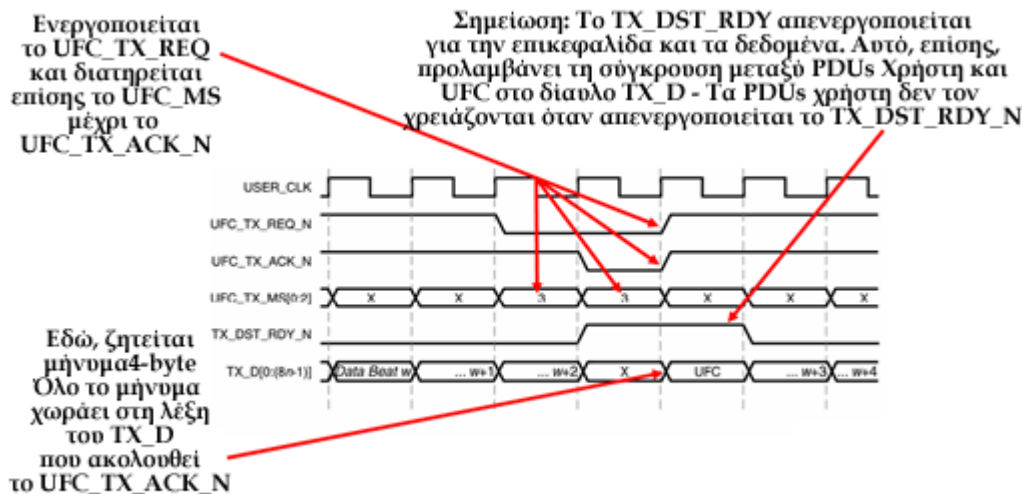


Σχήμα 2-47 : Παράδειγμα Ελέγχου Ροής Χρήστη

Όπως τα μηνύματα NFC, τα μηνύματα UFC μπορούν να διακόψουν ένα PDU καναλιού σε εξέλιξη. Το πραγματικό σύνολο κωδικών της επικεφαλίδας UFC φαίνεται παρακάτω:



Σχήμα 2-48 : Μήνυμα UFC εις Βάθος Ανάλυση



Σχήμα 2-49 : Παράδειγμα μετάδοσης ενός μηνύματος UFC:



## ΚΕΦΑΛΑΙΟ 3

### ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΥΛΟΠΟΙΗΣΕΙΣ

#### 3.1 Επισκόπηση

Οι «εργαστηριακές υλοποιήσεις» είναι ένα κεφάλαιο που αφορά τις λεπτομέρειες της σχεδίασης του υλικού της εργασίας. Έχει οργανωθεί σε δύο ενότητες όπου η μια περιλαμβάνει υλοποίηση της σειριακής μεταφοράς δεδομένων σε υψηλές ταχύτητες χωρίς την σύνδεση με κεντρική μονάδα επεξεργασίας και κάποιο δίαυλο. Η άλλη ενότητα αφορά την υλοποίηση σειριακής μεταφοράς δεδομένων σε υψηλές ταχύτητες με σύνδεση σε επεξεργαστή ο οποίος εκτελεί εφαρμογή χρήστη που ελέγχει την όλη διαδικασία. Ακόμα ο επεξεργαστής συνδέεται μέσω διαύλου με περιφερειακές συσκευές που εκτελούν από επιμέρους ενέργειες μέχρι το κυρίως έργο της μεταφοράς δεδομένων. Στην κάθε ενότητα περιγράφονται τα στάδια από τα οποία πέρασε η σχεδίαση και δίνεται έμφαση στις λεπτομέρειες της υλοποίησης που επιτελούν σημαντικές λειτουργίες όπως επίσης και σε αυτές που οι αρχές λειτουργίας τους δεν είναι προφανείς.

#### 3.2 Σχεδίαση χωρίς χρήση του PowerPC

##### 3.2.1 Το Bus Functional μοντέλο για πρωτόκολλο επιπέδου διασύνδεσης Aurora

Χρησιμοποιείται μοντελοποίηση σχεδίασης αναφοράς η οποία βασίζεται στα κυκλώματα του RocketIO με σκοπό την προσομοίωση της λειτουργίας και της προσαρμογής στις παραμέτρους της σχεδίασης χωρίς τη χρήση υλικού. Αυτή η υλοποίηση χρησιμοποιεί το μοντέλο Aurora Bus Functional Model (ABFM), που αναπαριστά ένα “μαύρο κουτί” που αποτελείται από συναρτήσεις και βιβλιοθήκες που χρησιμοποιούνται μόνο για σκοπούς προσομοίωσης

- Το ABFM μοντελοποιεί τη συμπεριφορά του πρωτοκόλλου Aurora και μπορεί να χρησιμοποιηθεί για την παραγωγή εναυσμάτων και την παρακολούθηση της απόκρισης σε αυτά, μιας σχεδίασης διεπαφής του Aurora που αναφέρεται ως συσκευή υπό δοκιμή (DUT). Το ABFM παρέχει παραμετροποίηση του πρωτοκόλλου (για παράδειγμα του αριθμού των γραμμών) και μπορεί να χρησιμοποιηθεί για τη δοκιμή όποιας υλοποίησης του πρωτοκόλλου Aurora με λίγη επιπλέον πληροφορία. Το ABFM παρέχει ευελιξία, μια καθαρή υλοποίηση του Aurora, και βελτιωμένη επίδοση έναντι της χρήσης άλλης σχεδίασης για επαλήθευση. Μόλις το ABFM ενσωματώνεται στο περιβάλλον επαλήθευσης, μπορεί να επικοινωνήσει με την DUT χρησιμοποιώντας τη διεπαφή γλώσσας προγραμματισμού (PLI) για περιβάλλοντα Verilog ή διεπαφή ξένης γλώσσας (FLI) για τα περιβάλλοντα VHDL+ModelSim. Και οι δύο, η PLI και FLI περιέχουν κλήσεις βασισμένες σε συνδιαλλαγές που χρησιμοποιούνται για να εγκαθιδρύσουν την επικοινωνία μεταξύ της DUT και του ABFM.
- Το αρχιτεκτονικό μοντέλο του Aurora (AAM) μοντελοποιεί τη συμπεριφορά του πρωτοκόλλου Aurora σε C. Ο στόχος του AAM είναι μόνο ο έλεγχος της υλοποίησης του Aurora, όχι η λειτουργία του SERDES στον πραγματικό κόσμο ή η υλοποίηση διεπαφής χρήστη στη DUT. Οι λανθάνουσες καταστάσεις (όποια

συμπεριφορά σχετική με χρονισμό) που βλέπει η DUT δεν μοντελοποιούνται στο AAM.

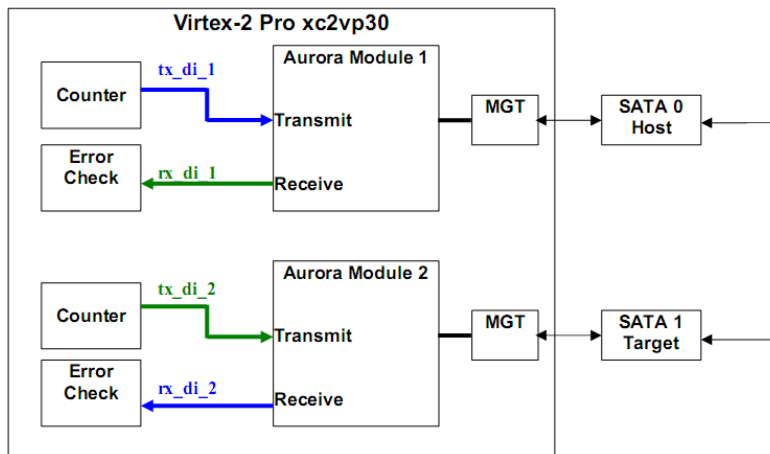
- Εκτελέστηκε προσαρμογή του μοντέλου αυτού στο λογισμικό και τις παραμέτρους του συστήματός μας και διενεργήθηκε προσομοίωση του κυκλώματος που περιλαμβάνει το Aurora με το Modelsim 6.0 SE

### 3.2.2 Υλοποίηση πρωτοκόλλου Aurora σε ένα XUP με loopback

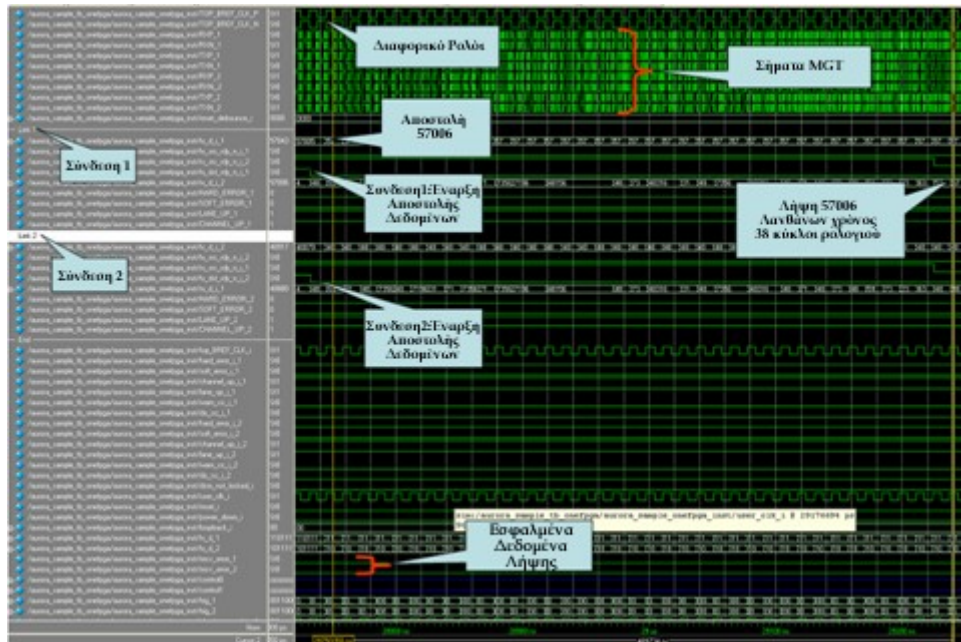
Γίνεται η πρώτη γνωριμία με τα κυκλώματα του RocketIO εκτελώντας τη σχεδίαση αναφοράς της Xilinx από το αρχείο aurora\_xupv2p.zip. Πρόκειται για μια υλοποίηση που χρησιμοποιεί το πρωτόκολλο επιπέδου διασύνδεσης Aurora κι άλλη λογική σε μια FPGA XUP Virtex™-II Pro μεταφέροντας δεδομένα σε μια αναδρομή προς τον εαυτό της. Εκτελούνται, λοιπόν, οι οδηγίες της αντίστοιχης τεκμηρίωσης της σχεδίασης αναφοράς ώστε να επαληθευτεί η σωστή λειτουργία της σειριακής μεταφοράς δεδομένων.

#### Σύνοψη Σχεδίασης

Η σχεδίαση δοκιμών αρχικοποιεί τον παραγόμενο πυρήνα Aurora δύο φορές. Το ένα στιγμιότυπο έχει διεπαφή με τον SATA0 οικοδεσπότη και το άλλο με τον SATA 1 στόχο μέσω MGT, που επιτρέπει την εκτέλεση δοκιμής με βρόχο χρησιμοποιώντας ένα καλώδιο Serial ATA. Από πλευράς FPGA, υπάρχει διεπαφή με ένα μετρητή που τροφοδοτεί τη μετάδοση και με ένα λειτουργικό τμήμα που ελέγχει για σφάλματα στην λήψη. Το Σχήμα 3-1 απεικονίζει πολύ περιγραφικά τη σχεδίαση αναφοράς.



Σχήμα 3-1 : Λειτουργικό Διάγραμμα της Σχεδίασης



Σχήμα 3-2 : Κυματομορφές Modelsim – Μια Προσομοίωση της FPGA

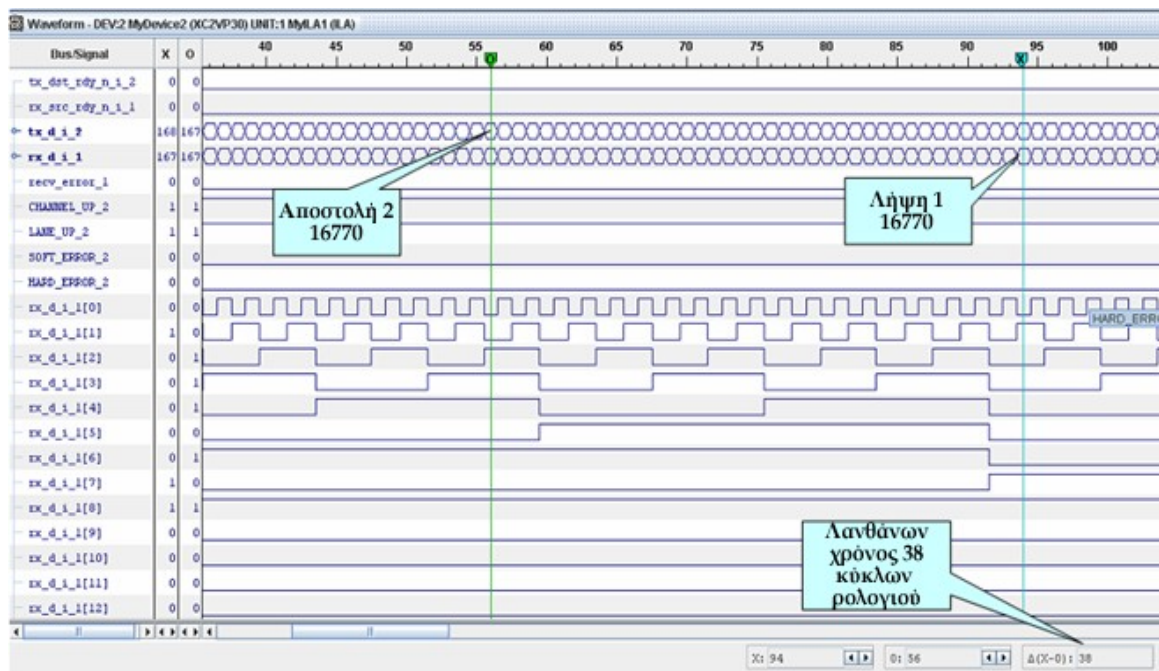
### Υλοποίηση σχεδίασης

Ακολουθούνται τα εξής βήματα για την υλοποίηση της σχεδίασης αναφοράς:

- Χρησιμοποιείται ο πυρήνας Aurora που διατίθεται ελεύθερα από το εργαλείο Core Generator της Xilinx. Επίσης υπάρχει και σχεδίαση αναφοράς, όπως ονομάζεται, από τη Xilinx που χρησιμοποιείται ως βάση για τις εργαστηριακές δοκιμές.
- Εκτελέστηκε προσαρμογή της σχεδίασης αναφοράς στα νέα εργαλεία της Xilinx και αντιμετωπίστηκαν επιπλοκές που παρουσιάστηκαν από τη μετατροπή.
- Εκτελέστηκε παραμετροποίηση του πυρήνα Aurora. Πιο συγκεκριμένα εκτελέστηκαν δοκιμές με διαφορετικές παραμέτρους κάθε φορά ώστε να καλυφθεί όλο το φάσμα λειτουργίας της σχεδίασης στην πλακέτα XUP κι έτσι επαληθεύτηκε η λειτουργία μετάδοσης δεδομένων από τον MGT(Host) της πλακέτας σε άλλο MGT(target) της ίδιας πλακέτας και αντίστροφα μέσω καλωδίου SATA.
- Χρήση Modelsim 6.2c SE για προσομοίωση της λειτουργίας. Εκτελούνται τροποποιήσεις σε script για compile των απαραίτητων αρχείων και για εκκίνηση της προσομοίωσης
- Η σχεδίαση «κατέβηκε» με επιτυχία σε μια XUP Virtex™-II Pro FPGA πλακέτα εκτελώντας ένα βρόχο.
- Επιβεβαίωση λειτουργίας με χρήση ChipScope
- Έγινε επαλήθευση της ορθής λειτουργίας των αντίστοιχων κυκλωμάτων του RocketIO της πλακέτας όπως επίσης και της σωστής λειτουργίας της σχεδίασης. Ακόμα μελετήθηκε η συμπεριφορά του κυκλώματος σε πραγματικές συνθήκες «στον πάγκο» κι εκτελέστηκαν δοκιμές για διάφορες τιμές των παραμέτρων καθώς και σε οριακές καταστάσεις.

### Συμπεράσματα

Ολοκληρώθηκαν τα κυριότερα στάδια παραγωγής του πυρήνα του Aurora με χρήση του Core Generator αρχικοποιώντας τον παραγόμενο πυρήνα σε μια απλή σχεδίαση. Εκτελέστηκε προσομοίωση HDL της σχεδίασης χρησιμοποιώντας Modelsim. Στη συνέχεια, παρήχθησαν οι πυρήνες ICON και ILA για το Chipscope Pro και εισήχθησαν μες στη σχεδίαση. Τελικώς, κλήθηκε ο Chipscope Pro Analyzer να επιτελέσει μια επαλήθευση της ορθότητας των σημάτων και της σωστής λειτουργίας της σχεδίασης πάνω στο chip.



Σχήμα 3-3 : Κυματομορφές Chipscope Pro

### 3.2.3 Υλοποίηση πρωτοκόλλου Aurora σε δύο XUP

Πρόκειται για επέκταση της προηγούμενης υλοποίησης ώστε να εκτελεστεί σειριακή επικοινωνία μεταξύ δύο πλακετών που διαθέτουν RocketIO. Αυτή η υλοποίηση χρησιμοποιεί το πρωτόκολλο επιπέδου διασύνδεσης Aurora κι άλλη λογική σε δυο FPGA XUP Virtex™-II Pro μεταφέροντας δεδομένα από την μια πλακέτα στην άλλη.

- Χρησιμοποιείται ο πυρήνας Aurora που διατίθεται ελεύθερα από το εργαλείο Core Generator της Xilinx. Επίσης υπάρχει και σχεδίαση αναφοράς, όπως ονομάζεται, από τη Xilinx που χρησιμοποιείται ως βάση για τις εργαστηριακές δοκιμές.
- Εκτελέστηκε προσαρμογή της προϋπάρχουσας σχεδίασης αναφοράς για μια πλακέτα ώστε να λειτουργήσει η σχεδίαση σε δυο πλακέτες.
- Εκτελέστηκε από το λογισμικό polling του περιφερειακού ώστε να ανιχνευθεί έγκαιρα η λήψη των δεδομένων που έχουν ήδη σταλεί από το περιφερειακό της άλλης πλακέτας.

- ο Έγινε επαλήθευση της ορθής λειτουργίας των αντίστοιχων κυκλωμάτων του RocketIO και των δύο πλακετών όπως επίσης και της σωστής λειτουργίας της σχεδίασης. Το πιο σημαντικό, μελετήθηκε η επίδραση που έχει στη σειριακή μεταφορά δεδομένων σε υψηλές ταχύτητες η σηματοδοσία ρολογιού στα δύο μέλη της μεταφοράς, από ανεξάρτητα μεταξύ του σήματα ρολογιών. Επιβεβαιώθηκε, λοιπόν, ότι κάτι τέτοιο δεν προκαλεί πρόβλημα στην επικοινωνία των δύο πλακετών. Ακόμα μελετήθηκε η συμπεριφορά του κυκλώματος σε πραγματικές συνθήκες «στον πάγκο» κι εκτελέστηκαν δοκιμές για διάφορες τιμές των παραμέτρων καθώς και σε οριακές καταστάσεις.

### 3.3 Σχεδίαση με χρήση του PowerPC

#### 3.3.1 Λίγα λόγια για το Bit Error Rate Tester

Πρώτα από όλα επιχειρείται να χρησιμοποιηθεί ο Bit Error Rate Tester που αποτελεί σχεδίαση αναφοράς για δοκιμή του RocketIO. Αποδεικνύεται ότι η σχεδίαση είναι πολύ πολύπλοκη για τα πρώτα βήματα ενός αρχάριου χρήστη στην εξοικείωση με τα κυκλώματα του RocketIO. Ο βαθμός πολυπλοκότητας αυξάνεται λόγω της ασυμβατότητας των εκδόσεων του λογισμικού με βάση το οποίο είχε δημιουργηθεί η σχεδίαση καθώς και την υλοποίησή της για πλακέτα διαφορετική από τη διαθέσιμη. Επιπλέον, το πολύ μεγάλο πλήθος αρχείων καθιστούν ακόμα πιο δύσκολη τη μελέτη της σχεδίασης σε χαμηλό επίπεδο (επίπεδο κώδικα VHDL). Ωστόσο, αυτή η πρώτη επαφή με μια σχεδίαση αναφοράς είχε ως αποτέλεσμα τη μελέτη κι εκμάθηση βασικών όρων, καθώς και προϋποθέσεων και περιορισμών για τη δημιουργία της τελικής σχεδίασης της εργασίας, όπως αναφέρονται στη συνέχεια.

#### Περίληπτικά Πομποδέκτης του RocketIO στο Bit-Error Rate Tester

Περιγράφεται η υλοποίηση μιας σχεδίασης αναφοράς bit-error rate tester (BERT) ενός RocketIO™ πομποδέκτη που επιδεικνύει μια σειριακή σύνδεση (1.0 Gb/s ως 3.125 Gb/s) μεταξύ δυο RocketIO multi-gigabit πομποδεκτών (MGT) ενσωματωμένων σε μια απλή Virtex-II Pro™ FPGA. Για να οικοδομηθεί το σύστημα, μια IBM CoreConnect™ δομή συνδέει τον επεξεργαστή PowerPC™ 405 (PPC405) με εξωτερική μνήμη και άλλα περιφερειακά χρησιμοποιώντας τον τοπικό δίαυλο του επεξεργαστή (PLB). Χρησιμοποιείται η σχεδίαση αναφοράς στο αρχείο με όνομα, XAPP661.zip, και το Embedded Development Kit (EDK) για την κατασκευή ενός ευπροσάρμοστου κι εκτεταμένου συστήματος PPC405. Ένα λειτουργικό τμήμα bit-error rate tester (XBERT) δυο-καναλιών της Xilinx χρησιμοποιείται για την παραγωγή και επαλήθευση των σειριακών δεδομένων υψηλών-ταχυτήτων που αποστέλλονται και παραλαμβάνονται από τους πομποδέκτες RocketIO. Η παραγωγή δεδομένων προς αποστολή πραγματοποιείται χρησιμοποιώντας ψευδο-τυχαίες ακολουθίες (PRBS) δυαδικών παραστάσεων. Η επαλήθευση ορθότητας εισερχομένων δεδομένων εκτελείται στο δέκτη από το λειτουργικό τμήμα XBERT που συγκρίνει τα εισερχόμενα δεδομένα με τα αναμενόμενα δεδομένα για λάθη. Η XBERT υποστηρίζει αρκετούς διαφορετικούς τύπους PRBS και μοτίβων ρολογιού με δυνατότητα επιλογής από το χρήστη. Οι μετρητές πλαισίων στο δέκτη χρησιμοποιούνται για να εντοπίσουν το συνολικό αριθμό λέξεων δεδομένων στα



πλαίσια που έχουν ληφθεί, το συνολικό αριθμό λέξεων δεδομένων με σφάλματα bit, και το συνολικό αριθμό σφαλμάτων bit. Ο επεξεργαστής διαβάζει τις τιμές κατάστασης και τιμές μετρητή από το XBERT μέσω της διεπαφής του με τον PLB, κι έπειτα στέλνει τις πληροφορίες στο UART. Η σχεδίαση αναφοράς επίσης υποστηρίζει εντός του κυκλώματος μερική αναδιαμόρφωση των χαρακτηριστικών του πομποδέκτη RocketIO χρησιμοποιώντας την εσωτερική θύρα πρόσβασης διαμόρφωσης της Virtex-II Pro (ICAP). Αυτή η λύση χρησιμοποιείται για την εκτέλεση μερικής αναδιαμόρφωσης των χαρακτηριστικών ελέγχου της προ-έμφασης και της διαφορικής ταλάντωσης του πομποδέκτη RocketIO.

### Επισκόπηση Υλοποίησης Υλικού

Η σχεδίαση αναφοράς BERT του πομποδέκτη του RocketIO επιδεικνύει ένα σύστημα PPC405 και συσκευές PLB. Ο PLB αποτελείται από τέσσερις συσκευές σκλάβους. Ένας ελεγκτής block RAM (BRAM) συνδέει 64 KBytes BRAM στο δίαυλο χρησιμεύοντας ως μνήμη δεδομένων του επεξεργαστή. Επίσης συνδεδεμένο στο δίαυλο είναι το λειτουργικό τμήμα UART, ένα λειτουργικό τμήμα XBERT δύο-καναλιών, και ένα λειτουργικό τμήμα ICAP το καθένα από τα οποία ενώνεται μέσω μιας Intellectual Property Interface (IPIF) της Xilinx, που παρέχει τυποποιημένες συνδέσεις στο δίαυλο. Ο επεξεργαστής έχει δύο συνδέσεις αφέντη στο δίαυλο PLB, μια για cache μνήμη εντολών και μια για την cache δεδομένων.

### 3.3.2 Υλοποίηση πρωτοκόλλου Aurora με χρήση επεξεργαστή σε ένα XUP

#### Προδιαγραφές σχεδίασης

Αποτελεί την ολοκληρωμένη σχεδίαση που υλοποιείται στην εργασία αυτή. Σε πρώτη φάση είχαν εκτελεστεί δοκιμές με σχεδίαση αναφοράς που περιλάμβανε επεξεργαστή, δίαυλο και περιφερειακά. Κάτι τέτοιο βοήθησε στην εξέλιξη της τελικής σχεδίασης που διαμορφώθηκε όμως κρίθηκε ότι δεν αξίζει περαιτέρω ανάλυσης γιατί οι κύριες μέθοδοι λειτουργίας καλύπτονται από την περιγραφή της τελικής σχεδίασης. Η τελική σχεδίαση της εργασίας αποτελεί υλοποίηση λογικής για την κατασκευή περιφερειακού, που χρησιμοποιεί το πρωτόκολλο επιπέδου διασύνδεσης Aurora, πάνω στον δίαυλο PLB συνδεδεμένο με τον επεξεργαστή PowerPC σε μια XUP Virtex™-II Pro FPGA μεταφέροντας δεδομένα σε βρόχο με τον εαυτό του. Αυτή η σχεδίαση είναι πιο ευέλικτη γιατί επιτρέπει επικοινωνία με ποικιλία εφαρμογών χρήστη που τρέχουν στον PowerPC.

Υπάρχει μια σχεδίαση αναφοράς από την Xilinx με όνομα `aurora_804_v2_ml321_demo.zip`. Δε χρησιμοποιήθηκε ωστόσο ως βάση για τις εργαστηριακές δοκιμές λόγω της αναντιστοιχίας της με τα νέα εργαλεία της Xilinx και της μεγάλης πολυπλοκότητας στην προσαρμογή της σε αυτά. Επίσης πολύ σημαντική παράμετρος για τη μη χρησιμοποίησή της ήταν η αναφορά της σχεδίασης σε άλλη πλακέτα σε σχέση με την XUP.

- Χρησιμοποιείται ο πυρήνας Aurora που διατίθεται ελεύθερα από το εργαλείο Core Generator της Xilinx. Επίσης χρησιμοποιήθηκαν κι έτοιμες FIFO από τον Core Generator της Xilinx.
- Δημιουργήθηκε custom περιφερειακό για τη σηματοδότηση του ρολογιού στη σχεδίαση και στους MGTs.

- Δημιουργήθηκε custom περιφερειακό που περιλαμβάνει το Aurora καθώς κι επιπλέον λογική για τη διασύνδεσή του με τις RX και TX FIFOs όπως επίσης και με τη διεπαφή του PLB.
- Παραμετροποιήθηκε το αρχείο .ucf που φέρει τους περιορισμούς της σχεδίασης. Πιο συγκεκριμένα περιλαμβάνει πληροφορίες για την τοποθεσία των ακροδεκτών (pins) σημάτων ζωτικής σημασίας, όπως των διαφορικών εισόδων του ρολογιού καθώς και την τοποθεσία των MGT που χρησιμοποιούνται στη μετάδοση.
- Δημιουργήθηκε εφαρμογή που εκτελείται στον PowerPC που εκκινεί κι ελέγχει όλη τη διαδικασία της μεταφοράς δεδομένων.
- Έχει δημιουργηθεί το κατάλληλο API στην εφαρμογή αυτή για την αποστολή και λήψη δεδομένων.
- Υπάρχει δυνατότητα εγγραφής δεδομένων σε FIFOs έτοιμες για αποστολή μέσω Direct Memory Access για βελτιστοποίηση της ταχύτητας. Επίσης υπάρχει δυνατότητα απλής εγγραφής μέσω του PLB διαύλου.
- Εκτελέστηκε με επιτυχία downloading της σχεδίασης στην πλακέτα XUP και αντιμετωπίστηκαν σφάλματα στη λειτουργία της σχεδίασης. Επαληθεύτηκε ότι τα δεδομένα αποστέλλονται από MGT (Host) της πλακέτας σε άλλο MGT (target) της πλακέτας και αντίστροφα μέσω καλωδίου SATA.
- Επιβεβαιώνεται η λειτουργία μέσω χρήσης Terminal και προβολή των κατάλληλων ενδείξεων

#### Πλεονεκτήματα – Μειονεκτήματα

Συνδέοντας το περιφερειακό Aurora στο PLB, ελαχιστοποιείται το μειονέκτημα σύνδεσης σε δίαυλο OPB που περιορίζει τον πραγματικό ρυθμό μεταφοράς των δεδομένων. Ο δίαυλος PLB προσφέρεται για συνδέσεις υψηλών επιδόσεων καθώς υποστηρίζει υψηλό ρυθμό λειτουργίας και διαθέτει πλάτος 64 bits. Επιπλέον πλεονέκτημα, παρέχει η χρήση μεταφορών DMA, γιατί ο PowerPC είναι ελεύθερος να κάνει άλλες εργασίες ενώ τα δεδομένα μεταφέρονται αυτόματα από και προς το περιφερειακό Aurora.

Το κυρίως μειονέκτημα της εισαγωγής επεξεργαστή στη σχεδίαση είναι η καθυστέρηση που εισάγει η κλήση μιας εντολής εφαρμογής στο σύστημα. Αυτό συμβαίνει γιατί συναρτήσεις όπως printf έχουν μεγάλο κόστος χρόνου μέχρι να εκτελεστούν στο σύστημα.

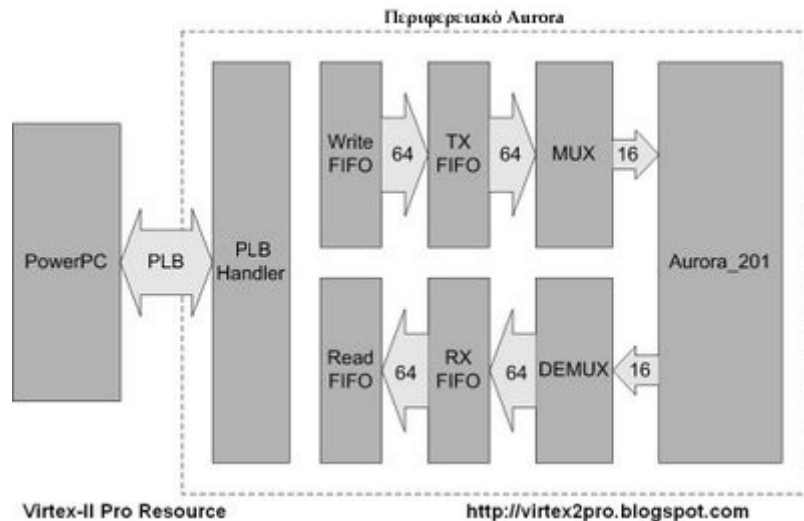
#### Συνοπτικά Πομποδέκτης Aurora για το PLB

Δημιουργείται ένα περιφερειακό που περιλαμβάνει τον πυρήνα Aurora για να υλοποιηθεί ένας σειριακός πομποδέκτης υψηλών ταχυτήτων που αποτελείται από έναν MGT του RocketIO. Πρόκειται για βελτιωμένη έκδοση πομποδέκτη Aurora ο οποίος αναπτύχθηκε με τη βοήθεια του άρθρου με τίτλο “Create an Aurora Transceiver” στην ιστοσελίδα “<http://virtex2pro.blogspot.com/2008/03/create-aurora-transceiver.html>”. Η βελτίωση επιτυγχάνεται συνδέοντας το περιφερειακό στο PLB αντί του OPB, που επιτρέπει πολύ πιο γρήγορη μεταφορά δεδομένων με την αξιοποίηση του ταχύτερου διαύλου και της ικανότητας Direct Memory Access.

Για να δοκιμαστεί η σχεδίαση, θα αρχικοποιηθούν δυο περιφερειακά Aurora, θα τους ανατεθούν διαφορετικοί MGTs του RocketIO και θα τοποθετηθεί ένα καλώδιο



SATA μεταξύ τους ώστε να δημιουργηθεί σύνδεση σε βρόχο. Ο PowerPC θα μεταφέρει δεδομένα μεταξύ των δυο περιφερειακών Aurora απλώς εκτελώντας ανάγνωση και εγγραφή στις FIFO ανάγνωσης κι εγγραφής. Για να δοκιμαστεί η σχεδίαση, η εφαρμογή δοκιμής θα εγγράψει δεδομένα στη FIFO εγγραφής του πρώτου περιφερειακού Aurora και θα τα διαβάσει από τη FIFO ανάγνωσης του δεύτερου περιφερειακού Aurora. Για να εκμεταλλευθεί το πλήρες εύρος ζώνης του PLB, ο PowerPC θα αναγνώσει και θα εγγράψει στις FIFO χρησιμοποιώντας μεταφορά DMA. Το περιφερειακό Aurora θα έχει επίσης εξόδους που θα υποδεικνύουν την κατάσταση της γραμμής μετάδοσης. Αυτοί οι έξοδοι θα είναι συνδεδεμένοι σε φωτοδιόδους (LEDs) στην πλακέτα XUPV2P.



Σχήμα 3-4 : Το περιφερειακό Aurora

Το παραπάνω σχήμα 3-4 απεικονίζει τη σχεδίαση του περιφερειακού Aurora. Χρησιμοποιεί FIFO ανάγνωσης κι εγγραφής για τη μεταφορά δεδομένων μεταξύ του PowerPC και της λογικής χρήστη. Επίσης χρησιμοποιεί ένα δεύτερο ζευγάρι FIFO που καλούνται RX και TX FIFO. Οι RX και TX FIFOs δημιουργούν μια buffered διασύνδεση μεταξύ των FIFO ανάγνωσης/εγγραφής και του πυρήνα Aurora. Αυτή η buffered διασύνδεση είναι απαραίτητη γιατί το ρολόι του PLB και του RocketIO είναι ανεξάρτητα στην πλακέτα XUPV2P. Το PLB χρονίζει τα δεδομένα προς τα μέσα και προς τα έξω από τις FIFO ανάγνωσης και εγγραφής στα 100MHz ενώ η λογική χρήστη χρονίζει τα δεδομένα προς τα μέσα και προς τα έξω από τον πυρήνα Aurora στα 75MHz. Για αυτό το λόγο, χρησιμοποιούμε RX και TX FIFO με ανεξάρτητες εισόδους ρολογιού ανάγνωσης και εγγραφής.

Οι TX και RX FIFO συνδέονται στον πυρήνα Aurora μέσω ενός πολυπλέκτη και αποπολυπλέκτη αντίστοιχα. Οι FIFO έχουν πλάτος 64 bits για να χρησιμοποιήσουν ολόκληρο το πλάτος δεδομένων του PLB. Θα παραχθεί πυρήνας Aurora με διεπαφή πλάτους 16 bit επειδή οι MGTs είναι βελτιστοποιημένοι για πλάτος 2 λέξεων, που είναι 16 bits αν χρησιμοποιείται κωδικοποίηση 8B/10B. Οι MUX και DEMUX χρειάζονται για να συνδέσουν την 64 bit διεπαφή FIFO στη 16 bit διεπαφή του πυρήνα Aurora. Ο MUX μετατρέπει κάθε εγγραφή 64 bit από την TX FIFO σε ομάδες των 16 bits διαμοιρασμένες σε 4 κύκλους ρολογιού. Ο DEMUX αποθηκεύει προσωρινά λέξεις των 4 x 16 bit από τον πυρήνα Aurora σε μια εγγραφή 64 bit στην RX FIFO. Αυτή η διαμόρφωση επιτρέπει την πλήρη χρήση του εύρους ζώνης του PLB.

Θα δημιουργηθεί ένα περιφερειακό Ψηφιακού Διαχειριστή Ρολογιού (DCM) για τους MGTs του RocketIO. Το περιφερειακό DCM θα περιέχει μόνο δυο buffer ρολογιού και δε θα χρειάζεται καταχωρητές ή FIFOs.

### Επεξεργαστής PowerPC

Αποτελεί το «μυαλό» της σχεδίασης και χρησιμοποιείται για την εκτέλεση της εφαρμογής της σχεδίασης καθώς και τη διαχείριση των διαύλων PLB ή OPB. Η συχνότητα λειτουργίας του στη σχεδίαση αυτή είναι 100 MHz. Η εφαρμογή που εκτελεί έχει γραφτεί σε γλώσσα προγραμματισμού C και έχει μεταγλωττιστεί με gcc compiler. Ακόμα διαθέτει 5 επίπεδα βελτιστοποίησης μεταγλώττισης (optimization levels), τα O-1, O-2, O-3, O-s κι επίπεδο χωρίς βελτιστοποίηση, για δημιουργία πιο αποδοτικού εκτελέσιμου αλλά με κόστος μεγαλύτερη διάρκεια μεταγλώττισης. Το εκτελέσιμο αρχείο που παράγεται έχει τη μορφή .elf. Η εφαρμογή, τέλος, μπορεί να φορτωθεί και να εκτελεστεί κατά την εκκίνηση της σχεδίασης.

### Εξωτερικοί ακροδέκτες (pins) σχεδίασης

Διακρίνονται ορισμένοι ακροδέκτες (pins) της σχεδίασης που θεωρούνται εξωτερικοί στη σχεδίαση. Έτσι παρατηρούνται τα sys\_clk\_pin και sys\_rst\_pin που χρησιμεύουν για τη σύνδεση με την πηγή ρολογιού καθώς και με το σήμα εκκίνησης κι επανεκκίνησης της πλακέτας. Αξίζει να σημειωθεί η ύπαρξη των σημάτων TOP\_BREF\_CLK\_P\_pin και TOP\_BREF\_CLK\_N\_pin που αποτελούν διαφορεικά σήματα ρολογιού. Συνδέονται μέσω του αρχείου ucf στην κατάλληλη τοποθεσία για την αποκλειστική δρομολόγηση σήματος ρολογιού πολύ υψηλής ακρίβειας και καθαρότητας, το οποίο θα χρειαστεί η εν λόγω σχεδίαση. Η τοποθεσία των συγκεκριμένων ακροδεκτών φαίνεται στον Πίνακα A-2 του παραρτήματος. Τέλος υπάρχουν οι ακροδέκτες (pins) HARD\_ERROR\_pin, SOFT\_ERROR\_pin, LANE\_UP\_pin και CHANNEL\_UP\_pin που έχουν ως σκοπό την εξαγωγή από τη σχεδίαση σημάτων που αντιστοιχούν στις ενδείξεις σφάλματος που υποδεικνύουν οι παραπάνω ονομασίες. Έπειτα, οι συγκεκριμένοι ακροδέκτες (pins) συνδέονται στις αντίστοιχες φωτοδιόδους (LEDs) της πλακέτας ώστε να παρέχουν εποπτική πληροφόρηση στο χρήστη για την κατάσταση του κυκλώματος.

### Περιφερειακό dcm\_module

Το περιφερειακό αυτό κατασκευάζεται ως ένα τυπικό περιφερειακό του επεξεργαστή PowerPC που όμως δεν χρειάζεται δίαυλο για την επικοινωνία του με τον επεξεργαστή. Χρησιμεύει στη δημιουργία του ρολογιού συστήματος στο οποίο θα γίνει εκτενέστερη αναφορά στη συνέχεια.

### Περιφερειακό mgt\_dcm

Το περιφερειακό αυτό κατασκευάζεται ως ένα τυπικό περιφερειακό του επεξεργαστή PowerPC που επικοινωνεί με αυτόν μέσω του διαύλου OPB ή PLB σε συνδεσμολογία σκλάβου (slave). Διαθέτει ένα IBUFGDS\_LVDS\_25 κι ένα BUFG. Δέχεται ως εισόδους τα σήματα TOP\_BREF\_CLK\_P, TOP\_BREF\_CLK\_N. Ως έξοδοι ορίζονται τα σήματα TOP\_BREF\_CLK και USER\_CLK. Το component IBUFGDS\_LVDS\_25 χρησιμεύει ως buffer ρολογιού διαφορετικής σηματοδότησης και αποκλειστικής εισόδου παρέχοντας δυνατό και σταθερό σήμα. Με τον όρο IBUFGDS περιγράφεται το buffer εισόδου που συνδέεται σε ένα από τα αποκλειστικά διαφορεικά

ζεύγη της συσκευής. Σκοπός του είναι η υλοποίηση ενός διαφορικού ρολογιού εξωτερικής εισόδου. Μπορεί ακόμα να χρησιμοποιηθεί για την άμεση σύνδεσή του με χαμηλής στρέβλωσης πηγή δρομολόγησης ρολογιού πάνω στη συσκευή ελαχιστοποιώντας τη καθυστέρηση ρολογιού. Ακολουθεί το πρότυπο LVDS το οποίο εξασφαλίζει δυνατά και σταθερά σήματα ρολογιού καθώς και ακεραιότητα της πληροφορίας αυτών σε σχεδιάσεις ρολογιού σύγχρονες ως προς την πηγή. Το component BUFG χρησιμεύει για την δρομολόγηση του σήματος εισόδου του σε οικουμενικό δίκτυο σημάτων ρολογιού. Οι διαφορικές εισοδοί, λοιπόν, οδηγούνται στο component IBUFGDS\_LVDS\_25 ώστε να έχει αποκλειστική δρομολόγηση το σήμα εξόδου του, το TOP\_BREF\_CLK, που θα χρησιμεύσει ως το σήμα ρολογιού αναφοράς των MGTs. Έτσι προκύπτει το σήμα ρολογιού αναφοράς που χρησιμεύει για τη μετάδοση δεδομένων σε πολύ υψηλές ταχύτητες και προκύπτει από αποκλειστική δρομολόγηση σήματος ρολογιού καθώς πρέπει να είναι χαμηλού θορύβου και ελάχιστης σκέδασης για την πλέον αξιόπιστη σηματοδότηση ρολογιού. Τέλος το σήμα εξόδου του IBUFGDS\_LVDS\_25 οδηγείται στο BUFG και παράγεται το λιγότερο καθαρό και αξιόπιστο, σήμα ρολογιού χρήστη για τη σχεδίαση χρήστη καθώς και για τον πυρήνα Aurora, το USER\_CLK. Το σήμα ρολογιού χρήστη πρέπει να είναι κλειδωμένο κατά συχνότητα με το σήμα ρολογιού αναφοράς. Ακόμα αξίζει να σημειωθεί ότι όταν υλοποιείται η σχεδίαση που περιλαμβάνει τον πυρήνα Aurora με πλάτος λέξης 4 bytes είναι αναγκαίο ένα δεύτερο ρολόι κλειδωμένο κατά φάση με το ρολόι χρήστη. Αυτό παράγεται με επιπλέον λογική που προστίθεται στο περιφερειακό αυτό. Έτσι το ρολόι χρήστη USER\_CLK «τρέχει» στη μισή συχνότητα του ρολογιού αναφοράς. Και υπάρχει ακόμα ένα ρολόι το USER\_CLK\_N\_2X που είναι κλειδωμένο κατά συχνότητα με το ρολόι αναφοράς και η θετική ακμή αυτού συμπίπτει με την αρνητική ακμή του ρολογιού χρήστη. Αυτό το ρολόι οδηγείται στη συνέχεια στον πυρήνα Aurora μαζί με το ρολόι χρήστη.

#### Περιφερειακά plb\_bram

Πρόκειται για ζεύγος περιφερειακών, ενός block μνήμης και του ελεγκτή της μνήμης αυτής (memory controller). Ο ελεγκτής της μνήμης επικοινωνεί με τον PowerPC μέσω του διαύλου PLB σε συνδεσμολογία σκλάβου (slave). Ο σκοπός της ύπαρξης αυτών των περιφερειακών είναι η παροχή κι επικοινωνία της μνήμης στην οποία φορτώνεται η εφαρμογή του επεξεργαστή, δηλαδή ο χώρος όπου αποθηκεύονται τα δεδομένα (data memory) και φορτώνονται οι εντολές (instruction memory) του προγράμματος που εκτελείται στον PowerPC. Αξίζει να σημειωθεί ότι υπάρχει η δυνατότητα δημιουργίας κρυφής (cache) μνήμης για τα δεδομένα ή για τις εντολές της εφαρμογής ή ακόμα και για τα δύο.

#### Περιφερειακό RS232 Uart

Το περιφερειακό αυτό κατασκευάζεται ως ένα τυπικό περιφερειακό του επεξεργαστή PowerPC που επικοινωνεί με αυτόν μέσω του διαύλου OPB ή PLB σε συνδεσμολογία σκλάβου (slave). Μπορεί να είναι απλού τύπου ή τύπου 16450 είτε 16550. Χρησιμεύει στην επικοινωνία του επεξεργαστή και πιο συγκεκριμένα της εφαρμογής που αυτός εκτελεί με τη σειριακή θύρα του υπολογιστή και κατ' επέκταση με

το τερματικό που προβάλλει στην οθόνη τις σχετικές πληροφορίες της εφαρμογής. Ο ρυθμός μετάδοσης δεδομένων του ανέρχεται στα 460 kb/s.

#### Περιφερειακό reset block

Το περιφερειακό αυτό κατασκευάζεται ως ένα τυπικό περιφερειακό του επεξεργαστή PowerPC που όμως δεν χρειάζεται δίαυλο για την επικοινωνία του με τον επεξεργαστή. Είναι απαραίτητο για την έναρξη λειτουργίας του συστήματος καθώς και για την επανεκκίνησή της.

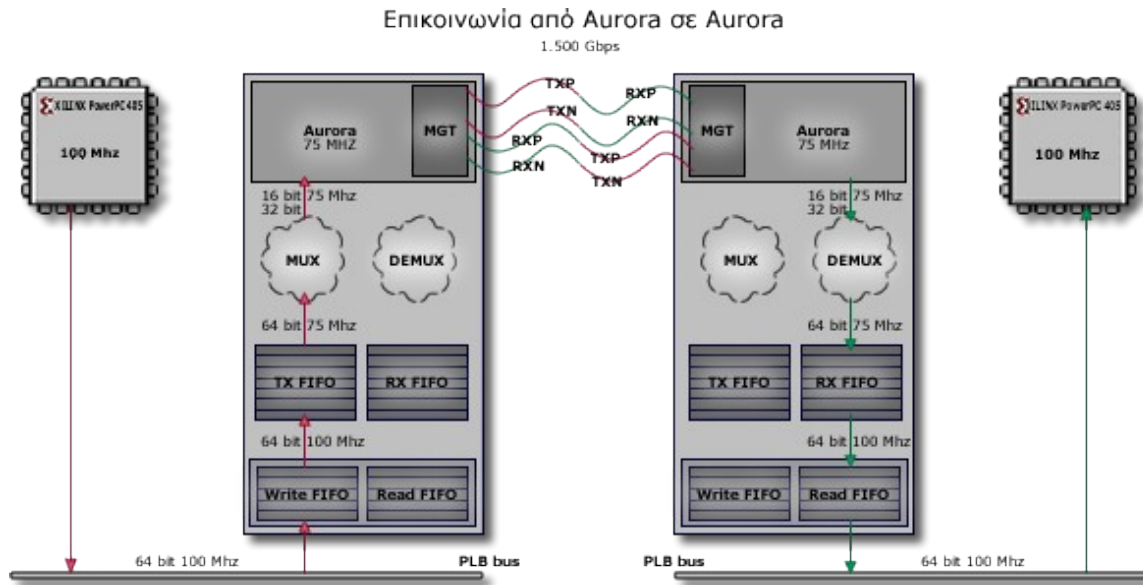
#### Περιφερειακό Throughput\_timer

Το περιφερειακό αυτό κατασκευάζεται ως ένα τυπικό περιφερειακό του επεξεργαστή PowerPC που επικοινωνεί με αυτόν μέσω του διαύλου PLB σε συνδεσμολογία σκλάβου (slave). Διαθέτει ένα καταχωρητή μήκους 64 bits. Δέχεται ως εισόδους τα σήματα START\_TIMER, GET\_LATENCY και GET\_THROUGHPUT καθώς και το ρολόι σχεδίασης χρήστη USER\_CLK. Ως έξοδοι ορίζονται τα σήματα LATENCY και THROUGHPUT.

Ο καταχωρητής μήκους 64 bits χρησιμεύει στην αποθήκευση δύο τιμών μετρητών. Στα 32 λιγότερο σημαντικά bits του καταχωρητή αποθηκεύεται η τιμή του μετρητή που συμβολίζει το λανθάνων χρόνο που χρειάζονται 64 bits δεδομένων από τη στιγμή που θα φτάσουν στην Write FIFO μέχρι τη στιγμή που θα φτάσουν στην Read FIFO σε κύκλους του ρολογιού USER\_CLK. Στα 32 πιο σημαντικά bits του καταχωρητή αποθηκεύεται η τιμή του μετρητή που συμβολίζει τον χρόνο που χρειάζονται όλα τα δεδομένα από τη στιγμή που θα φτάσουν στην Write FIFO μέχρι τη στιγμή που θα φύγουν από την Read FIFO σε κύκλους του ρολογιού USER\_CLK. Το σήμα START\_TIMER εκκινεί το μετρητή, ο οποίος είναι σύγχρονος με το ρολόι σχεδίασης του χρήστη USER\_CLK, συχνότητας στην απλή εκδοχή 75 MHz. Το START\_TIMER ενεργοποιείται μόλις εγγραφεί μια γραμμή δεδομένων στη Write FIFO. Το σήμα GET\_LATENCY καταχωρεί την τιμή του μετρητή στο σήμα LATENCY που συμβολίζει τον λανθάνων χρόνο. Το GET\_LATENCY ενεργοποιείται μόλις εγγραφεί μια γραμμή δεδομένων στη Read FIFO, γεγονός που σημαίνει ότι κατέφθασαν τα πρώτα δεδομένα που μεταδόθηκαν στην Read FIFO του περιφερειακού δέκτη. Το σήμα GET\_THROUGHPUT σταματά το μετρητή καταχωρώντας την τιμή του στο σήμα THROUGHPUT που συμβολίζει το χρόνο μεταφοράς. Το GET\_THROUGHPUT ενεργοποιείται μόλις εγγραφεί και η τελευταία γραμμή δεδομένων στη Read FIFO, γεγονός που σημαίνει ότι κατέφθασαν όλα τα δεδομένα που μεταδόθηκαν στην Read FIFO του περιφερειακού δέκτη.

#### Περιφερειακό auora\_plb

Κατασκευάζεται ως περιφερειακό του επεξεργαστή PowerPC που επικοινωνεί με αυτόν μέσω του διαύλου PLB σε συνδεσμολογία αφέντη (master).



Σχήμα 3.5 : Αναλυτικά η Επικοινωνία από Aurora σε Aurora

## Module Aurora

Σημαντική παράμετρος του Aurora module είναι το γεγονός ότι τα σήματα ελέγχου του module ενεργοποιούνται στην τιμή '0', είναι δηλαδή αρνητικής λογικής. Αξίζει να σημειωθεί ότι τα διαφορεικά σήματα αποστολής και λήψης δεδομένων TXP, TXN και RXP, RXN αντίστοιχα δεν χρησιμοποιούνται στη σχεδίασή μας για την μετάδοση των δεδομένων. Πρόκειται για σήματα εξωτερικά των κυκλωμάτων των MGTs που συνδέονται για την μετάδοση των δεδομένων από τον ένα MGT στον άλλο. Στη σχεδίασή μας, όμως λόγω της ύπαρξης του PowerPC, ελέγχεται η εγγραφή (είτε μέσω DMA είτε μέσω PLB) στη μνήμη του περιφερειακού και συγκεκριμένα στις FIFOs των δεδομένων και από εκεί εκτελείται μετάδοση των δεδομένων μέσω λογικής και σημάτων στις θύρες TXDATA και RXDATA του module των MGT που καλείται GT\_CUSTOM. Συνεπώς δε χρησιμοποιούνται οι εξωτερικές διαφορεικές θύρες που αναφέρθηκαν.

## Module GT\_CUSTOM

Πρόκειται για το module που υλοποιεί το Multi-Gigabit πομποδέκτη. Συνεπώς ο ορισμός του και η λειτουργία του είναι καθοριστική. Αυτό το module περικλείεται μέσα στον πυρήνα Aurora.

### Module Clock Compensation

Πρόκειται για το module που υλοποιεί τη διόρθωση ρολογιού (για περισσότερα βλέπε “Διόρθωση Ρολογιού”, σελ. 33). Απαρτίζεται από μια μηχανή πεπερασμένων καταστάσεων που ουσιαστικά αποτελεί ένα μετρητή τριών τομέων. Ο πρώτος τομέας μετρά την περίοδο των idles πριν γίνει η διόρθωση ρολογιού. Ο δεύτερος τομέας μετρά την περίοδο κατά την διάρκεια της οποίας δεν πρέπει να επιχειρηθούν οι λειτουργίες ελέγχου NFC και UFC επειδή δεν θα ολοκληρωθούν. Ο τελευταίος τομέας μετρά τους κύκλους που διαρκεί η ακολουθία διόρθωσης ρολογιού.

## Module Διεπαφής Περιφερειακού με Δίαυλο

Πρόκειται για το module που υλοποιεί τη διεπαφή του περιφερειακού Aurora με το δίαυλο PLB. Απαρτίζεται από λογική που συνδέει τα σήματα του διαύλου PLB με το περιφερειακό και αντιστρόφως. Επισημαίνεται η ύπαρξη δύο FIFOs στο εσωτερικό του,

που χρησιμεύουν στην επικοινωνία του περιφερειακού με την εφαρμογή χρήστη που εκτελείται στον επεξεργαστή PowerPC. Οι FIFOs αυτές διαθέτουν πλάτος 64 bits και βάθος 512 ή 1024 γραμμών. Ακόμα, χρονίζονται από το ρολόι του διαύλου και παρέχουν σήματα empty, full, vacancy, occupancy που εμφανίζουν ενδείξεις σχετικά με το πλήθος των δεδομένων στις FIFOs. Κατόπιν πειραματικών μελετών υπολογίζεται ότι ο χρόνος ανάγνωσης και εγγραφής των FIFOs αυτών είναι 1 κύκλος ρολογιού τους. Έτσι, υλοποιούνται στη διεπαφή του περιφερειακού οι Write FIFO και Read FIFO της σχεδίασης. Σημειωτέον, με τις αντίστοιχες εντολές transferDMA, όπως και WritePLB και ReadPLB μεταφέρονται δεδομένα από και προς το περιφερειακό μέσω DMA ή του PLB. Αυτός είναι κι ο κυρίως λόγος που χρησιμοποιείται αυτό το επιπλέον ζεύγος FIFO. Επιπλέον, τα σήματα που παρακολουθούν το πλήθος δεδομένων στις FIFOs αυτές, βοηθούν την έναρξη και παύση μετάδοσης δεδομένων από και προς αυτές, καθώς και στην ενεργοποίηση των κατάλληλων σημάτων για τη μέτρηση του throughput της σχεδίασης μέσω του υλικού.

Ρολόι σχεδίασης Aurora, ρολόι των MGTs και ρολόι συστήματος

#### *Ρολόι σχεδίασης*

Το ρολόι της σχεδίασής μας είναι το USER\_CLK. Τα χαρακτηριστικά αυτού του ρολογιού είναι ότι τρέχει σε συχνότητα 75 MHz και παράγεται από το περιφερειακό mgt\_dcm v1.00.a. Η περίοδός του ορίζεται στο αρχείο .ucf, στη γραμμή:

```
NET USER_CLK PERIOD = 75 MHz;
```

Το ρολόι αυτό εισέρχεται στο περιφερειακό aurora\_plb v1.00.a από τη θύρα USER\_CLK του περιφερειακού. Στη συνέχεια οδηγείται στη RX FIFO με όνομα rx\_fifo\_i, μέσω της θύρας της wr\_clk ώστε να εκτελείται συγχρονισμένα με αυτό το ρολόι η εγγραφή των ληφθέντων δεδομένων στη FIFO αυτή καθώς και στην TX FIFO με όνομα tx\_fifo\_i, μέσω της θύρας της rd\_clk ώστε να εκτελείται συγχρονισμένα με αυτό το ρολόι η ανάγνωση των δεδομένων προς αποστολή. Επίσης χρησιμοποιείται ως είσοδος στο standard\_cc\_module μέσω της θύρας του, USER\_CLK. Τέλος αυτό το σήμα USER\_CLK το δέχεται ως είσοδο το module aurora\_201 στη θύρα USER\_CLK. Το μόνο σύνθετο και το κυριότερο module από αυτά είναι το aurora\_201 το οποίο αποτελείται και από άλλα modules. Συνεπώς εμπεριέχει το module GT\_CUSTOM που είναι αυτό που συνδέει όλη τη σχεδίαση με τους MGTs στο οποίο και τοποθετείται ως είσοδος στις θύρες TXUSRCLK, TXUSRCLK2 και RXUSRCLK, RXUSRCLK2. Ακόμα το ρολόι της σχεδίασης χρησιμοποιείται στα modules TX\_STREAM και RX\_STREAM εφόσον η σχεδίασή είναι σε λειτουργία streaming και στα αντίστοιχα modules TX\_LL και RX\_LL για τη λειτουργία με πακέτα. Επίσης, παρέχεται ως είσοδος ρολογιού στα λειτουργικά τμήματα AURORA\_LANE και GLOBAL\_LOGIC. Τέλος, αξίζει να σημειωθεί ότι υπάρχουν δύο process στο user\_logic του aurora\_plb όπου γίνεται πολύπλεξη και αποπολύπλεξη οι οποίες λειτουργίες γίνονται με βάση το ρολόι της σχεδίασης USER\_CLK. Στην υλοποίηση με λειτουργία πακέτου προστίθεται ένα επιπλέον process για τη λειτουργία native flow control που συγχρονίζεται με το ρολόι USER\_CLK. Επίσης στις υλοποιήσεις για τη μέτρηση του throughput μέσω του υλικού υπάρχει ακόμα ένα process ελεγχόμενο από το ρολόι USER\_CLK.

#### *Ρολόι των MGTs*

Το ρολόι των MGTs είναι το TOP\_BREF\_CLK. Τα χαρακτηριστικά αυτού του ρολογιού είναι ότι τρέχει σε συχνότητα 75 MHz και παράγεται από το περιφερειακό



mgt\_dcm v1.00.a μέσω διαφορικής σηματοδότησης των TOP\_BREF\_CLK\_P και TOP\_BREF\_CLK\_N. Η περίοδός του ορίζεται στο αρχείο .ucf, στη γραμμή:

```
NET TOP_BREF_CLK PERIOD = 75 MHz;
```

Το ρολόι των MGT πρέπει να είναι υψηλής ποιότητας και για αυτό προέρχεται από αποκλειστική δρομολόγηση ρολογιού πάνω στην πλακέτα. Έτσι στο .ucf αρχείο ορίζονται οι παράμετροι των διαφορικών ρολογιών που οδηγούνται στο mgt\_dcm και παράγουν το υψηλής αξιοπιστίας ρολόι των MGTs. Οι τοποθεσίες LOC=F16 και LOC=G16 φέρουν το ρολόι BREF\_CLK που είναι και το πλέον αξιόπιστο που διαθέτει η FPGA και χρησιμοποιείται για την μεταφορά δεδομένων σε ταχύτητες > 2.5 GBaud per second. Το ρολόι αυτό εισέρχεται στο περιφερειακό από τη θύρα TOP\_BREF\_CLK και στη συνέχεια οδηγείται στο module aurora\_201 όπου ορίζεται ως είσοδος και συνδέεται στη θύρα TOP\_BREF\_CLK. Εντός του aurora\_201 καταλήγει στο module που υλοποιείται ο MGT το GT\_CUSTOM. Συνδέεται στην είσοδό του, BREFCLK.

#### *Ρολόι συστήματος, δίαυλου και επεξεργαστή*

Το ρολόι του συστήματος είναι το sys\_clk\_s. Τα χαρακτηριστικά αυτού του ρολογιού είναι ότι τρέχει σε συχνότητα 100 MHz και παράγεται από το περιφερειακό dcm\_module v1.00.a. Η περίοδός του ορίζεται στο αρχείο .ucf ως 10000 ps μέσω του εξωτερικού σήματος sys\_clk\_pin που συνδέεται στο dcm\_clk\_s. Οδηγείται, λοιπόν το σήμα dcm\_clk\_s στο dcm\_module και εξάγεται το σήμα sys\_clk\_s που αποτελεί και το ρολόι του επεξεργαστή καθώς και του διαύλου plb όπως επίσης και του διαύλου orb. Τέλος στο interface plb\_ipif του περιφερειακού aurora\_plb v1.00.a με το δίαυλο plb εισέρχεται ένα ρολόι. Το ρολόι αυτό είναι το ρολόι PLB\_CLK δηλαδή το ρολόι του διαύλου plb που όπως αναφέρθηκε είναι το ρολόι συστήματος sys\_clk\_s. Επιπροσθέτως, το plb\_ipif interface βγάζει ως έξοδο το ρολόι Bus2IP\_Clk. Πρόκειται για ρολόι του IPIC τμήματος του plb\_ipif που παρέχεται ως είσοδος στο περιφερειακό και συμπίπτει με το PLB\_Clk. Αυτό το σήμα ρολογιού, στη συνέχεια, οδηγείται στη RX FIFO με όνομα rx\_fifo\_i, μέσω της θύρας της rd\_clk, όπως επίσης και στην TX FIFO με όνομα tx\_fifo\_i, μέσω της θύρας της wr\_clk.

#### *Άλλα ρολόγια*

Αξίζει να σημειωθεί ότι μέσα στο module aurora\_201 το σήμα από τη θύρα εξόδου του module GT\_CUSTOM με όνομα RXRECCLK οδηγείται στη θύρα εισόδου του module PHASE\_ALIGN. Πρόκειται για το ανακτημένο ρολόι RXRECCLK που προέρχεται από τη διαίρεση του ρυθμού λήψης σειριακών δεδομένων με το 20.

#### *Write και Read FIFOs*

Κατ' αρχάς οι είσοδοι και έξοδοι των FIFOs αυτών είναι συνδεδεμένες στις TX και RX FIFOs και στη διεπαφή με το δίαυλο PLB. Ο χρονισμός των FIFOs αυτών εξασφαλίζεται από ένα ρολόι, το PLB\_Clk, του οποίου η τιμή ανατίθεται στο σήμα ρολογιού Bus2IP\_Clk\_i, Bus2IP\_Clk\_i <= PLB\_Clk; και κατ' επέκταση στο Bus2IP\_Clk. Πρόκειται για το ρολόι του διαύλου PLB που λειτουργεί σε συχνότητα 100 MHz. Το πλάτος των δεδομένων που αποθηκεύουν οι FIFOs είναι 64 bits, όσο και το πλάτος των δεδομένων που μεταφέρει ο δίαυλος PLB.

#### *Ανάγνωση*

Για τη διαδικασία ανάγνωσης από τη Write FIFO, ενεργοποιείται η θύρα IP2WFIFO\_RdReq από την εφαρμογή χρήστη, και τίθεται στην τιμή '1', έτσι

προκαλώντας μια ανάγνωση στην επόμενη θετική ακμή του ρολογιού Bus2IP\_Clk. Η θύρα ενεργοποίησης ανάγνωσης ελέγχεται από δύο σήματα το WFIFO2IP\_empty και το tx\_fifo\_almost\_full. Έτσι ενεργοποιείται μόνο όταν και τα δύο δεν είναι ενεργά, που σημαίνει ότι η Write FIFO δε θα είναι άδεια και η TX FIFO δε θα είναι σχεδόν γεμάτη. Τότε η FIFO τοποθετεί στην έξοδο WFIFO2IP\_Data την επόμενη διαθέσιμη λέξη πλάτους 64 bits κι ενεργοποιεί στην τιμή '1' το σήμα WFIFO2IP\_RdAck, υποδεικνύοντας μια επιτυχημένη ανάγνωση και ελευθερώνοντας μια θέση στη FIFO. Η έξοδος της Write FIFO είναι συνδεδεμένη με την είσοδο της TX FIFO, tx\_fifo\_din <= WFIFO2IP\_Data. Ακόμα η θύρα επικύρωσης ανάγνωσης WFIFO2IP\_RdAck είναι συνδεδεμένη με τη θύρα ενεργοποίησης εγγραφής της TX FIFO, tx\_fifo\_we <= WFIFO2IP\_RdAck.

Για τη διαδικασία ανάγνωσης από την Read FIFO, ο χρήστης αρκεί να καλέσει την κατάλληλη συνάρτηση μέσω λογισμικού. Αυτή η συνάρτηση εκτελεί μεταφορά δεδομένων, μεγέθους όσων bytes ζητηθεί από το χρήστη, από τη Read FIFO του περιφερειακού στη μνήμη του επεξεργαστή, μέσω του διαύλου PLB είτε μέσω DMA. Αυτή η ενέργεια πραγματοποιείται κατόπιν ελέγχου συνθήκης που εξασφαλίζει ότι η Read FIFO δε θα είναι άδεια και άρα θα έχει δεδομένα προς ανάγνωση.

### *Εγγραφή*

Για τη διαδικασία εγγραφής στη Write FIFO, ο χρήστης αρκεί να καλέσει την κατάλληλη συνάρτηση μέσω λογισμικού. Αυτή η συνάρτηση εκτελεί μεταφορά δεδομένων, μεγέθους όσων bytes ζητηθεί από το χρήστη, από τη μνήμη του επεξεργαστή στη Write FIFO του περιφερειακού, μέσω του διαύλου PLB ή μέσω DMA. Αυτή η ενέργεια πραγματοποιείται κατόπιν ελέγχου συνθήκης που εξασφαλίζει ότι η Write FIFO διαθέτει τόσες θέσεις ελεύθερες όσες πρόκειται να καταλάβουν τα δεδομένα προς εγγραφή.

Για τη διαδικασία εγγραφής στη Read FIFO ενεργοποιείται η θύρα IP2RFIFO\_WrReq και τίθεται στην τιμή '1', έτσι προκαλώντας μια εγγραφή στην επόμενη θετική ακμή του ρολογιού Bus2IP\_Clk. Η θύρα ενεργοποίησης εγγραφής ελέγχεται από το σήμα rx\_fifo\_valid. Έτσι ενεργοποιείται μόνο όταν έχει διαδραματιστεί μια έγκυρη ανάγνωση από την RX FIFO, IP2RFIFO\_WrReq <= rx\_fifo\_valid;. Τότε η Read FIFO δέχεται από την είσοδο IP2RFIFO\_Data την επόμενη διαθέσιμη λέξη πλάτους 64 bits η οποία καταλαμβάνει μια θέση σε αυτή. Η είσοδος δεδομένων της Read FIFO είναι συνδεδεμένη με την έξοδο δεδομένων της RX FIFO, IP2RFIFO\_Data <= rx\_fifo\_dout;

### TX και RX FIFOs

Κατ' αρχάς οι είσοδοι κι εξοδοι των FIFOs είναι στενά συνδεδεμένες με τις μεθόδους πολύπλεξης και αποπολύπλεξης. Ένα σημαντικό στοιχείο για τη σηματοδότηση του ρολογιού τους, όπως έχει ήδη αναφερθεί, είναι πως η FIFO με όνομα rx\_fifo\_i, δέχεται μέσω της θύρας της rd\_clk, όπως επίσης και η FIFO με όνομα tx\_fifo\_i, δέχεται μέσω της θύρας της wr\_clk το ρολόι που βγάζει ως έξοδο η διεπαφή plb\_ipif του περιφερειακού με το δίαυλο plb. Αυτό το ρολόι ταυτίζεται με το ρολόι του διαύλου. Επίσης η FIFO με όνομα rx\_fifo\_i και η tx\_fifo\_i, δέχεται στη θύρα της wr\_clk και rd\_clk, αντίστοιχα, το ρολόι της σχεδίασης USER\_CLK. Ακόμα ένα σημαντικό στοιχείο για τις FIFOs είναι πως η είσοδος και έξοδος των δεδομένων τους είναι πλάτους 64 bits

όπως και το πλάτος των ενσωματωμένων Read και Write FIFO της διεπαφής του περιφερειακού με το δίαυλο, όπως επίσης και του διαύλου του ίδιου.

#### *Ανάγνωση*

Για τη διαδικασία ανάγνωσης από την TX FIFO ο χρήστης ενεργοποιεί τη θύρα της, `rd_en` και τη θέτει στην τιμή '1', έτσι προκαλώντας μια ανάγνωση στην επόμενη θετική ακμή του ρολογιού `rd_clk`. Τότε η FIFO τοποθετεί στην έξοδο `dout` την επόμενη διαθέσιμη λέξη πλάτους 64 bits και ενεργοποιεί στην τιμή '1' το σήμα `valid`, υποδεικνύοντας μια επιτυχημένη ανάγνωση και ελευθερώνοντας μια θέση στη FIFO. Η έξοδος της TX FIFO είναι συνδεδεμένη με την είσοδο του `mux` και το σήμα `valid` με το σήμα ενεργοποίησης εγγραφής `mux_we` του `mux`. Η θύρα `rd_en` της TX FIFO είναι συνδεδεμένη με το σήμα `tx_fifo_re` το οποίο με τη σειρά του ελέγχεται από το σήμα `mux_rreq`.

Για τη διαδικασία ανάγνωσης από την RX FIFO ο χρήστης ενεργοποιεί το σήμα της, `rd_en` και το θέτει στην τιμή '1', έτσι προκαλώντας μια ανάγνωση στην επόμενη θετική ακμή του ρολογιού `rd_clk`. Τότε η FIFO τοποθετεί στην έξοδο `dout` την επόμενη διαθέσιμη λέξη πλάτους 64 bits και ενεργοποιεί στην τιμή '1' το σήμα `valid`, υποδεικνύοντας μια επιτυχημένη ανάγνωση και ελευθερώνοντας μια θέση στη FIFO. Η έξοδος της RX FIFO είναι συνδεδεμένη με την είσοδο της Read FIFO και το σήμα `valid` με το σήμα ενεργοποίησης εγγραφής της Read FIFO. Η θύρα `rd_en` της RX FIFO είναι συνδεδεμένη με το σήμα `rx_fifo_re`. Αυτό με τη σειρά του διαμορφώνεται από τα σήματα `not rx_fifo_empty` της RX FIFO και το σήμα `not RFIFO2IP_AlmostFull` της Read FIFO που εξάγεται από τη διεπαφή `plb_ipif`.

#### *Εγγραφή*

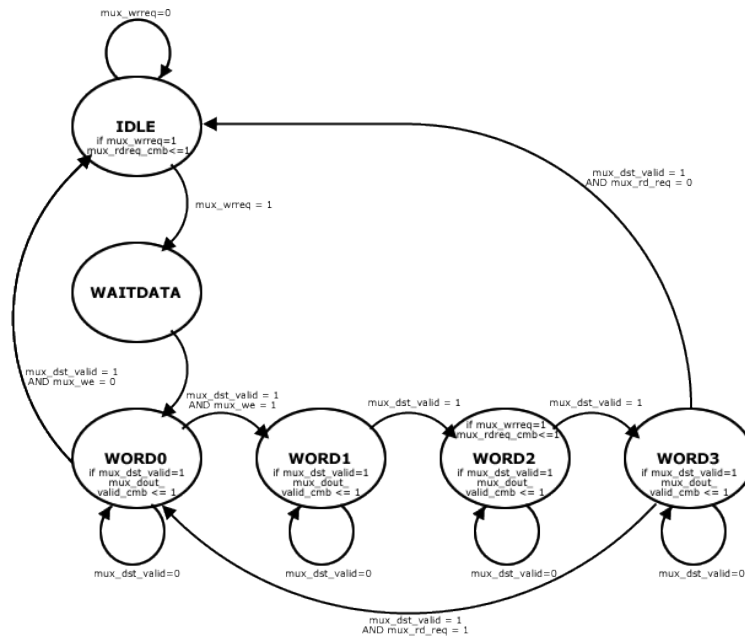
Για τη διαδικασία εγγραφής στην TX FIFO ο χρήστης ενεργοποιεί τη θύρα της, `wr_en` και τη θέτει στην τιμή '1', έτσι προκαλώντας μια εγγραφή στην επόμενη θετική ακμή του ρολογιού `wr_clk`. Τότε η FIFO δέχεται από την είσοδο `din` την επόμενη διαθέσιμη λέξη πλάτους 64 bits, η οποία καταλαμβάνει μια θέση στην TX FIFO. Η είσοδος της TX FIFO είναι συνδεδεμένη με την έξοδο της Write FIFO. Η θύρα `wr_en` της TX FIFO είναι συνδεδεμένη με το σήμα `tx_fifo_we` το οποίο με τη σειρά του ελέγχεται από το σήμα έγκυρης ανάγνωσης `WFIFO2IP_RdAck` της Write FIFO που εξάγεται από τη διεπαφή `plb_ipif`.

Για τη διαδικασία εγγραφής στην RX FIFO ο χρήστης ενεργοποιεί τη θύρα της, `wr_en` και τη θέτει στην τιμή '1', έτσι προκαλώντας μια εγγραφή στην επόμενη θετική ακμή του ρολογιού `wr_clk`. Τότε η FIFO δέχεται από την είσοδο `din` την επόμενη διαθέσιμη λέξη πλάτους 64 bits καταλαμβάνοντας μια θέση στη FIFO. Η είσοδος της RX FIFO είναι συνδεδεμένη με την έξοδο του `demux`. Η θύρα `wr_en` της RX FIFO είναι συνδεδεμένη με το σήμα `rx_fifo_we` το οποίο με τη σειρά του ελέγχεται από το σήμα `demux_valid`.

#### Πολυπλεξία και αποπολυπλεξία

Η πολυπλεξία και αποπολυπλεξία στη σχεδίαση αυτή είναι απαραίτητη γιατί χρειάζεται να μεταφερθούν τα δεδομένα προς αποστολή ή λήψη από το interface του διαύλου `plb` που είναι πλάτους 64 bits στο interface του module του Aurora που είναι πλάτους 16 bits ή 32 bits και αντιστρόφως.

## Πολυπλεξία



Σχήμα 3.6 : Μηχανή FSM πολυπλεξίας

Για τη μετατροπή των δεδομένων που λαμβάνονται από το διάυλο plb πλάτους 64 bits και προορίζονται για αποστολή μέσω του TX interface του Aurora πλάτους 16 ή 32 bits είναι αναγκαίος ένας πολυπλέκτης. Έτσι σε συνδυασμό με την TX FIFO αποθηκεύονται προσωρινά τα δεδομένα πλάτους 64 bits και μεταβιβάζονται στο TX interface του Aurora ανά 16 ή 32 bits σε διάστημα 4 ή 2 κύκλων αντίστοιχα.

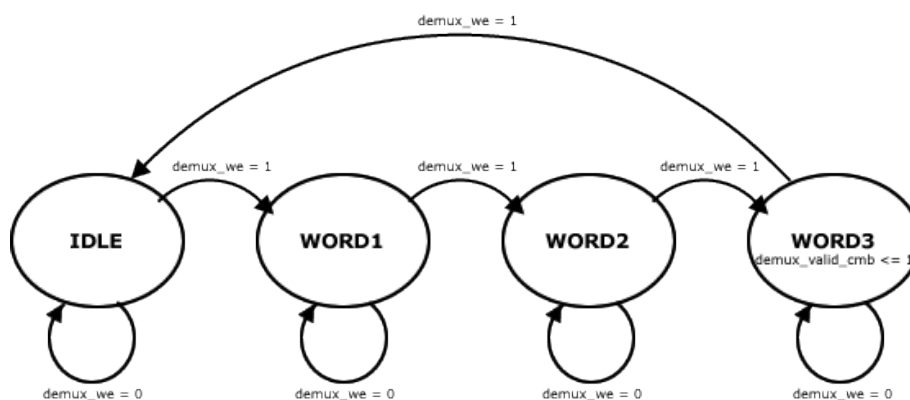
Για την εκτέλεση αυτής της διαδικασίας χρησιμοποιούνται δυο process, το ένα αποτελεί μια FSM πολύπλεξης και το άλλο έναν controller των καταστάσεων αυτής που συγχρονίζεται με το ρολόι της σχεδίασης USER\_CLK. Αρχικοποιείται η μηχανή της πολύπλεξης στην κατάσταση `mux_c_state <= IDLE` όπου το σήμα `mux_c_state` συμβολίζει την τρέχουσα κατάσταση της μηχανής. Σε αυτήν την κατάσταση παραμένει η μηχανή μέχρις ότου το σήμα `mux_wrrreq` ενεργοποιηθεί στην τιμή `not tx_fifo_empty` που συμβαίνει όταν η TX FIFO δεν θα είναι άδεια, άρα έχει μέσα δεδομένα προς αποστολή. Όταν το σήμα `mux_wrrreq` πάρει την τιμή '1', τότε η μηχανή μεταβαίνει στην κατάσταση WAITDATA αφού πρώτα θέτει την τιμή '1' στο σήμα `mux_rdreq_cmb`. Η κατάσταση

αυτή είναι αποκλειστικά και μόνο βοηθητική και χρησιμεύει στο να περιμένει η μηχανή ένα κύκλο ώστε να μεταδοθεί η τιμή του σήματος `mux_rdreq_cmb` στο σήμα `mux_rdreq`. Το σήμα αυτό στην επόμενη θετική ακμή θα ενεργοποιήσει τη θύρα `rd_en` της TX FIFO προκαλώντας την εμφάνιση στην έξοδο της TX FIFO 64 bit δεδομένων. Τότε η μηχανή θα έχει μπει στην κατάσταση `WORD0` που όπως υποδηλώνεται από την ονομασία της είναι η κατάσταση όπου η πρώτη λέξη του Aurora πλάτους 16 ή 32 bits αντιγράφεται από την TX FIFO μέσω του σήματος `mux_din` στο σήμα του πολυπλέκτη `mux_dout_cmb` με σκοπό στην επόμενη θετική ακμή να μεταβιβαστεί στο σήμα του πολυπλέκτη `mux_dout` και από εκεί στην `TX_D` θύρα του Aurora module μέσω του σήματος `tx_d_i`. Αξίζει να σημειωθεί ότι το σήμα `mux_din` αντιγράφεται στο `mux_din_buf`, ώστε να αποθηκευτούν προσωρινά τα δεδομένα της εξόδου της FIFO, μέχρι να ολοκληρωθεί η πολύπλεξη όλων των λέξεων που αυτή φέρει. Για να εξασφαλιστεί ότι όλα είναι έτοιμα στον πυρήνα Aurora για τη μετάδοση δεδομένων εκτελείται έλεγχος του σήματος `mux_dst_valid` το οποίο παίρνει τιμές από το σήμα `tx_dst_rdy_n_i` το οποίο με τη σειρά του είναι συνδεδεμένο με τη θύρα `TX_DST_RDY_N` του πυρήνα Aurora. Αυτό το σήμα ελέγχει αν ο πυρήνας Aurora που βρίσκεται στην πλευρά του δέκτη της γραμμής μετάδοσης είναι έτοιμος να δεχτεί πληροφορίες. Σε περίπτωση που δεν είναι έτοιμος τότε απενεργοποιείται το σήμα `TX_DST_RDY_N` (στην τιμή '1') και ο πολυπλέκτης κάνει παύση της μετάδοσης περιμένοντας να ενεργοποιηθεί ξανά. Αυτή η διαδικασία υλοποιείται κρατώντας τα δεδομένα χρήστη στην έξοδο του πολυπλέκτη καθώς και διατηρώντας όλα τα σήματα σταθερά στην τιμή που βρίσκονταν. Το ίδιο ισχύει για το σήμα επόμενης κατάστασης πράγμα που έχει ως αποτέλεσμα η FSM να παραμένει στην ίδια κατάσταση μέχρι να συνεχιστεί κανονικά η μετάδοση. Σημειώνεται ότι έχουν οριστεί βοηθητικά σήματα όπως `mux_we_prop` και `mux_rdreq_prop` που έχουν ως στόχο τη διάδοση της τιμής των σημάτων `mux_we` και `mux_rdreq` αντίστοιχα χωρίς να επηρεάζονται τα σήματα ελέγχου στα οποία είναι συνδεδεμένα αυτά, εκτελώντας ανεπιθύμητες ενέργειες όπως ανάγνωση της TX FIFO για `mux_rdreq = '1'`. Στην περίπτωση που το σήμα `mux_dst_valid` ενεργοποιείται (στην τιμή '0'), εξετάζεται το σήμα `mux_we` το οποίο παίρνει τιμές από το σήμα `tx_fifo_valid` το οποίο με τη σειρά του είναι συνδεδεμένο με τη θύρα `valid` της TX FIFO. Αυτή θα έχει ήδη πάρει την τιμή '1' εφόσον έχει γίνει επιτυχής ανάγνωση. Παράλληλα δίνεται η τιμή '1' στο σήμα `mux_dout_valid_cmb` ώστε στον επόμενο κύκλο να ενεργοποιηθεί στην τιμή '0' η θύρα `TX_SRC_RDY_N` του Aurora\_201 μέσω του σήματος `tx_src_rdy_n_i` που ορίζεται στην τιμή `not mux_dout_valid` και να εκτελεστεί η αποστολή δεδομένων από το Aurora\_201. Ομοίως συνεχίζεται η μετάβαση στις επόμενες καταστάσεις της μηχανής με την ίδια λογική. Αξίζει να σημειωθεί ότι στην κατάσταση `WORD2` εφόσον το σήμα `mux_wreq` φέρει την τιμή '1', τότε η μηχανή αναθέτει την τιμή '1' στο σήμα `mux_rdreq_cmb` με σκοπό στην μεθεπόμενη κατάσταση να εκτελεστεί ανάγνωση δεδομένων από την TX FIFO. Τέλος από την τελευταία κατάσταση, την `WORD3` η επόμενη κατάσταση θεωρείται η `WORD0` εφόσον είναι ενεργοποιημένο το σήμα `mux_rdreq` ή κατάσταση `IDLE` εφόσον το σήμα αυτό είναι απενεργοποιημένο δηλαδή δεν υπάρχουν διαθέσιμα δεδομένα στην TX FIFO.

Πολύ σημαντική λειτουργία επιτελούν και τα σήματα `frame_size` και `frame_cnt`. Τα σήματα αυτά αρχικοποιούνται από την FSM στην τιμή μηδέν. Μόλις αυτή εισέλθει σε κατάσταση που πολυπλέκονται δεδομένα τότε στο σήμα `frame_size` ανατίθεται η τιμή του μεγέθους του πακέτου που αποστέλλεται σε bytes και το σήμα `frame_cnt` αυξάνεται σε κάθε κατάσταση όπου εκτελείται πολυπλεξία δεδομένων κατά 2 ή 4 bytes. Πιο συγκεκριμένα, με το που θα ανατεθεί η πρώτη λέξη των δεδομένων στο αντίστοιχο σήμα της FSM, η ίδια λέξη ορίζεται ως `frame_size`. Αυτό κατόπιν σύμβασης που γίνεται στον

ορισμό πρωτοκόλλου της σχεδίασης (βλέπε ενότητα “Προδιαγραφές Πρωτοκόλλου”, σελ. 88). Ταυτοχρόνως, το σήμα `frame_cnt` λαμβάνει την τιμή  $2 \times \text{AURORA\_WIDTH}$  (2 ή 4 bytes). Ουσιαστικά, το σήμα μέτρησης μεγέθους πλαισίου του πακέτου που αποστέλλεται θα είναι σαν να έχει μετρήσει μια λέξη παραπάνω. Αυτό γίνεται, λόγω της ανάγκης αναγνώρισης του τέλους πακέτου ένα κύκλο πριν τη μεταφορά των αντίστοιχων δεδομένων στην είσοδο του πυρήνα Aurora. Γιατί εκ κατασκευής της FSM χρειάζεται ένας κύκλος ώστε να περάσει ένα σήμα από την είσοδο η ενδιάμεσο αποθηκευτικό χώρο στην έξοδό της. Συνεπώς, με την έναρξη της μέτρησης των bytes πακέτου που αποστέλλονται ενεργοποιείται το σήμα `fsm_tx_sof_n` που συνδέεται στην είσοδο `TX_SOF_N` του Aurora την αρχή του πακέτου. Με την ολοκλήρωση του πλήθους των bytes που περιλαμβάνει το πακέτο ενεργοποιούνται τα σήματα `fsm_tx_eof_n` και `fsm_tx_rem` που είναι συνδεδεμένα στις εισόδους `TX_EOF_N` και `TX_REM` του Aurora που σηματοδοτούν το τέλος του πακέτου.

### Αποπολυπλεξία



Σχήμα 3.7 : Μηχανή FSM αποπολυπλεξίας

Για τη μετατροπή των δεδομένων που λαμβάνονται από το RX interface του Aurora πλάτους 16 ή 32 bits και προορίζονται για αποστολή μέσω του διαύλου `rlb` πλάτους 64 bits είναι αναγκαίος ένας αποπολυπλέκτης. Έτσι σε συνδυασμό με την RX FIFO αποθηκεύονται προσωρινά τα δεδομένα πλάτους 16 bits και μεταβιβάζονται στο δίαυλο `rlb` αφού συμπληρωθεί μια σειρά δεδομένων πλάτους 64 bits σε διάστημα 4 ή 2 κύκλων του Aurora, αντίστοιχα.

Για την εκτέλεση αυτής της διαδικασίας χρησιμοποιούνται δυο process, το ένα αποτελεί μια FSM αποπολύπλεξης και το άλλο έναν controller των καταστάσεων αυτής που συγχρονίζεται με το ρολόι της σχεδίασης `USER_CLK`. Αρχικοποιείται η μηχανή της αποπολύπλεξης στην κατάσταση `demux_c_state <= IDLE` όπου το σήμα `demux_c_state` συμβολίζει την τρέχουσα κατάσταση της μηχανής. Σε αυτήν την κατάσταση παραμένει η μηχανή μέχρις ότου το σήμα `demux_we` ενεργοποιηθεί στην τιμή `not rx_src_rdy_n_i` που συμβαίνει όταν η διεπαφή RX είναι έτοιμη γιατί μόλις έχει λάβει δεδομένα, άρα διαθέτει έγκυρα δεδομένα προς μεταβίβαση στην RX FIFO. Στην κατάσταση `IDLE` η μηχανή αντιγράφει τα δεδομένα από το σήμα `demux_din` στο σήμα του αποπολυπλέκτη `demux_dout_cmb` με σκοπό στην επόμενη θετική ακμή του ρολογιού να μεταβιβαστούν τα δεδομένα από το σήμα `demux_dout_cmb` στο σήμα `demux_dout` και από εκεί στην RX FIFO μέσω του σήματος `rx_fifo_din`. Αυτό συνεχίζεται μέχρι να ολοκληρωθεί η σύνθεση μιας γραμμής 64 bit προς εγγραφή στην RX FIFO στην τελευταία κατάσταση την `WORD3` και στη συνέχεια επανέρχεται η μηχανή αποπολυπλεξίας στην κατάσταση `IDLE`. Σημειωτέον δεν υπάρχει κατάσταση `WORD0`. Κάτι πολύ σημαντικό είναι πως



στην τελευταία κατάσταση της μηχανής ενεργοποιείται το σήμα `demux_valid_cmb` σε '1'. Συνεπώς, στην επόμενη θετική ακμή ρολογιού η τιμή αυτή περνά στο σήμα `demux_valid` και από εκεί στο σήμα `rx_fifo_we` που είναι συνδεδεμένο με τη θύρα `wr_en` της RX FIFO. Έτσι εγγράφονται τα δεδομένα εξόδου του αποπολυπλέκτη σε μια γραμμή της RX FIFO. Τέλος, σημαντική είναι και η διαδικασία που εφαρμόζεται όταν το σήμα `demux_we` είναι απενεργοποιημένο, που συμβαίνει όταν η διεπαφή RX δεν είναι έτοιμη και τα δεδομένα στις θύρες εξόδου της μπορούν να αγνοηθούν. Τότε η FSM παραμένει στην ίδια κατάσταση περιμένοντας το σήμα `demux_we` να ενεργοποιηθεί ξανά. Έχει οριστεί, επιπλέον, το σήμα `demux_end` που εξετάζει τη θύρα εξόδου της διεπαφής RX με όνομα `RX_EOF_N`. Μόλις εντοπίσει ότι είναι ενεργοποιημένη (στην τιμή '0') ενεργοποιείται κι αυτό με τη σειρά του ώστε να επισημάνει το τέλος της λήψης του τρέχοντος πακέτου και κατ' επέκταση την τελευταία εγγραφή στη RX FIFO.

### *Έλεγχος Ροής*

Όλο το σύστημα που υλοποιείται περιλαμβάνει 4 FIFOs απαραίτητες ως μέσα προσωρινής αποθήκευσης και μέσα επικοινωνίας με τον επεξεργαστή έτσι ώστε να μην υπάρχει απώλεια δεδομένων. Για αυτό το σκοπό είναι απαραίτητοι κι οι έλεγχοι της πληρότητας των FIFOs. Αν υπάρχει χώρος για την επόμενη γραμμή δεδομένων, η λειτουργία του συστήματος είναι απρόσκοπτη, αλλιώς έχει υλοποιηθεί μέθοδος που θέτει το σύστημα σε αναμονή μέχρι να αδειάσει η αντίστοιχη FIFO. Οι TX FIFO και Read FIFO, όπως έχει αναφερθεί, ελέγχονται από τις παρακάτω γραμμές κώδικα :

```
IP2WFIFO_RdReq <= (not WFIFO2IP_empty) and (not tx_fifo_almost_full);
rx_fifo_re <= (not rx_fifo_empty) and (not RFIFO2IP_AlmostFull);
```

Έτσι αν οι εν λόγω FIFOs είναι “σχεδόν γεμάτες” τότε δεν εκτελείται ανάγνωση της αντίστοιχης FIFO που προηγείται, όπως εδώ της Write FIFO για την TX FIFO και της RX FIFO για την Read FIFO. Η Write FIFO ελέγχεται στον κώδικα του API, όπου και εξετάζεται η συνθήκη πληρότητάς της και κατά πόσο έχει αρκετό χώρο για τον όγκο των δεδομένων που θα τις μεταφερθούν. Σε περίπτωση που δεν έχει αρκετό διαθέσιμο χώρο εκτελείται `while(1) loop` μέχρις ότου απελευθερωθούν θέσεις της και χωράει τα συγκεκριμένα δεδομένα. Τέλος, σημειώνεται ότι η RX FIFO κατέχει πολύ σημαντική θέση στο σύστημα μιας και καταλαμβάνει την πιο καίρια θέση, αυτής της πρώτης FIFO του δέκτη. Συνεπώς δεν πρέπει φτάσει στο σημείο που δεν θα μπορεί να δέχεται δεδομένα λόγω πληρότητας και τα εισερχόμενα δεδομένα να “πέφτουν στο κενό”. Κάτι τέτοιο εξασφαλίζεται με τη λειτουργία τοπικού ελέγχου ροής (NFC), δυνατότητας που παρέχει ο πυρήνας Aurora.

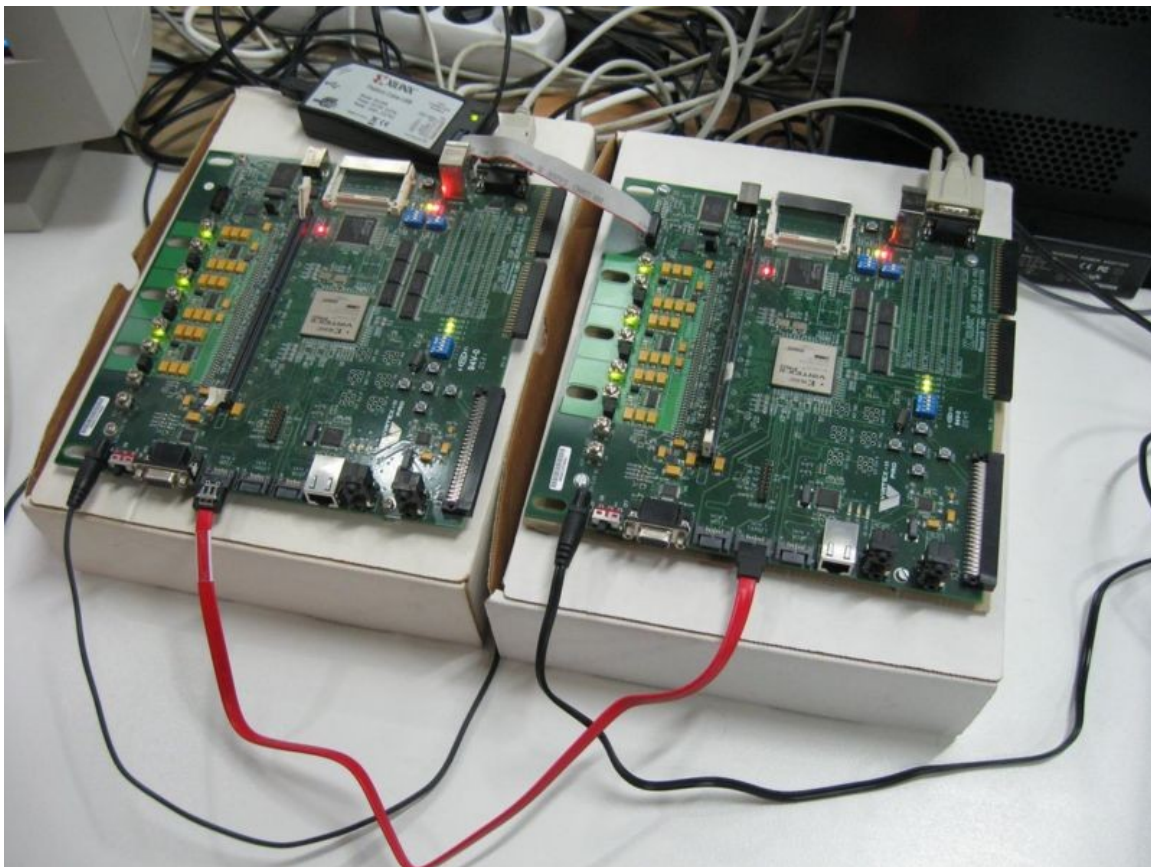
Αναλυτικά, όταν η RX FIFO είναι “σχεδόν γεμάτη” το σήμα `rx_fifo_almost_full` θα είναι ενεργοποιημένο. Από εκεί και πέρα το σήμα δίνεται ως είσοδος σε μια μηχανή πεπερασμένων καταστάσεων. Πρόκειται για δύο process που υλοποιούν αυτήν την FSM. Ένα process που φέρει τις καταστάσεις της μηχανής και τους αντίστοιχους ελέγχους κι ένα άλλο που υλοποιεί το χρονισμό της μηχανής με το ρολόι της σχεδίασης καθώς και τις καταχωρήσεις των αντίστοιχων τιμών στα σήματα εξόδου ή σε σήματα εσωτερικής χρήσης της μηχανής. Έτσι, λοιπόν, όταν ενεργοποιηθεί το σήμα `rx_fifo_almost_full` τότε ενεργοποιούνται ακολούθως και οι θύρες εισόδου του πυρήνα Aurora με όνομα `NFC_REQ_N` και `NFC_NB`. Στο δεύτερο σήμα εισόδου ανατίθεται η τιμή “1111” που συμβολίζει τη λειτουργία XOFF. Αυτή η λειτουργία τοπικού ελέγχου ροής εκτελεί διακοπή της ροής δεδομένων στη γραμμή μετάδοσης. Κάτι που πραγματοποιείται μετά την ολοκλήρωση της αποστολής της τρέχουσας λέξης (που βρίσκεται στο `TX_D` τη στιγμή που δοθεί το σήμα) γιατί ο τοπικός έλεγχος ροής βρίσκεται σε “completion mode” δηλαδή περιμένει πρώτα να ολοκληρωθεί η αποστολή της τρέχουσας λέξης και μετά

αρχίζει τη διαδικασία αποστολής idles. Το σήμα XOFF συνεπάγεται διακοπή ροής δεδομένων και αποστολή idles μέχρι νεωτέρας, δηλαδή μέχρι την αποστολή σήματος συνέχισης της μετάδοσης δεδομένων XON. Ταυτόχρονα αναμένεται η ενεργοποίηση του σήματος NFC\_ACK\_N. Μόλις ενεργοποιηθεί το σήμα αυτό αρχίζει η αποστολή idles δίνοντας έτσι την ευκαιρία στη RX FIFO να αδειάσει γραμμές δεδομένων ώστε να χωράει τα επόμενα. Εν συνεχεία, λοιπόν, παράγονται idles κι απενεργοποιούνται τα σήματα RX\_SRC\_RDY\_N και μαζί του το TX\_DST\_RDY\_N. Άρα η σχεδίαση χρήστη αγνοεί τα σήματα δεδομένων του πυρήνα. Η FSM φροντίζει να διατηρεί τις τιμές των σημάτων της όπως επίσης και τις τιμές που υπάρχουν στην διεπαφή TX. Μόλις απενεργοποιηθεί το σήμα rx\_fifo\_almost\_full τότε ενεργοποιούνται ακολούθως και οι θύρες εισόδου του πυρήνα Aurora με όνομα NFC\_REQ\_N και NFC\_NB. Στο δεύτερο σήμα εισόδου ανατίθεται η τιμή “0000” που συμβολίζει τη λειτουργία XON. Έτσι ο πυρήνας επανέρχεται στην κανονική λειτουργία μετάδοσης δεδομένων.

### 3.3.2 Υλοποίηση πρωτοκόλλου Aurora με χρήση επεξεργαστή σε δύο XUP

Πρόκειται για επέκταση της προηγούμενης υλοποίησης ώστε να εκτελεστεί σειριακή επικοινωνία μεταξύ δύο πλακετών XUP Virtex™-II Pro που διαθέτουν RocketIO. Αυτή η υλοποίηση περιλαμβάνει το πρωτόκολλο επιπέδου διασύνδεσης Aurora, που χρησιμοποιείται από περιφερειακό υλοποιημένο πάνω στο δίαυλο PLB συνδεδεμένο με τον επεξεργαστή PowerPC, όπως επίσης κι άλλη λογική που υλοποιείται σε δυο XUP Virtex™-II Pro μεταφέροντας δεδομένα από την μια πλακέτα στην άλλη. Αυτή η σχεδίαση είναι πιο ευέλικτη γιατί επιτρέπει επικοινωνία με ποικιλία εφαρμογών χρήστη που τρέχουν στον PowerPC και ταυτόχρονα δίνει τη δυνατότητα πραγματικής επικοινωνίας δυο πλακετών. Χρησιμοποιήθηκε η σχεδίαση που δημιουργήθηκε για μια πλακέτα XUP με αλλαγές στην κυρίως εφαρμογή της σχεδίασης και μικρή μετατροπή όπως τη χρήση από την κάθε πλακέτα μόνο ενός από τα δύο περιφερειακά Aurora που κατασκευάζονται στην προηγούμενη σχεδίαση.

- Τροποποιήθηκε η εφαρμογή που εκτελείται στον PowerPC που εκκινεί κι ελέγχει όλη τη διαδικασία της μεταφοράς δεδομένων. Στη νέα main συνάρτηση του προγράμματος, εφαρμόστηκε μέθοδος polling ώστε να ελέγχεται συνεχώς η παρουσία δεδομένων στις FIFOs της σχεδίασης του δέκτη. Μόλις η Read FIFO της πλακέτας με το περιφερειακό δέκτη αποκτήσει την πρώτη γραμμή δεδομένων εκτελείται συνάρτηση λήψης.
  - Εκτελέστηκε με επιτυχία downloading της σχεδίασης σε δυο πλακέτες XUP και αντιμετωπίστηκαν σφάλματα στη λειτουργία της σχεδίασης. Επαληθεύτηκε ότι τα δεδομένα αποστέλλονται από MGT της μιας πλακέτας σε MGT της άλλης πλακέτας και αντίστροφα μέσω καλωδίου SATA.
  - Επιβεβαιώνεται η λειτουργία μέσω χρήσης Terminal και προβολή των κατάλληλων ενδείξεων σε δυο PC
- Παρατίθενται φωτογραφίες του εν λόγω συστήματος και λεπτομερειών του:



Σχήμα 3-8 : Σύνδεση 2 πλακετών XUP

## ΚΕΦΑΛΑΙΟ 4

### ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ

Η κεντρική ιδέα με απλά λόγια είναι να δημιουργηθεί μια διεπαφή χρήστη σε γλώσσα προγραμματισμού C. Σκοπός αυτής της διεπαφής είναι να εκκινεί, να διεξάγει και να τερματίζει, όπως επίσης και να ελέγχει τη σειριακή μεταφορά δεδομένων σε πολύ υψηλές ταχύτητες.

#### 4.1 Λειτουργικότητα

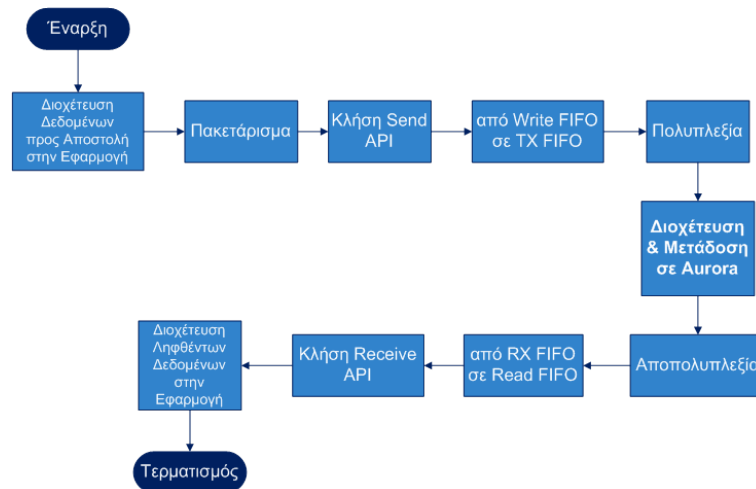
##### 4.1.1 Γενικά

Η διαδικασία της μεταφοράς δεδομένων

Η διαδικασία της μεταφοράς δεδομένων από ένα περιφερειακό Aurora σε ένα άλλο, είτε αυτό βρίσκεται στην ίδια πλακέτα είτε σε άλλη, περιλαμβάνει τα εξής βήματα:

- Βήμα 1° Διοχέτευση δεδομένων προς αποστολή στην εφαρμογή χρήστη. Κάτι τέτοιο γίνεται μέσω δείκτη στη διεύθυνση όπου βρίσκονται τα δεδομένα. Τα δεδομένα στην προκειμένη περίπτωση δημιουργούνται στην εφαρμογή χρήστη κατόπιν δυναμικής δέσμευσης μνήμης bram. Ανατίθενται σε αυτά τιμές οι οποίες αποτελούν ένα μετρητή ο οποίος αυξάνεται κατά ένα. Τονίζεται ότι υπάρχει δυνατότητα διοχέτευσης δεδομένων στην εφαρμογή χρήστη, τα οποία υπάρχουν κι έχουν οριστεί ήδη, αρκεί να παρέχεται η διεύθυνση αυτών.
- Βήμα 2° Εισαγωγή δεδομένων σε συνάρτηση του API, με όνομα createPackets, η οποία αναλαμβάνει να “πακετάρει” τα δεδομένα, δηλαδή να τα τοποθετήσει σε πλαίσια με καθορισμένο μέγιστο πλήθος bytes (μέγιστη μονάδα μετάδοσης, Maximum Transmission Unit MTU) και να ορίσει τις αντίστοιχες επικεφαλίδες πλαισίων. Επιστρέφονται, λοιπόν, τα δεδομένα από αυτήν την συνάρτηση έτοιμα για αποστολή.
- Βήμα 3° Καλείται από την εφαρμογή χρήστη, συνάρτηση του API με όνομα sendDMA ή sendPLB, στην οποία και διοχετεύεται δείκτης των δεδομένων έτοιμων για αποστολή. Η συνάρτηση αυτή αναλαμβάνει να αντιγράψει τα δεδομένα μέσω Direct Memory Access στη Write FIFO της διεπαφής περιφερειακού Aurora ή στο δίαυλο PLB και κατόπιν στη Write FIFO της διεπαφής του περιφερειακού Aurora αντίστοιχα. Τέλος αξίζει να σημειωθεί ότι η κλήση της συνάρτησης αποστολής των δεδομένων καθώς και η εφαρμογή της στο υλικό χρειάζεται πολλές δεκάδες κύκλους για να ολοκληρωθεί, πράγμα που εισάγει στο σύστημα κάποια καθυστέρηση.
- Βήμα 4° Τα δεδομένα μόλις φτάσουν στη Write FIFO του περιφερειακού Aurora μεταφέρονται ακολούθως στην TX FIFO της σχεδίασης. Ο λόγος που συμβαίνει κάτι τέτοιο είναι η διαφορά στην ταχύτητα του ρολογιού του διαύλου PLB, 100 MHz και του ρολογιού του περιφερειακού Aurora (USER\_CLK) 75MHz. Έτσι τα δεδομένα θα βγαίνουν και θα μπαίνουν

- στη Write FIFO πιο γρήγορα από ότι θα βγαίνουν και θα μπαίνουν στον πυρήνα του Aurora, πρόβλημα που επιλύθηκε με τις επιπλέον FIFOs και τα ανεξάρτητα ρολόγια ανάγνωσης κι εγγραφής που αυτές διαθέτουν.
- Βήμα 5°** Διαβάζονται δεδομένα από την TX FIFO πλάτους 64 bits. Όμως η διεπαφή του πυρήνα του Aurora έχει πλάτος 16 ή 32 bits. Για αυτό προστίθεται λογική που διαβάζει μια γραμμή δεδομένων από την TX FIFO και διοχετεύει στον πυρήνα Aurora 16 ή 32 bits δεδομένων σε διάστημα 4 ή 2 κύκλων. Επίσης, μέσω της λογικής αυτής ελέγχονται και όλα τα απαραίτητα σήματα ελέγχου του πυρήνα Aurora.
- Βήμα 6°** Τα δεδομένα διοχετεύονται στον πυρήνα του Aurora, ο οποίος αναλαμβάνει τη μετάδοσή τους μέσα από το κανάλι μετάδοσης που εγκαθιδρύει. Έτσι τα δεδομένα μεταδίδονται ανά 16 ή 32 bits σειριακά με ταχύτητες της τάξης 1 ως 3 Gbps.
- Βήμα 7°** Τα δεδομένα λαμβάνονται από τον πυρήνα του Aurora στο περιφερειακό δέκτη που έχει πλάτος 16 ή 32 bits κι έπειτα ομαδοποιούνται ξανά ανά 64 bits σε διάστημα 4 ή 2 κύκλων ώστε να αποθηκευτούν στην RX FIFO. Επίσης ελέγχονται και όλα τα απαραίτητα σήματα ελέγχου του πυρήνα Aurora.
- Βήμα 8°** Εκτελείται μεταφορά των δεδομένων στην RX FIFO της σχεδίασης κι ακολούθως στη Read FIFO της διεπαφής του περιφερειακού. Ο λόγος που συμβαίνει κάτι τέτοιο είναι η διαφορά στην ταχύτητα του ρολογιού του διαύλου PLB 100 MHz και του ρολογιού του περιφερειακού Aurora (USER\_CLK) 75MHz. Έτσι τα δεδομένα θα βγαίνουν και θα μπαίνουν στη Read FIFO πιο γρήγορα από ότι θα βγαίνουν και θα μπαίνουν στον πυρήνα του Aurora, πρόβλημα που επιλύθηκε με τις επιπλέον FIFOs και τα ανεξάρτητα ρολόγια ανάγνωσης κι εγγραφής που αυτές διαθέτουν.
- Βήμα 9°** Καλείται από την εφαρμογή χρήστη, συνάρτηση του API με όνομα receiveDMA ή receivePLB, στην οποία και διοχετεύεται δείκτης διεύθυνσης έτοιμης να δεχθεί τα ληφθέντα δεδομένα. Η συνάρτηση αυτή αναλαμβάνει να αντιγράψει τα δεδομένα από τη Read FIFO της διεπαφής περιφερειακού Aurora μέσω Direct Memory Access ή μέσω διαύλου PLB προς τη διεύθυνση που παρέχεται από την εφαρμογή χρήστη. Τέλος αξίζει να σημειωθεί ότι η κλήση της συνάρτησης λήψης των δεδομένων καθώς και η εφαρμογή της στο υλικό χρειάζεται πολλές δεκάδες κύκλους για να ολοκληρωθεί, πράγμα που εισάγει στο σύστημα κάποια καθυστέρηση.
- Βήμα 10°** Διοχέτευση ληφθέντων δεδομένων στην εφαρμογή χρήστη. Κάτι τέτοιο γίνεται μέσω δείκτη στη διεύθυνση όπου μπορούν να αποθηκευτούν τα δεδομένα. Στην προκειμένη περίπτωση, στην εφαρμογή χρήστη εκτελείται δυναμική δέσμευση μνήμης bram. Και καταχωρούνται σε αυτήν τα ληφθέντα δεδομένα. Τονίζεται ότι υπάρχει δυνατότητα διοχέτευσης δεδομένων στην εφαρμογή χρήστη μέσω οποιας διεύθυνσης είναι διαθέσιμη για αποθήκευση των δεδομένων αυτών



Σχήμα 4.1 : Διάγραμμα Ροής Λειτουργικότητας

### Η Εφαρμογή Χρήστη

Η εφαρμογή φορτώνεται στην instruction memory του επεξεργαστή κι εκτελείται από τον ίδιο τον επεξεργαστή PowerPC της FPGA. Το λογισμικό της υλοποίησής μας αποτελείται από δύο τμήματα. Το τμήμα της διεπαφής με το χρήστη που υλοποιείται σαν βιβλιοθήκη και το τμήμα της κυρίως εφαρμογής. Η κυρίως λειτουργικότητα, λοιπόν, της εφαρμογής περιέχεται στη διεπαφή χρήστη και ο κυρίως κορμός της εφαρμογής (συνάρτηση main) χρησιμοποιείται περισσότερο για την δοκιμή της λειτουργικότητας και αλλάζει ανάλογα με τις ανάγκες των εργαστηριακών δοκιμών. Υλοποιείται η παρακάτω λειτουργικότητα:

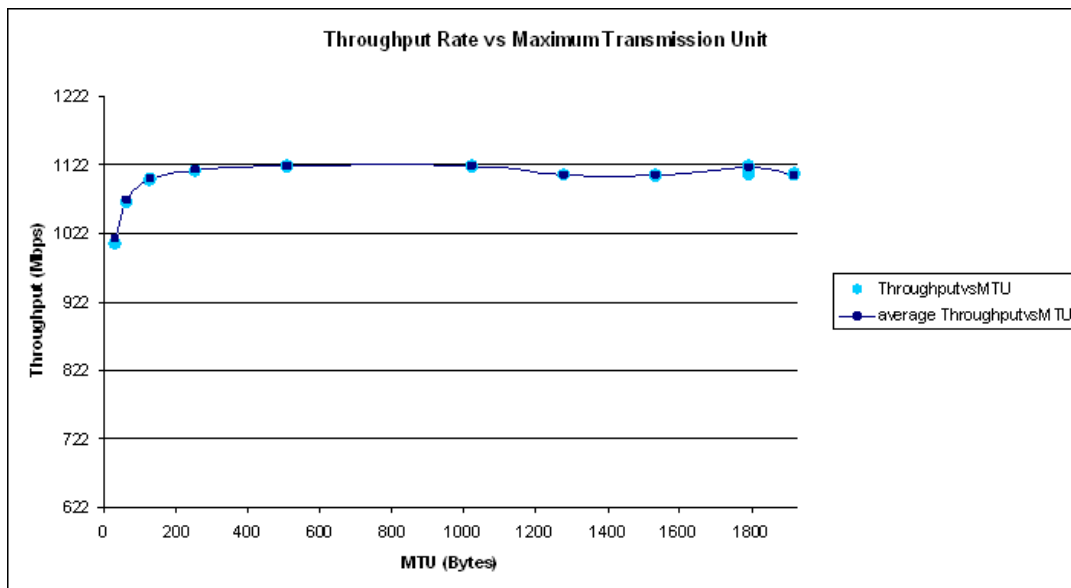
- Παρέχεται ένα API (Application Program Interface) μέσω συνάρτησης για αποστολή λήψη δεδομένων είτε με DMA (Direct Memory Access) είτε μέσω του διαύλου PLB. Πρόκειται για συναρτήσεις send/receive που δέχονται ως ορίσματα ένα δείκτη τύπου Xuint προς τα δεδομένα προς αποστολή/λήψη και μια μεταβλητή τύπου int που δηλώνει το μέγεθος των δεδομένων προς αποστολή/λήψη. Η ακριβής υλοποίηση του API κρύβεται από το χρήστη και προστατεύεται ακόμα από εσφαλμένη τροποποίησή της. Αλλά κυρίως παρέχεται πλήρης πρόσβαση στη λειτουργικότητα της εφαρμογής γρήγορα κι εύκολα με interface δύο απλών συναρτήσεων χωρίς να χρειάζεται ο χρήστης να εισέλθει εις βάθος σε λεπτομέρειες υλοποίησης.
- Επαληθεύεται η μεταφορά δεδομένων χωρίς σφάλματα στο δεύτερο περιφερειακό στην ίδια πλακέτα συγκρίνοντας τα δεδομένα που εστάλησαν και τα ληφθέντα δεδομένα. Κάτι τέτοιο μπορεί να απαιτεί ύπαρξη ειδικών λειτουργιών στον πυρήνα Aurora και συνεπώς και την προσθήκη λογικής επιπέδου HDL. Σε αυτήν τη λογική ελέγχεται και η πληρότητα των FIFOs κι αναλόγως γίνεται η διαχείριση της ροής δεδομένων. Υπάρχει η δυνατότητα εκτέλεσης τεχνικών polling ή interrupts για αυτό το σκοπό.
- Γίνεται επισκόπηση της κατάστασης των δύο FIFO Read και Write ώστε να ικανοποιούνται οι συνθήκες για ομαλή λειτουργία της σχεδίασης.
- Εκτελούνται οι απαραίτητοι έλεγχοι ώστε να εξασφαλιστεί η απρόσκοπτη αποστολή ή λήψη ενός σημαντικού σε όγκο δεδομένων.
- Πραγματοποιείται μέτρηση πραγματικού ρυθμού μετάδοσης δεδομένων



## 4.2 Προδιαγραφές πρωτοκόλλου

Στόχος η αναγνώριση πακέτων πληροφορίας ώστε να υπάρχει η δυνατότητα σε ανώτερο επίπεδο να ελέγχεται η ακεραιότητα του πακέτου που παραλήφθηκε και σε περίπτωση σφάλματος αναλόγως τη διενέργεια επανάληψης της αποστολής του ή ακύρωσης της λήψης του. Αναπτύχθηκαν, λοιπόν, κάποιοι θεμελιώδεις κανόνες για την εύρυθμη λειτουργία της σχεδίασης.

- Κάθε πακέτο που αποστέλλεται έχει μια επικεφαλίδα η οποία δηλώνει το μέγεθος σε bytes του πακέτου αυτού.
- Ως επικεφαλίδα ορίζεται η πρώτη λέξη του PDU που αποστέλλεται μήκους 2 bytes και προσμετράται στο πλήθος των bytes που αναγράφονται σε αυτήν ως μέγεθος του πακέτου.
- Το μέγεθος του πακέτου πρέπει να είναι αριθμός πολλαπλάσιος του 8. Αυτό αποτελεί σύμβαση που γίνεται λόγω του γεγονότος ότι όταν εκτελείται η μεταφορά σε λειτουργία DMA τα δεδομένα μεταφέρονται υποχρεωτικά ανά 8 bytes.
- Ορίζεται η σταθερά MTU (Maximum Transmission Unit) που αποτελεί το μέγιστος μέγεθος σε bytes του πακέτου. Στη συγκεκριμένη σχεδίαση και κατόπιν μελέτης του MTU ως προς το ρυθμό throughput, όπως φαίνεται στο Σχήμα 4.2, ορίζεται το MTU στα 256 bytes.



Σχήμα 4.2 : Μελέτη Μεταβολής στο Throughput ανάλογα με το MTU πακέτου

- Έχει προβλεφθεί και άτυπα ένα ελάχιστο μέγεθος πακέτου το οποίο είναι 5 bytes. Κάτι τέτοιο γίνεται για λόγους αποδοτικότητας καθώς υπάρχει η δυνατότητα αποστολής πακέτων μέχρι και 1 byte όμως πρόκειται για μια λειτουργία περιττή στη γενική περίπτωση.
- Κατά τη διάρκεια της πολύπλεξης των δεδομένων του πακέτου του χρήστη υπάρχει ένα σήμα που μετρά τον αριθμό των bytes που έχουν αποσταλεί και αναλόγως αναλαμβάνει την ενεργοποίηση των σημάτων SOF\_N και EOF\_N στην αρχή ή στο τέλος του μετρήματος αντίστοιχα. Επίσης ρυθμίζεται και το σήμα REM στο τέλος του πακέτου (μαζί με το EOF\_N) αναλόγως του πλήθους των δεδομένων που αποστέλλονται.

- Τέλος ορίζεται συνάρτηση η οποία λαμβάνει τα δεδομένα του χρήστη και αναλαμβάνει τη δημιουργία πακέτων δεδομένων με τον υπολογισμό των επικεφαλίδων του κάθε πακέτου και ενσωμάτωσή τους σε αυτό.

## 4.3 Βιβλιοθήκη API

### 4.3.1 Διεπαφή με το χρήστη

- Παρέχεται ένα API (Application Program Interface) μέσω συνάρτησης για αποστολή δεδομένων με DMA (Direct Memory Access). Πρόκειται για τη συνάρτηση `int sendDMA (Xuint16, int size)`. Αυτή η συνάρτηση δέχεται ως ορίσματα ένα δείκτη τύπου `Xuint16` προς τα δεδομένα προς αποστολή και μια μεταβλητή τύπου `int` που δηλώνει το μέγεθος των δεδομένων προς αποστολή. Κατ' αρχάς ελέγχεται αν η Write FIFO έχει αρκετό χώρο διαθέσιμο για την αποθήκευση όλων των δεδομένων που διοχετεύονται στη συνάρτηση. Κι αυτό γιατί είναι σκόπιμο να μεταφερθούν όλα τα δεδομένα με μια κλήση της συνάρτησης ώστε να ελαχιστοποιηθεί η καθυστέρηση που εισέρχεται στο σύστημα λόγω αυτής (βλέπε ενότητα “Λειτουργικότητα”, σελ. 86). Στη συνέχεια η συνάρτηση επανεκκινεί τη DMA:

```
AURORA_PLB_mResetDMA0(aurora_plb_0_baseaddr)
```

όπου `aurora_plb_0_baseaddr` στιγμαίотυπο του περιφερειακού Aurora πομπού που ορίζεται στη διεύθυνση βάσης `XPAR_AURORA_PLB_0_BASEADDR`, που ανατίθεται από το υλικό. Η συνάρτηση θέτει τη μάσκα ελέγχου για DMA:

```
AURORA_PLB_mSetDMA0Control(aurora_plb_0_baseaddr,  
DMA_SINC_MASK | DMA_DLOCAL_MASK);
```

όπου η δεύτερη παράμετρος ορίζει ότι η διεύθυνση πηγής ανήκει στη μνήμη, ενώ η τρίτη παράμετρος ορίζει ότι η διεύθυνση προορισμού βρίσκεται τοπικά στο περιφερειακό. Στη συνέχεια αρχίζει η μεταφορά των δεδομένων στη WriteFIFO της διεπαφής του περιφερειακού, για το σύνολο των δεδομένων που έχει δεχθεί ως είσοδο η συνάρτηση:

```
AURORA_PLB_DMA0Transfer(aurora_plb_0_baseaddr, (Xuint32)data,  
aurora_plb_0_baseaddr+AURORA_PLB_WRFIFO_DATA_OFFSET,  
data_byte_len);
```

όπου το δεύτερο όρισμα δηλώνει τη διεύθυνση πηγής, το τρίτο όρισμα δηλώνει τη διεύθυνση προορισμού, στην προκειμένη περίπτωση συντίθεται από τη διεύθυνση βάσης αυξημένη κατά την παράμετρο που δίδεται από το σύστημα ως προς τη σχετική θέση της Write FIFO μες στο περιφερειακό. Έπειτα αναμένεται να ολοκληρωθεί η DMA μεταφορά δεδομένων. Στη συνέχεια γίνεται έλεγχος για τυχόν σφάλματα στη μεταφορά DMA:

```
if(AURORA_PLB_mDMA0Error(aurora_plb_0_baseaddr))  
xil_printf("Error writing to AURORA_PLB_0");
```

Κατόπιν, αναλαμβάνει το υλικό. Έτσι τα δεδομένα από τη Write FIFO θα μεταφερθούν στην TX FIFO και από εκεί κατόπιν πολύπλεξης στον πυρήνα Aurora ώστε να σταλούν μέσω του MGT. Τέλος τυπώνονται στο terminal τα δεδομένα που εστάλισαν με χρήση της συνάρτησης `xil_printf()` που είναι πιο γρήγορη από την συνηθισμένη συνάρτηση `printf()`.

- Παρέχεται ένα API (Application Program Interface) μέσω συνάρτησης για λήψη δεδομένων με DMA (Direct Memory Access). Πρόκειται για τη συνάρτηση `int receiveDMA (Xuint16, int size)`. Κατ' αρχάς ελέγχεται αν η Read FIFO έχει δεδομένα διαθέσιμα για ανάγνωση και δεν είναι άδεια. Έπειτα αναλαμβάνεται η

μεταφορά όλων των δεδομένων της Read FIFO. Κι αυτό γιατί είναι σκόπιμο να μεταφερθούν όλα τα δεδομένα με μια κλήση της συνάρτησης ώστε να ελαχιστοποιηθεί η καθυστέρηση που εισέρχεται στο σύστημα λόγω αυτής (βλέπε ενότητα “Λειτουργικότητα”, σελ. 86). Αυτή η συνάρτηση δέχεται ως ορίσματα ένα δείκτη τύπου `Xuint16` που θα δεχτεί τα ληφθέντα δεδομένα και μια μεταβλητή τύπου `int` που δηλώνει το μέγεθος των δεδομένων προς λήψη. Στη συνέχεια η συνάρτηση επανεκκινεί τη DMA:

```
AURORA_PLB_mResetDMA0(aurora_plb_1_baseaddr)
```

όπου `aurora_plb_1_baseaddr` είναι στιγμιότυπο του περιφερειακού Aurora δέκτη που ορίζεται στη διεύθυνση βάσης `XPAR_AURORA_PLB_1_BASEADDR`, που ανατίθεται από το υλικό. Η συνάρτηση θέτει τη μάσκα ελέγχου για DMA:

```
AURORA_PLB_mSetDMA0Control(aurora_plb_1_baseaddr,  
DMA_DINC_MASK | DMA_SLOCAL_MASK);
```

όπου η δεύτερη παράμετρος ορίζει ότι η διεύθυνση προορισμού ανήκει στη μνήμη, ενώ η τρίτη παράμετρος ορίζει ότι η διεύθυνση πηγής βρίσκεται τοπικά στο περιφερειακό. Στη συνέχεια αρχίζει η μεταφορά των δεδομένων από την Read FIFO της διεπαφής του περιφερειακού, για το σύνολο των δεδομένων που έχει δεχθεί ως είσοδο η συνάρτηση:

```
AURORA_PLB_DMA0Transfer(aurora_plb_1_baseaddr,  
aurora_plb_1_baseaddr+AURORA_PLB_RDFIFO_DATA_OFFSET,  
(Xuint32)data, data_byte_len);
```

όπου το δεύτερο όρισμα δηλώνει τη διεύθυνση πηγής, στην προκειμένη περίπτωση συντίθεται από τη διεύθυνση βάσης αυξημένη κατά την παράμετρο που δίδεται από το σύστημα ως προς τη σχετική θέση της Read FIFO μες στο περιφερειακό, το τρίτο όρισμα δηλώνει τη διεύθυνση προορισμού. Έπειτα αναμένει να ολοκληρωθεί η DMA μεταφορά δεδομένων. Στη συνέχεια γίνεται έλεγχος για τυχόν σφάλματα στη μεταφορά DMA:

```
if(AURORA_PLB_mDma0Error(aurora_plb_1_baseaddr))  
xil_printf("Error reading from AURORA_PLB_1");
```

Μετά, αναλαμβάνει το υλικό. Έτσι τα δεδομένα από τη Read FIFO θα μεταφερθούν στην RX FIFO και από εκεί κατόπιν αποπολύπλεξης στο χρήστη και πιο συγκεκριμένα στη μνήμη του συστήματός μας `bram`. Τέλος τυπώνονται στο terminal τα δεδομένα που εστάλησαν με χρήση της συνάρτησης `xil_printf()` που είναι πιο γρήγορη από την συνηθισμένη συνάρτηση `printf()`.

- Παρέχεται ένα API (Application Program Interface) μέσω συνάρτησης για αποστολή δεδομένων μέσω του διαύλου PLB. Πρόκειται για τη συνάρτηση `int sendPLB(Xuint16, int size)`. Αυτή η συνάρτηση δέχεται ως ορίσματα ένα δείκτη τύπου `Xuint16` προς τα δεδομένα προς αποστολή και μια μεταβλητή τύπου `int` που δηλώνει το μέγεθος των δεδομένων προς αποστολή. Κατ’ αρχάς ελέγχεται αν η Write FIFO έχει αρκετό χώρο διαθέσιμο για την αποθήκευση δεδομένων μεγέθους 64 bits από αυτά που διοχετεύονται στη συνάρτηση. Κι αυτό γιατί τα δεδομένα, μεταφέρονται ανά 8 bytes στο περιφερειακό με κλήση της συνάρτησης `AURORA_PLB_WriteToFIFO(aurora_plb_0_baseaddr, data64);`. Έτσι, αρχίζει η μεταφορά δεδομένων προς τη WriteFIFO της διεπαφής του περιφερειακού μέσω του διαύλου PLB, για το σύνολο των δεδομένων που έχει δεχθεί ως είσοδο η συνάρτηση. Στη συνέχεια, τα δεδομένα από τη Write FIFO θα μεταφερθούν στην TX FIFO κι από εκεί στον πυρήνα Aurora ώστε να σταλούν μέσω του MGT. Τέλος, τυπώνονται στο terminal τα δεδομένα που εστάλησαν.
- Παρέχεται ένα API (Application Program Interface) μέσω συνάρτησης για αποστολή δεδομένων μέσω του διαύλου PLB. Πρόκειται για τη συνάρτηση `int`

receivePLB (Xuint16, int size). Αυτή η συνάρτηση δέχεται ως ορίσματα ένα δείκτη τύπου Xuint16 που θα δεχτεί τα ληφθέντα δεδομένα και μια μεταβλητή τύπου int που δηλώνει το μέγεθος των δεδομένων προς λήψη. Κατ' αρχάς, ελέγχεται αν η Read FIFO έχει διαθέσιμα δεδομένα για ανάγνωση. Τα δεδομένα, μεταφέρονται ανά 8 bytes από το περιφερειακό με κλήση της συνάρτησης:

```
AURORA_PLB_ReadFromFIFO(aurora_plb_0_baseaddr, data64);
```

Έτσι, αρχίζει η μεταφορά δεδομένων από τη Read FIFO της διεπαφής του περιφερειακού μέσω του διαύλου PLB, για το σύνολο των δεδομένων που έχει δεχθεί ως είσοδο η συνάρτηση. Στη συνέχεια, τα δεδομένα από τη Read FIFO θα μεταφερθούν στην RX FIFO κι από εκεί, κατόπιν αποπολύπλεξης, στο χρήστη και πιο συγκεκριμένα στη μνήμη του συστήματος μας bram. Τέλος, τυπώνονται στο terminal τα ληφθέντα δεδομένα.

- Παρέχεται συνάρτηση για δημιουργία πακέτων από ακολουθία δεδομένων. Πρόκειται για τη συνάρτηση Xuint16 \*createPackets (int \*,Xuint16\*). Η συνάρτηση δέχεται ως ορίσματα ένα δείκτη προς τα δεδομένα χρήστη και τον αριθμό των δεδομένων αυτών. Επιστρέφει δείκτη προς τα νέα δεδομένα που περιλαμβάνουν τα εισαχθέντα δεδομένα χρήστη μέσα σε πλαίσια με τις αντίστοιχες επικεφαλίδες τους.



Σχήμα 4.3 : Μέθοδος για το “Πακετάρισμα” Δεδομένων

## 4.4 Main (Κυρίως εφαρμογή)

### 4.4.1 Για loopback σε μια πλακέτα XUP

- Εκτελείται έλεγχος της κατάστασης των δύο FIFO Read και Write ώστε να ικανοποιούνται συνθήκες. Συγκεκριμένα ελέγχονται τα σήματα empty και Occupancy της Read FIFO που δείχνουν αν είναι άδεια ή πόσες θέσεις της είναι κατειλημμένες. Ακόμα ελέγχονται τα σήματα full και Vacancy της Write FIFO που δείχνουν αν είναι γεμάτη ή το πλήθος των άδειων θέσεων που διαθέτει.
- Εκτελούνται οι απαραίτητοι έλεγχοι ώστε να εξασφαλιστεί η απρόσκοπτη αποστολή ή λήψη σημαντικών, σε όγκο, δεδομένων.
- Υπάρχει δυνατότητα επαλήθευσης της μεταφοράς δεδομένων χωρίς σφάλματα στο δεύτερο περιφερειακό στην ίδια πλακέτα συγκρίνοντας τα δεδομένα που

εστάλησαν και τα ληφθέντα δεδομένα. Κάτι τέτοιο δεν απαιτεί ύπαρξη ειδικών λειτουργιών στον πυρήνα AuToRa και συνεπώς την προσθήκη λογικής επιπέδου HDL.

- Πραγματοποιείται μέτρηση πραγματικού ρυθμού μετάδοσης δεδομένων μέσω δομής για αποθήκευση χρόνου και συναρτήσεων για την επικοινωνία με ειδικό περιφερειακό timer συνδεδεμένο στο δίαυλο opb. Η δομή αυτή ορίζεται στο λογισμικό ως εξής:

```
XTmrCtr T;
```

η οποία αρχικοποιείται:

```
void clockstart() {
    XTmrCtr_Initialize(&T, XPAR_OPB_TIMER_0_DEVICE_ID);
    XTmrCtr_Reset(&T, 0);
    XTmrCtr_Start(&T, 0);
}
```

Για το σκοπό αυτό είναι σημαντική η τοποθέτηση συναρτήσεων

```
XTmrCtr_GetValue(&T, 0);
```

στα κατάλληλα σημεία στον κώδικα για ανάκτηση τιμής του περιφερειακού timer που δηλώνει το χρόνο εκτέλεσης του προγράμματος χρήστη σε κύκλους του επεξεργαστή PowerPC κατόπιν διενέργειας ελέγχων συνθηκών πληρότητας των FIFOs της διεπαφής του περιφερειακού AuToRa.

- Ακόμα πραγματοποιείται μέτρηση πραγματικού ρυθμού μετάδοσης δεδομένων με αυτοσχέδιο περιφερειακό που επικοινωνεί με τον επεξεργαστή μέσω του διαύλου PLB σε συνδεσμολογία σκλάβου (slave). Διαθέτει έναν καταχωρητή μήκους 64 bits που χρησιμεύει στην αποθήκευση δύο τιμών μετρητών. Εκτελείται ανάγνωση του καταχωρητή αυτού με τη συνάρτηση:

```
THROUGHPUT_TIMER_ReadSlaveReg0(my_timer_p, &Reg64Value);
```

Στα 32 λιγότερο σημαντικά bits του καταχωρητή αποθηκεύεται η τιμή του μετρητή που συμβολίζει το λανθάνων χρόνο μετάδοσης για 64 bits δεδομένων. Στα 32 πιο σημαντικά bits του καταχωρητή αποθηκεύεται η τιμή του μετρητή που συμβολίζει το συνολικό χρόνο μετάδοσης όλων των δεδομένων. Τέλος υπολογίστηκαν ο ρυθμός throughput της μετάδοσης των δεδομένων του συστήματος καθώς και ο λανθάνων ρυθμός μετάδοσης των δεδομένων του συστήματος με τους παρακάτω μαθηματικούς τύπους:

```
throughput_time = (data_byte_len*8)/
((Reg64Value.Upper+1)*clk_period)*1000;
latency_time = (8 * 8)/
(Reg64Value.Lower*clk_period)*1000;
```

όπου

```
float clk_period = 13.333; /*clk_frequency = 75 Mhz*/
```

#### 4.4.2 Για μεταφορά δεδομένων από μια πλακέτα XUP σε μια άλλη

- Εκτελείται polling, μέσω κώδικα γλώσσας C και εντολής loop while(1), για τον συνεχή περιοδικό έλεγχο της ολοκλήρωσης της μετάδοσης δεδομένων από τη μια πλακέτα στην άλλη καθώς και για την επαλήθευση της αξιοπιστίας μεταφοράς τους. Υπάρχει δυνατότητα υλοποίησης των παραπάνω λειτουργιών μέσω interrupts.
- Εκτελείται έλεγχος της κατάστασης των δύο FIFO Read και Write ώστε να ικανοποιούνται οι συνθήκες. Συγκεκριμένα ελέγχονται τα σήματα empty και Occupancy της Read FIFO που δείχνουν αν είναι άδεια ή πόσες θέσεις της είναι

κατελημμένες. Ακόμα ελέγχονται τα σήματα full και Vacancy της Write FIFO που δείχνουν αν είναι γεμάτη ή το πλήθος των άδειων θέσεων που διαθέτει.

- Εκτελούνται οι απαραίτητοι έλεγχοι ώστε να εξασφαλιστεί η απρόσκοπτη αποστολή ή λήψη ενός σημαντικών, σε όγκο, δεδομένων.
- Υπάρχει δυνατότητα επαλήθευσης της μεταφοράς δεδομένων χωρίς σφάλματα στο δεύτερο περιφερειακό στην πλακέτα προορισμού, συγκρίνοντας τα δεδομένα που εστάλησαν και τα ληφθέντα δεδομένα. Κάτι τέτοιο δεν απαιτεί ύπαρξη ειδικών λειτουργιών στον πυρήνα Auora και συνεπώς την προσθήκη λογικής επιπέδου HDL, αν και υπάρχει δυνατότητα να πραγματοποιηθεί σύγκριση ορθότητας δεδομένων στο υλικό.
- Πραγματοποιείται μέτρηση πραγματικού ρυθμού μετάδοσης δεδομένων μέσω δομής για αποθήκευση χρόνου και συναρτήσεων για την επικοινωνία με ειδικό περιφερειακό timer συνδεδεμένο στο δίαυλο opb. Η δομή αυτή ορίζεται στο λογισμικό ως εξής:

```
XTmrCtr T;
```

η οποία αρχικοποιείται:

```
void clockstart() {
    XTmrCtr_Initialize(&T, XPAR_OPB_TIMER_0_DEVICE_ID);
    XTmrCtr_Reset(&T, 0);
    XTmrCtr_Start(&T, 0);
}
```

Για το σκοπό αυτό είναι σημαντική η τοποθέτηση συναρτήσεων

```
XTmrCtr_GetValue(&T, 0);
```

στα κατάλληλα σημεία στον κώδικα και διενέργεια ελέγχου συνθηκών πληρότητας των FIFOs της διεπαφής του περιφερειακού Auora για ανάκτηση τιμής του περιφερειακού timer που δηλώνει το χρόνο εκτέλεσης του προγράμματος χρήστη σε κύκλους του επεξεργαστή PowerPC.



## ΚΕΦΑΛΑΙΟ 5

### ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ ΣΧΕΔΙΑΣΗΣ

#### 5.1 Προαπαιτούμενα και προετοιμασία για το πείραμα

##### 5.1.1 Προϋποθέσεις

Πέρα από τις προφανείς προϋποθέσεις, το λογισμικό XPS και την πλακέτα XUPV2P, αυτή η εργασία προϋποθέτει τα ακόλουθα:

Καλώδιο SATA: Εάν χρησιμοποιείται η πλακέτα XUPV2P, ένα καλώδιο SATA είναι απαραίτητο για την λειτουργία της σχεδίασης. Εάν δεν γίνεται χρήση της πλακέτας XUPV2P, πρέπει να υπάρχει κάποιος τρόπος να συνδεθούν φυσικά δυο MGTs. Στην προκειμένη περίπτωση έγινε χρήση καλωδίου SATA II ταχύτητας μέχρι 3.0 Gbps (ρυθμός Baud) και μήκους 40 cm.

CORE Generator: Και τα δυο, ο πυρήνας Aurora και οι TX/RX FIFO θα παραχθούν με τον CORE Generator. Χρειάζεται να γίνει εγγραφή στη Xilinx για την απόκτηση πιστοποιητικού χρήστη του πυρήνα Aurora. Είναι μια δωρεάν και απλή διαδικασία που περιλαμβάνει αποδοχή της συμφωνίας πιστοποιητικού μετά από την οποία αποστέλλεται ένα πιστοποιητικό με οδηγίες ξεκλειδώματος του πυρήνα Aurora στον CORE Generator.

#### 5.2 Δοκιμές αναλυτικά

##### 5.2.1 Έλεγχος ορθότητας λειτουργίας σημάτων κατάστασης

Με σκοπό τον έλεγχο της ορθότητας λειτουργίας του περιφερειακού Aurora μέσω των σημάτων κατάστασής του, συνδέονται τα συγκεκριμένα σήματα εξόδου του περιφερειακού Channel\_up, Hard\_error, soft\_error, Lane\_up στις φωτοδιόδους (LEDs) της πλακέτας XUP Virtex II Pro. Έτσι οι φωτοδιόδοι (LEDs) απεικονίζουν τις εξόδους κατάστασης:

LED 0 - CHANNEL_UP,	## Το κανάλι είναι σε λειτουργία
LED 1 - HARD_ERROR,	## Εντοπίστηκε σφάλμα στη λειτουργία του υλικού του RocketIO πομποδέκτη
LED 2 - SOFT_ERROR,	## Εντοπίστηκε σφάλμα στη μετάδοση ενός ή πολλών bit δεδομένων
LED 3 - LANE_UP.	## Η γραμμή είναι σε λειτουργία

Σημειώτεον ότι λόγω της καλωδίωσης των φωτοδιόδων (LEDs) στο XUPV2P, ένα λογικό υψηλό ("1") θέτει τη φωτοδίοδο (LED) στο OFF, ενώ ένα λογικό χαμηλό ("0") θέτει τη φωτοδίοδο (LED) στο ON. Στη συνέχεια εφόσον αποσυνδεθεί το καλώδιο SATA παρατηρείται με ποιο τρόπο αλλάζουν οι εξοδοι κατάστασης. Οι CHANNEL\_UP και LANE\_UP φωτοδιόδοι (LEDs) θα πρέπει να τεθούν αυτόματα στο ON. Αν γίνει επανασύνδεση του καλωδίου SATA, το κανάλι θα πρέπει να εγκαθιδρύεται ξανά και οι

CHANNEL\_UP και LANE\_UP φωτοδίοδοι (LEDs) θα πρέπει να τίθενται αυτομάτως στο OFF. Τέλος, εκτελείται έλεγχος ορθότητας λειτουργίας του συστήματος με κλήση εντολών της εφαρμογής χρήστη, που τυπώνουν στην ροή εξόδου πληροφορίες. Τέτοιες πληροφορίες είναι τα δεδομένα που αποστέλλονται όπως και τα δεδομένα που παραλαμβάνονται. Έτσι μεταφέρονται μέσω της σειριακής θύρας της πλακέτας στο terminal του προγράμματος Tera Term Pro οι αντίστοιχες πληροφορίες που επιβεβαιώνουν την ορθή λειτουργία του συστήματος.

```

HyperTerminal
File Edit View Call Transfer Help

Test 1: Transmit data from AURORA_PLB_0 to AURORA_PLB_1

Writing data to the AURORA_PLB_0 peripheral:
0x1111 0x2222 0x3333 0x4444 0x5555 0x6666 0x7777 0x8888 0x9999 0xA000

Reading data from the AURORA_PLB_1 peripheral:
0x1111 0x2222 0x3333 0x4444 0x5555 0x6666 0x7777 0x8888 0x9999 0xA000

Test 2: Transmit data from AURORA_PLB_1 to AURORA_PLB_0

Writing data to the AURORA_PLB_1 peripheral:
0x1111 0x2222 0x3333 0x4444 0x5555 0x6666 0x7777 0x8888 0x9999 0xA000

Reading data from the AURORA_PLB_0 peripheral:
0x1111 0x2222 0x3333 0x4444 0x5555 0x6666 0x7777 0x8888 0x9999 0xA000

Virtex-II Pro Resource http://virtex2pro.blogspot.com

Connected 1.79 KB Auto default 9600 Baud SERIAL CAPS NUM Capture Print Help

```

Σχήμα 5-1 : Έξοδος της Σχεδίασης στο Terminal

### 5.2.2 Χαρακτηριστικά διαφορετικών σχεδιάσεων

Έχουν υλοποιηθεί 6 διαφορετικές σχεδιάσεις του συστήματός μας.

- **Aurora σε λειτουργία Streaming απλή έκδοση** : Πρόκειται για την πιο απλή σχεδίαση του συστήματος όπου το περιφερειακό διαθέτει τον πυρήνα Aurora με τις λιγότερες θύρες για Streaming mode.
  - Ο πυρήνας είναι σε λειτουργία Streaming
  - Διαθέτει διεπαφή με μήκος λέξης 2 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-2
  - Δεν υπάρχει instruction ούτε data cache
  - Έχει δημιουργηθεί BRAM μεγέθους 64 KB
  - Περιλαμβάνονται 4 FIFOs 64 x 512 (πλάτος x βάθος)
  - Τα μηνύματα που δίνει ως εξόδο η εφαρμογή τυπώνονται στο terminal μέσω του OPB Uart.
  - Χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά.
- **Aurora σε λειτουργία Streaming προχωρημένη έκδοση** : Πρόκειται για την πιο προχωρημένη σχεδίαση του συστήματος όπου το περιφερειακό διαθέτει τον πυρήνα Aurora με τις περισσότερες θύρες κι επιλογές για Streaming mode.
  - Ο πυρήνας είναι σε λειτουργία Streaming
  - Διαθέτει διεπαφή με μήκος λέξης 4 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-3
  - Υπάρχει instruction και data cache

- Έχει δημιουργηθεί BRAM μεγέθους 128 KB που είναι και το μέγιστο δυνατό μέγεθος για BRAM.
- Περιλαμβάνονται 4 FIFOs 64 x 1024 (πλάτος x βάθος)
- Τα μηνύματα που δίνει ως έξοδο η εφαρμογή τυπώνονται στο terminal μέσω του PLBUart16550.
- Δεν χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά, αντιθέτως όλα είναι συνδεδεμένα στο δίαυλο PLB.
- **Aurora σε λειτουργία Streaming έκδοση μέτρησης throughput:** Πρόκειται για την σχεδίαση του συστήματος για τη μέτρηση throughput με προσθήκη ενός επιπλέον περιφερειακού και επιπλέον λογικής.
  - Ο πυρήνας είναι σε λειτουργία Streaming
  - Διαθέτει διεπαφή με μήκος λέξης 2 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-3
  - Υπάρχει instruction και data cache
  - Έχει δημιουργηθεί BRAM μεγέθους 128 KB που είναι και το μέγιστο δυνατό μέγεθος για BRAM.
  - Περιλαμβάνονται 4 FIFOs 64 x 512 (πλάτος x βάθος)
  - Τα μηνύματα που δίνει ως έξοδο η εφαρμογή τυπώνονται στο terminal μέσω του PLBUart16550.
  - Δεν χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά, αντιθέτως όλα είναι συνδεδεμένα στο δίαυλο PLB.
- **Aurora σε λειτουργία Packet απλή έκδοση :** Πρόκειται για την πιο απλή σχεδίαση σε λειτουργία packet του συστήματος όπου το περιφερειακό διαθέτει τον πυρήνα Aurora με τις λιγότερες θύρες για packet mode.
  - Ο πυρήνας είναι σε λειτουργία Packet
  - Δεν χρησιμοποιείται κανένα επιπλέον χαρακτηριστικό του Aurora core
  - Διαθέτει διεπαφή με μήκος λέξης 2 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-2
  - Δεν υπάρχει instruction ούτε data cache
  - Έχει δημιουργηθεί BRAM μεγέθους 64 KB
  - Περιλαμβάνονται 4 FIFOs 64 x 512 (πλάτος x βάθος)
  - Τα μηνύματα που δίνει ως έξοδο η εφαρμογή τυπώνονται στο terminal μέσω του OPB Uart.
  - Χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά.
- **Aurora σε λειτουργία Packet προχωρημένη έκδοση :** Πρόκειται για την πιο προχωρημένη σχεδίαση σε λειτουργία packet του συστήματος όπου το περιφερειακό διαθέτει τον πυρήνα Aurora με τις περισσότερες θύρες κι επιλογές.
  - Ο πυρήνας είναι σε λειτουργία Packet
  - Χρησιμοποιείται ο τοπικός έλεγχος ροής (native flow control) ώστε να διακόπτει την αποστολή δεδομένων όταν η RX FIFO του παραλήπτη είναι σχεδόν γεμάτη.
  - Διαθέτει διεπαφή με μήκος λέξης 4 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-3
  - Υπάρχει instruction και data cache
  - Έχει δημιουργηθεί BRAM μεγέθους 128 KB που είναι και το μέγιστο δυνατό μέγεθος για BRAM.

- Περιλαμβάνονται 4 ασύμμετρες FIFOs δυο 64 x 1024 (πλάτος x βάθος) και δυο 64 x 512 (πλάτος x βάθος). Πιο συγκεκριμένα έχουν μεγαλώσει οι FIFOs που έχουν το μεγαλύτερο κίνδυνο αν γεμίσουν με δεδομένα να προκαλέσουν καθυστέρηση στο σύστημα. Αυτές είναι οι Write FIFO και RX FIFO.
- Τα μηνύματα που δίνει ως έξοδο η εφαρμογή τυπώνονται στο terminal μέσω του PLBUart16550.
- Δεν χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά, αντιθέτως όλα είναι συνδεδεμένα στο δίαυλο PLB.
- **Aurora σε λειτουργία Packet έκδοση μέτρησης throughput:** Πρόκειται για την σχεδίαση του συστήματος για τη μέτρηση throughput με προσθήκη ενός επιπλέον περιφερειακού και επιπλέον λογικής.
  - Ο πυρήνας είναι σε λειτουργία Packet
  - Διαθέτει διεπαφή με μήκος λέξης 2 bytes
  - Το επίπεδο μεταγλώττισης του κώδικα της εφαρμογής είναι O-3
  - Υπάρχει instruction και data cache
  - Έχει δημιουργηθεί BRAM μεγέθους 128 KB που είναι και το μέγιστο δυνατό μέγεθος για BRAM.
  - Περιλαμβάνονται 4 FIFOs 64 x 512 (πλάτος x βάθος)
  - Τα μηνύματα που δίνει ως έξοδο η εφαρμογή τυπώνονται στο terminal μέσω του PLBUart16550.
  - Δεν χρησιμοποιείται ο δίαυλος OPB για επιμέρους περιφερειακά, αντιθέτως όλα είναι συνδεδεμένα στο δίαυλο PLB.

Οι παραπάνω σχεδιάσεις με μικρές αλλαγές του τύπου αλλαγή της συνδεσμολογίας των ακροδεκτών (pins) των φωτοδιόδων (LEDs) για τον έλεγχο της ορθής λειτουργίας του συστήματος. Εφαρμόστηκαν και στην περίπτωση πραγματοποίησης μεταφοράς σε δύο πλακέτες XUP.

### 5.2.3 Δοκιμή αποστολής και λήψης δεδομένων με χρήση των συναρτήσεων *send receive* με αντιγραφή των δεδομένων μέσω DMA

Εκτελέστηκε πείραμα αποστολής μέσω DMA (Direct Memory Access) Transfer από 8 bytes μέχρι 1904 bytes σε λειτουργία Streaming και Packet με χρήση ενός XUP με εξωτερικό loopback που γίνεται με καλώδιο SATA και δύο XUP συνδεδεμένα με καλώδιο SATA 40cm. Κατ' αρχάς, τα δεδομένα προετοιμάζονται από την εφαρμογή χρήστη και ορίζονται ως μια αυξανόμενη ακολουθία αριθμών για λόγους ευκολίας. Έπειτα αποστέλλονται από τη μνήμη του επεξεργαστή PPC στη μνήμη του περιφερειακού που στη συγκεκριμένη περίπτωση είναι μια FIFO που ονομάζεται Write FIFO. Η μέθοδος που χρησιμοποιείται είναι η Direct Memory Access ώστε να εκτελεστεί η μεταφορά όσο πιο άμεσα γίνεται χωρίς να παρεμβάλλεται ο δίαυλος PLB που χρησιμοποιείται και από άλλα περιφερειακά. Έπειτα μέσω του περιφερειακού, του πυρήνα Aurora που εμπεριέχει, και των MGTs που χρησιμοποιεί, γίνεται αποστολή των δεδομένων. Τα δεδομένα μεταδίδονται μέσω καλωδίου SATA. Στη συνέχεια λαμβάνονται από άλλο περιφερειακό που φέρει πυρήνα Aurora και κάνει χρήση των MGTs. Τέλος, πάλι με τη μέθοδο Direct Memory Access αποστέλλονται τα ληφθέντα δεδομένα από τη Read FIFO, όπως λέγεται, του περιφερειακού, στην μνήμη του επεξεργαστή PPC. Το πείραμα εκτελέστηκε με επιτυχία. Πιο συγκεκριμένα, ελέγχθηκε η αξιόπιστη μεταφορά των δεδομένων χωρίς σφάλματα bit, λέξεων ή πλαισίων δεδομένων

και χωρίς σφάλματα στις επικεφαλίδες πλαισίων των πακέτων. Επίσης, διαπιστώθηκε η απρόσκοπτη λειτουργία της γραμμής και του καναλιού μετάδοσης με παρατήρηση των φωτοδιδόνων. Ακόμα επιβεβαιώθηκε ο πολύ υψηλός ρυθμός μεταφοράς των δεδομένων.

### *5.2.4 Δοκιμή αποστολής και λήψης δεδομένων με χρήση των συναρτήσεων send receive με αντιγραφή των δεδομένων μέσω διαύλου PLB*

Εκτελέστηκε πείραμα αποστολής μέσω του διαύλου PLB από 8 bytes μέχρι 1904 bytes σε λειτουργία Streaming και Packet με χρήση ενός XUP με εξωτερικό loopback που γίνεται με καλώδιο SATA και δύο XUP συνδεδεμένα με καλώδιο SATA 40cm. Κατ' αρχάς, τα δεδομένα προετοιμάζονται από την εφαρμογή χρήστη και ορίζονται ως μια αυξανόμενη ακολουθία αριθμών για λόγους ευκολίας. Έπειτα αποστέλλονται από τη μνήμη του επεξεργαστή PPC, μέσω του διαύλου PLB, στη μνήμη του περιφερειακού που στη συγκεκριμένη περίπτωση είναι μια FIFO που ονομάζεται Write FIFO. Έπειτα μέσω του περιφερειακού, του πυρήνα Aurora που εμπεριέχει, και των MGTs που χρησιμοποιεί, γίνεται αποστολή των δεδομένων. Τα δεδομένα μεταδίδονται μέσω καλωδίου SATA. Στη συνέχεια λαμβάνονται από άλλο περιφερειακό που φέρει πυρήνα Aurora και κάνει χρήση των MGTs. Τέλος, πάλι με τη χρήση του διαύλου PLB αποστέλλονται τα ληφθέντα δεδομένα από τη Read FIFO, όπως λέγεται, του περιφερειακού, στην μνήμη του επεξεργαστή PPC. Το πείραμα εκτελέστηκε με επιτυχία. Πιο συγκεκριμένα, ελέγχθηκε η αξιόπιστη μεταφορά των δεδομένων χωρίς σφάλματα bit, λέξεων ή πλαισίων δεδομένων και χωρίς σφάλματα στις επικεφαλίδες πλαισίων των πακέτων. Επίσης, διαπιστώθηκε η απρόσκοπτη λειτουργία της γραμμής και του καναλιού μετάδοσης με παρατήρηση των φωτοδιδόνων. Ακόμα επιβεβαιώθηκε ο πολύ υψηλός ρυθμός μεταφοράς των δεδομένων.

### *5.2.5 Δοκιμή αποστολής και λήψης απεριόριστου αριθμού δεδομένων απρόσκοπτα με χρήση των συναρτήσεων send receive και έλεγχο των FIFO*

Εκτελέστηκε πείραμα αποστολής είτε μέσω DMA είτε μέσω διαύλου PLB από 1904 bytes μέχρι 19040 bytes σε λειτουργία Streaming και Packet με χρήση δύο XUP συνδεδεμένων με καλώδιο SATA 40cm. Κατ' αρχάς, τα δεδομένα προετοιμάζονται από την εφαρμογή χρήστη και ορίζονται ως μια αυξανόμενη ακολουθία αριθμών για λόγους ευκολίας. Έπειτα αποστέλλονται από τη μνήμη του επεξεργαστή PPC στη μνήμη του περιφερειακού που στη συγκεκριμένη περίπτωση είναι μια FIFO που ονομάζεται Write FIFO σε τμήματα των 1904 bytes. Για αυτόν το σκοπό, εκτελείται ένα for loop στο περιφερειακό πομπό, που επαναλαμβάνει την κλήση της συνάρτησης αποστολής κι ένα αντίστοιχο στο περιφερειακό δέκτη που επαναλαμβάνει ίσες φορές τη συνάρτηση λήψης δεδομένων. Αυτό συμβαίνει γιατί έχει υπολογιστεί πειραματικά ότι μπορούν να οριστούν μέχρι 1904 bytes δεδομένων λόγω περιορισμένου μεγέθους μνήμης του επεξεργαστή την οποία και καταλαμβάνει η εφαρμογή χρήστη. Η μέθοδος Direct Memory Access χρησιμοποιείται ώστε να εκτελεστεί η μεταφορά όσο πιο άμεσα γίνεται χωρίς να παρεμβάλλεται ο δίαυλος PLB που χρησιμοποιείται και από άλλα περιφερειακά. Επίσης χρησιμοποιείται εναλλακτικά και ο δίαυλος PLB. Έπειτα μέσω του περιφερειακού, του πυρήνα Aurora που εμπεριέχει, και των MGTs που χρησιμοποιεί, γίνεται αποστολή των δεδομένων. Τα δεδομένα μεταδίδονται μέσω καλωδίου SATA. Στη συνέχεια λαμβάνονται από άλλο περιφερειακό που φέρει πυρήνα Aurora και κάνει χρήση των MGTs. Τέλος, πάλι με τη μέθοδο Direct Memory Access είτε με χρήση του διαύλου PLB, αποστέλλονται τα ληφθέντα δεδομένα από τη Read FIFO, όπως λέγεται, του



περιφερειακού, στην μνήμη του επεξεργαστή PPC. Το πείραμα εκτελέστηκε με επιτυχία. Πιο συγκεκριμένα, ελέγχθηκε η αξιόπιστη μεταφορά των δεδομένων χωρίς σφάλματα bit, λέξεων ή πλαισίων δεδομένων και χωρίς σφάλματα στις επικεφαλίδες πλαισίων των πακέτων. Επίσης, διαπιστώθηκε η απρόσκοπτη λειτουργία της γραμμής και του καναλιού μετάδοσης με παρατήρηση των φωτοδιδώδων. Ακόμα επιβεβαιώθηκε ο πολύ υψηλός ρυθμός μεταφοράς των δεδομένων.

#### *5.2.6 Μέτρηση του ρυθμού μεταφοράς δεδομένων με χρήση custom component για αυτό το σκοπό είτε με χρήση timer του EDK*

Εκτελέστηκε πείραμα αποστολής μέσω DMA είτε μέσω διαύλου PLB από 8 bytes μέχρι 1904 bytes σε λειτουργία Streaming και Packet με χρήση ενός XUP με εξωτερικό loopback που γίνεται με καλώδιο SATA και δύο XUP συνδεδεμένα με καλώδιο SATA 40cm. Πιο συγκεκριμένα, επιβεβαιώθηκε ο πολύ υψηλός ρυθμός μεταφοράς των δεδομένων. Για αυτό το σκοπό κατασκευάστηκε περιφερειακό με την ονομασία throughput\_timer. Αυτό το περιφερειακό ξεκινά να μετρά από τη στιγμή που πρωτοεμφανίζονται δεδομένα στην Write FIFO κι έπειτα αποθηκεύει την τιμή του μετρητή την ώρα που το πρώτο ληφθέν δεδομένο φτάνει στην Read FIFO. Έτσι υπολογίζεται ο λανθάνων χρόνος του συστήματος, δηλαδή η καθυστέρηση που εισάγει το μέσο γραμμής μετάδοσης. Στη συνέχεια θα αποθηκευτεί η τιμή του μετρητή την στιγμή που έχει φύγει και το τελευταίο ληφθέν δεδομένο από την Read FIFO. Αυτή η τιμή λόγω αυξημένης δυσκολίας στην άμεση μέτρησής της υπολογίζεται έμμεσα. Έτσι μετράται η τιμή του μετρητή κατά την οποία γεμίζει η Read FIFO. Κατόπιν πειραμάτων και δειγματοληψίας των σημάτων που ενεργούν στη σχεδίαση αυτή παρατηρείται ότι η Read FIFO χρειάζεται 1 κύκλο ρολογιού χρήστη USER\_CLK ώστε να εκτελέσει ανάγνωση μίας γραμμής δεδομένων της. Η ανάγνωση της Read FIFO εκτελείται κατόπιν εντολής στην εφαρμογή χρήστη η οποία έχει πολύ μεγάλη καθυστέρηση να εκκινήσει λόγο παρεμβολής της εφαρμογής και του επεξεργαστή. Συνεπώς συνοψίζοντας τις παραπάνω παρατηρήσεις γίνεται η εκτίμηση ότι είναι δυνατόν να υπάρξει σήμα που ενεργοποιεί την ανάγνωση της Read FIFO από την στιγμή που είναι διαθέσιμη σε αυτήν η πρώτη γραμμή δεδομένων. Έχοντας κατά νου ότι η ανάγνωση εκτελείται σε ένα κύκλο ρολογιού, λοιπόν, υπολογίζεται ότι η τιμή του μετρητή την στιγμή που έχει αδειάσει ολόκληρη η Read FIFO θα είναι ένας κύκλος επιπλέον της τιμής του μετρητή όταν έχουν φτάσει σε αυτή όλα τα δεδομένα. Με αυτόν τον τρόπο, λοιπόν, θα υπολογιστεί ο ρυθμός μεταφοράς των δεδομένων από τον ένα περιφερειακό στο άλλο. Το πείραμα εκτελέστηκε με επιτυχία κι επιτεύχθηκαν ρυθμοί μεταφοράς δεδομένων σταθερά πάνω από 1 Gbps με ρυθμό μετάδοσης στη γραμμή 1.5 Gbaud per second και κωδικοποίηση 8B/10B.

#### *5.2.7 Επιβεβαίωση λήψης σταλμένων δεδομένων και μέτρηση ρυθμού σφαλμάτων*

Εκτελέστηκε πείραμα αποστολής μέσω DMA είτε μέσω διαύλου PLB από 8 bytes μέχρι 1904 bytes σε λειτουργία Streaming και Packet με χρήση ενός XUP με εξωτερικό loopback που γίνεται με καλώδιο SATA και δύο XUP συνδεδεμένα με καλώδιο SATA 40cm. Πιο συγκεκριμένα, ελέγχθηκε η αξιόπιστη μεταφορά των δεδομένων χωρίς σφάλματα bit, λέξεων ή πλαισίων δεδομένων και χωρίς σφάλματα στις επικεφαλίδες πλαισίων των πακέτων. Επίσης, διαπιστώθηκε η απρόσκοπτη λειτουργία της γραμμής και του καναλιού μετάδοσης με παρατήρηση των φωτοδιδώδων. Κατά τη διάρκεια της λήψης των δεδομένων που έχουν σταλεί συγκρίνονται οι τιμές τους με τις αντίστοιχες τιμές που

υπολογίζονται εύκολα λόγω του γεγονότος ότι τα δεδομένα αποτελούν μια ακολουθία δεδομένων που αυξάνεται η τιμή τους κατά ένα.

### 5.3 Περιορισμοί

Η σχεδίαση που υλοποιείται σε αυτήν την εργασία λειτουργεί μέσα σε ορισμένα πλαίσια τα οποία εγγυώνται την ορθή λειτουργίας της.

Όσον αφορά την τροφοδοσία της συσκευής είναι απαραίτητο να είναι αδιάκοπη. Σε περίπτωση σφάλματος τροφοδοσίας δεν υπάρχει κάποιος μηχανισμός ανάκαμψης της σχεδίασης. Ακόμα το μέσον που αποτελεί το υλικό της γραμμής μετάδοσης (καλώδιο SATA) πρέπει να είναι συνδεδεμένο συνεχώς κατά τη μεταφορά δεδομένων. Σε αντίθετη περίπτωση θα υπάρχει απώλεια πληροφορίας. Δεν είναι δυνατόν να ξεπεραστεί η συχνότητα λειτουργίας ρολογιού σχεδίασης και αναφοράς των MGTs των 156.25 Mhz. Σημαντικός περιορισμός, επίσης, εισάγεται από τη χωρητικότητα μνήμης PLB BRAM του επεξεργαστή PowerPC που στην προκειμένη περίπτωση είναι 64 KB. Για αυτό δεν μπορεί να δεσμευθεί πολύ μεγάλος χώρος για τα δεδομένα των δοκιμών. Αυτό συνεπάγεται μέγιστο πλήθος δεδομένων προς αποστολή 1904 bytes. Το πρόβλημα αυτό επιλύθηκε εκτελώντας ένα βρόχο πολλών επαναλήψεων του κώδικα της αποστολής και της λήψης. Ακόμα ένας παράγοντας που περιορίζει το σύστημα και εισάγει καθυστέρηση είναι το γεγονός ότι η κλήση της συνάρτησης μεταφοράς δεδομένων από τις FIFO του περιφερειακού στη μνήμη του επεξεργαστή, όπως επίσης και η συνάρτηση που τυπώνει στο Terminal μηνύματα της διαδικασίας των δοκιμών, καθυστερεί να εκκινήσει πολλούς κύκλους. Επιπλέον περιορισμό αποτελεί η σύμβαση ότι το μέγεθος του πακέτου πρέπει να είναι αριθμός πολλαπλάσιος του 8. Κάτι τέτοιο γίνεται λόγω του γεγονότος ότι όταν εκτελείται η μεταφορά σε λειτουργία DMA τα δεδομένα μεταφέρονται υποχρεωτικά ανά 8 bytes (64 bits). Τέλος υπάρχει άτυπα η σύμβαση ότι το πακέτο πρέπει να έχει ελάχιστο μέγεθος 5 bytes. Έχει προβλεφθεί αυτό το μέγεθος για λόγους αποδοτικότητας καθώς υπάρχει η δυνατότητα αποστολής πακέτων μέχρι και 1 byte όμως πρόκειται για μια λειτουργία περιττή στη γενική περίπτωση και δεν υλοποιήθηκε.

### 5.4 Στατιστικά αποτελέσματα, συμπεράσματα

Κατόπιν των δοκιμών που έλαβαν χώρα στο εργαστήριο MHL διαπιστώθηκε η σωστή λειτουργία του συστήματος της εργασίας αυτής. Επιτεύχθηκε το επιθυμητό δηλαδή η σειριακή μεταφορά δεδομένων μέσω πλακέτας(ών) XUP σε πολύ υψηλές ταχύτητες.

Το κόστος υλοποίησης της σχεδίασης της εργασίας είναι μικρό. Αυτό το συμπέρασμα εξάγεται από την παρατήρηση της ποσότητας λογικής που χρησιμοποιείται. Παρατίθεται Πίνακας που συνοψίζει το βαθμό στον οποίο χρησιμοποιούνται λογικές πύλες, Lookup Tables, και κυκλώματα της συσκευής 2vp30ff896-7 που επιλέχθηκε.

#	My Design loopback		My Design 2 XUP	
<b>Slices</b>	5426 από 13696	39 %	3208 από 13696	23%
<b>Slice Flip Flop</b>	5759 από 27392	21 %	3471 από 27392	12%
<b>4 input LUTs</b>	8765 από 27392	31 %	5414 από 27392	19%
<b>used as logic</b>	7946		5008	
<b>used as Shift Registers</b>	599		406	
<b>used as RAMs</b>	220			
<b>IOs</b>	17		19	



<b>Bonded IOBs</b>	15 από 556	2 %	17 από 556	3%
<b>BRAMs</b>	48 από 136	35 %	78 από 136	57%
<b>GCLKs</b>	2 από 16	12 %	3 από 16	18%
<b>PPC405s</b>	1 από 2	50 %	1 από 2	50%
<b>GTs</b>	2 από 8	25 %	1 από 8	12%
<b>DCMs</b>	1 από 8	12 %	2 από 8	25%

Πίνακας 5.1 : Σύνοψη Χρησιμοποιούμενων Πόρων

Σχετικά με το ρολόι της σχεδίασης και του ρολογιού αναφοράς των MGTs, όπως έχει αναφερθεί, η ελάχιστη τιμή του είναι 60 MHz και η μέγιστη τιμή που μπορεί να επιτευχθεί είναι 156.25 Mhz. Μελετήθηκαν εμπειρικά το μέγιστο μέγεθος πακέτων MTU της σχεδίασης σε σχέση με το throughput πειραματικών αποτελεσμάτων για το βέλτιστο μέγιστο μέγεθος πακέτων MTU.

## ΚΕΦΑΛΑΙΟ 6

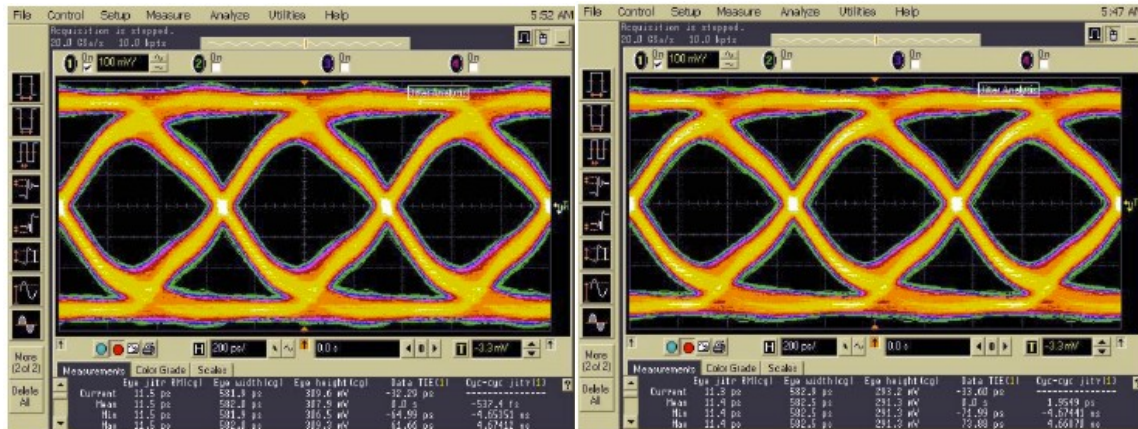
### ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ

1. Ενσωμάτωση της σχεδιάσής μας σε όλες τις σχεδιάσεις του εργαστηρίου ώστε να χειρίζεται την Είσοδο/Εξοδο των πειραματικών δεδομένων
2. Ανάπτυξη πρωτοκόλλου επιπέδου Δικτύου και πρωτοκόλλου επιπέδου Μεταφοράς ως επέκταση του πρωτοκόλλου επιπέδου Διασύνδεσης της εργασίας.
3. Ανάπτυξη πρωτοκόλλου συμβατού με το πρωτόκολλο SATA. Στόχος είναι η διασύνδεση με σκληρό δίσκο SATA ή με όποια συσκευή SATA. Ενδεχομένως για τη διασύνδεση με το σκληρό δίσκο ενός υπολογιστή να είναι απαραίτητη κάρτα PCI που να εκτελεί αυτή τη δουλειά.
4. Πλήρης αξιοποίηση του πρωτοκόλλου διασύνδεσης Aurora και παραμέτρων του, καθώς και βελτιστοποίηση της δικής μας σχεδίασης.
5. Πιο συγκεκριμένα, βελτιστοποίηση της υλοποίησης πολυπλεξίας κι αποπολυπλεξίας με ελάττωση των σημάτων και δημιουργία πιο περιεκτικού κώδικα όπου είναι δυνατό. Συγχώνευση, αν είναι δυνατόν, των δύο ειδών FIFO (Write Fifo, TX Fifo και Read Fifo, RX Fifo) και παράλληλα δημιουργία, αν χρειαστεί, αντίστοιχων drivers για το περιφερειακό `aurora_plb`.
6. Προσθήκη περισσότερης μνήμης στο σύστημα ώστε να ελαχιστοποιηθεί ο περιορισμός μνήμης για τα δεδομένα προς αποστολή.
7. Αξιοποίηση όλων των συσκευών MGT που διαθέτει μια πλακέτα XUP Virtex II Pro
8. Προσθήκη συνδέσεων μεταξύ περισσότερων MGTs και συνεπώς δημιουργία καναλιού με μεγαλύτερο αριθμό γραμμών.
9. Δημιουργία δικτύου που θα συνδέει παραπάνω από δυο XUP Virtex II Pro και ακόμα διασυνδέσεις μεταξύ διαφορετικών πλακετών που διαθέτουν τα κατάλληλα κυκλώματα του RocketIO, όπως επίσης και με PC.

## ΠΑΡΑΡΤΗΜΑ

### Διάγραμμα οφθαλμού σειριακής μετάδοσης δεδομένων

Η Xilinx προτείνει τη ρύθμιση της ιδιότητας TX\_DIF\_CTRL στα 600 (600 mV) και την ιδιότητα TXPREEMPHASIS στην τιμή 2 (25%) όταν καλώδιο SATA μήκους 1.0 μέτρο ή λιγότερο χρησιμοποιείται για την σύνδεση του MGT host στο MGT target. Τυπικά διαγράμματα ματιού για μετάδοση δεδομένων στα 1.5 Gb/s χρησιμοποιώντας καλώδια SATA 0.5 μέτρα και 1.0 μέτρο παρουσιάζονται στο Σχήμα A-1.



Σχήμα A-1: 1.5 Gb/s Σειριακή Μετάδοση Δεδομένων μέσω 0.5 και 1.0 μέτρου καλώδιο SATA

### Ονομασίες και τοποθεσίες καίριων ακροδεκτών (pins) του MGT

Αξίζει να σημειωθεί ότι με βάση αυτόν τον πίνακα επιλέγονται οι ακροδέκτες (pins) με τους οποίους συνδέεται η διαφορική είσοδος ρολογιού αναφοράς BREFCLK του MGT. Ως γνωστόν, αυτό το σήμα ρολογιού διατίθεται μέσω αποκλειστικής δρομολόγησης. Ο Πίνακας A-2 δείχνει τους αριθμούς ακροδέκτη του BREFCLK για όλες τις συσκευασίες. Σημειώτεον ότι αυτοί οι ακροδέκτες/δρομολογήσεις πρέπει να χρησιμοποιούνται για τις λειτουργίες του BREFCLK.

Συσκευασία	Άνω		Κάτω	
	BREFCLK Αριθμός Pin	BREFCLK2 Αριθμός Pin	BREFCLK Αριθμός Pin	BREFCLK2 Αριθμός Pin
FG256	A8/B8	A9/B9	R8/T8	T9/R9
FG456	C11/D11	C12/D12	W11/Y11	Y12/W12
FG676	B13/C13	B14/C14	AD13/AE13	AE14/AD14
FF672	B14/C14	C13/B13	AD14/AE14	AE13/AD13
FF896	F16/G16	G15/F15	AH16/AJ16	AJ15/AH15
FF1152	H18/J18	J17/H17	AK18/AL18	AL17/AK17
FF1148	N/A	N/A	N/A	N/A
FF1517	E20/D20	J20/K20	AR20/AT20	AL20/AK20
FF1704	G22/F22	F21/G21	AU22/AT22	AT21/AU21
FF1696	N/A	N/A	N/A	N/A

Πίνακας A-2 : Αριθμοί Ακροδεκτών (Pins) των BREFCLK

Πίνακας προσδιορισμού των ακροδεκτών (pins) Εισόδου/Εξόδου των MGTs που διαθέτει μια πλακέτα XUP Virtex II Pro

Signal	MGT Location	PAD Name	I/O Pin	Notes
SATA_PORT0_TXN	MGT_X0Y1	TXNPAD4	A27	HOST
SATA_PORT0_TXP	MGT_X0Y1	TXPPAD4	A26	–
SATA_PORT0_RXN	MGT_X0Y1	RXNPAD4	A24	–
SATA_PORT0_RXP	MGT_X0Y1	RXPPAD4	A25	–
SATA_PORT0_IDLE	–	–	B15	–
SATA_PORT1_TXN	MGT_X1Y1	TXNPAD6	A20	TARGET
SATA_PORT1_TXP	MGT_X1Y1	TXPPAD6	A19	–
SATA_PORT1_RXN	MGT_X1Y1	RXNPAD6	A17	–
SATA_PORT1_RXP	MGT_X1Y1	RXPPAD6	A18	–
SATA_PORT1_IDLE	–	–	AK3	–
SATA_PORT2_TXN	MGT_X2Y1	TXNPAD7	A14	HOST
SATA_PORT2_TXP	MGT_X2Y1	TXPPAD7	A13	–
SATA_PORT2_RXN	MGT_X2Y1	RXNPAD7	A11	–
SATA_PORT2_RXP	MGT_X2Y1	RXPPAD7	A12	–
SATA_PORT2_IDLE	–	–	C15	–
MGT_TXN	MGT_X3Y1	TXNPAD9	A7	USER
MGT_TXP	MGT_X3Y1	TXPPAD9	A6	–
MGT_RXN	MGT_X3Y1	RXNPAD9	A4	–
MGT_RXP	MGT_X3Y1	RXPPAD9	A5	–
MGT_CLK_N	–	–	G16	BREFCLK
MGT_CLK_P	–	–	F16	–
EXTERNAL_CLOCK_N	–	–	F15	BREFCLK2
EXTERNAL_CLOCK_P	–	–	G15	–

Πίνακας A-3 : Σήματα του SATA και του MGT

## 8B/10B Χαρακτήρες Ελέγχου

Παρατίθεται λίστα με την κωδικοποίηση 12 ειδικών συμβόλων γνωστών ως χαρακτήρες ελέγχου ή χαρακτήρες-K.

Όνομα	Hex	8 bits	RD -	RD +
K28.0	1C	00011100	0011110100	1100001011
K28.1	3C	00111100	0011111001	1100000110
K28.2	5C	01011100	0011110101	1100001010
K28.3	7C	01111100	0011110011	1100001100
K28.4	9C	10011100	0011110010	1100001101
K28.5	BC	10111100	0011111010	1100000101
K28.6	DC	11011100	0011110110	1100001001

K28.7	FC	11111100	0011111000	1100000111
K23.7	F7	11110111	1101101000	0001010111
K27.7	FB	11111011	1101101000	0010010111
K29.7	FD	11111101	1011101000	0100010111
K30.7	FE	11111110	0111101000	1000010111

Πίνακας A-4 : Χαρακτήρες Ελέγχου ή αλλιώς Χαρακτήρες K

Αυτοί οι χαρακτήρες ελέγχου χρησιμοποιούνται για λόγους ευθυγράμμισης, έλεγχου, και διαίρεση του εύρους ζώνης σε υπο-κανάλια.

### Λανθάνων Χρόνος Μονοπατιού Δεδομένων (Data Path Latency)

Με τις πολλές διαμορφώσεις του MGT, και οι δύο λανθάνουσες καταστάσεις του μονοπατιού δεδομένων, τόσο αυτή της αποστολής όσο και αυτή της λήψης ποικίλουν. Παρακάτω είναι πίνακας που παρέχει προσεγγιστικές λανθάνουσες καταστάσεις για κοινές διαμορφώσεις.

Εξάρτημα/Διεργασία		Λανθάνων Χρόνος		
Υλικό TX/Διεπαφή GT		1 Byte Data Path 2.5 κύκλοι TXUSRCLK2 1.25 κύκλοι TXUSRCLK	2 Byte Data Path 1 κύκλος TXUSRCLK2 1 κύκλος TXUSRCLK	4 Byte Data Path 1.25 κύκλοι TXUSRCLK2 2.5 κύκλοι TXUSRCLK
TX CRC	συμπεριλαμβάνεται	7 κύκλοι TXUSRCLK		
	παρακάμτεται	1 κύκλος TXUSRCLK		
8B/10B Κωδικοποιητής	συμπεριλαμβάνεται	1 κύκλος TXUSRCLK		
	παρακάμτεται	1 κύκλος TXUSRCLK		
TX FIFO		4 κύκλοι TXUSRCLK (±0.5)		
TX SERDES		SERDES_10B = FALSE 1.5 κύκλοι TXUSRCLK	SERDES_10B = TRUE 0.5 κύκλος TXUSRCLK (προσεγγιστικά)	
RX SERDES		1.5 κύκλοι ανακτημένου ρολογιού (RXRECCLK)		
Εντοπισμός Κόμμα/ Επανευθυγράμμιση		2.5 ή 3.5 κύκλοι ανακτημένου ρολογιού (μερικά bits παρακάμτουν ένα καταχωρητή, αναλόγως την ευθυγράμμιση κόμματος)		
8B/10B Αποκωδικοποιητής	συμπεριλαμβάνεται	1 κύκλος ανακτημένου ρολογιού		
	παρακάμτεται	1 κύκλος ανακτημένου ρολογιού		
RX FIFO		18 κύκλοι RXUSRCLK (±0.5)		
Υλικό RX/Διεπαφή GT		1 Byte Data Path 2.5 κύκλοι RXUSRCLK2 1.25 κύκλοι RXUSRCLK	2 Byte Data Path 1 κύκλος RXUSRCLK2 1 κύκλος RXUSRCLK	4 Byte Data Path 1.25 κύκλοι RXUSRCLK2 2.5 κύκλοι RXUSRCLK

Πίνακας A-5 : Ο Λανθάνοντας Χρόνος μέσα από Διάφορες Συνιστώσες/Διεργασίες του Πομποδέκτη

### Διόρθωση Ρολογιού

						Μέγιστος Αριθμός Κύκλων Πριν τη Διόρθωση			
Συχνότητα Ταλαντωτή (MHz)	Ακρίβεια OSC (PPM)	Ταχύτητα Γραμμής (GB/s)	Fmax (MHz)	Fmin (MHz)	Διαφορά /Κύκλο (ps)	Αφαίρεση 1 Ακολουθίας	Αφαίρεση 2 Ακολουθιών	Αφαίρεση 3 Ακολουθιών	Αφαίρεση 4 Ακολουθιών
156.25	100	3.125	156.26 56	156.23 44	1.2800	4,999	9,999	14,998	19,998
156.25	50	3.125	156.25 78	156.24 22	0.6400	9,999	19,998	29,998	39,997
156.25	20	3.125	156.25 31	156.24 69	0.2560	24,999	49,999	74,998	99,997
125	100	2.500	125.01 25	124.98 75	1.6000	4,999	9,998	14,998	19,997
125	50	2.500	125.00 63	124.99 38	0.8000	9,999	19,998	29,998	39,997
125	20	2.500	125.00 25	124.99 75	0.3200	24,999	49,998	74,998	99,997
62.5	100	1.250	62.506 3	62.493 8	3.2000	4,999	9,998	14,998	19,997
62.5	50	1.250	62.503 1	62.496 9	1.6000	9,999	19,998	29,998	39,997

62.5	20	1.250	62.501 3	62.498 8	0.6400	24,999	74,998	74.998	99,997
------	----	-------	-------------	-------------	--------	--------	--------	--------	--------

Πίνακας A-6 : Διόρθωση Ρολογιού

### Σήματα Ελέγχου Δεσμού Καναλιού

Άλλα σήματα ελέγχου περιλαμβάνουν τα χαρακτηριστικά:

- CHAN\_BOND\_WAIT
- CHAN\_BOND\_OFFSET
- CHAN\_BOND\_LIMIT
- CHAN\_BOND\_ONE\_SHOT

Τυπικές τιμές για αυτά τα χαρακτηριστικά είναι:

CHAN\_BOND\_WAIT = 8

CHAN\_BOND\_OFFSET = CHAN\_BOND\_WAIT

CHAN\_BOND\_LIMIT = 2 x CHAN\_BOND\_WAIT

Χαμηλότερες τιμές δεν ενδείκνυνται. Χρησιμοποιούνται υψηλότερες τιμές, μόνο αν οι ακολουθίες δεσμού καναλιού είναι απομακρυσμένες περισσότερο από 17 bytes. Ο Πίνακας A-7 δείχνει διαφορετικές ρυθμίσεις για το CHAN\_BOND\_ONE\_SHOT και το ENCHANSYNC σε εφαρμογές Master και Slave.

	Master	Slave
CHAN_BOND_ONE_SHOT	TRUE ή FALSE όπως είναι επιθυμητό	FALSE
ENCHANSYNC	Δυναμικός έλεγχος όπως είναι επιθυμητό	Tie High

Πίνακας A-7: Ρυθμίσεις Master/Slave των Χαρακτηριστικών Δεσμού Καναλιού

## BIBΛΙΟΓΡΑΦΙΑ

- [1] **Xup Virtex II Pro User Guide**, UG069 (v1.0) March 8, 2005
- [2] **RocketIO™ Transceiver User Guide**, rocketIOuserGuide\_ug024.pdf, UG024 (v2.5) December 9, 2004
- [3] **RocketIO Transceiver Bit-Error Rate Tester**, rocketIOBERT\_xapp661.pdf, Author : Dai Huang and Michael Matera
- [4] [http://www.xilinx.com/products/design\\_resources/conn\\_central/grouping/aurora.htm](http://www.xilinx.com/products/design_resources/conn_central/grouping/aurora.htm)
- [5] **LogiCORE Aurora v2.8 User Guide**, ug061.pdf, UG061 (v2.8) October 10, 2007
- [6] **IP Overview: Aurora Reference Design**, ip\_aurora.pdf, 1- 62 © 2004 Xilinx
- [7] **Aurora Quick Start Guide XUP Virtex-2 Pro Development System**, Aurora\_QuickStart.pdf
- [8] **Simulation Waveform**, Simulation\_Waveforms.pdf
- [9] **Aurora Bus Functional Model**, ug058\_abfm.pdf, ug058 (v1.4) January 18, 2006
- [10] **Aurora 804 Demonstration**, aurora\_804\_ml321\_demo.pdf, GVDT132 (v2.0) January 29, 2004
- [11] [Aurora Transceiver for the PLB](#), aurora-transceiver-for-plb.html
- [12] <http://virtex2pro.blogspot.com/>
- [13] [http://www.avmentor.gr/tech/jitter\\_0.htm](http://www.avmentor.gr/tech/jitter_0.htm)
- [14] **“White Paper on Transmit Pre-emphasis and Receive Equalization”** Authors: Johnny Zhang, Zhi Wong
- [15] **“Communications Systems Engineering”** Authors: John G. Proakis , Masoud Salehi, 2002
- [16] **“Communication Networks”** Authors: Jean Walrand, 1998
- [17] High-Speed Serial I/O Made Simple, **“Connectivity Solutions: Edition 1.0”**, Abhijit Athavale (Marketing Manager, Connectivity Solutions, Xilinx, Inc.) and Carl Christensen (Technical Marketing)
- [18] **Parameterizable LocalLink FIFO**, XAPP691 (v1.0.1) May 10, 2007, Author: Wen Ying Wei, Dai Huang
- [19] **XLVDSPro Demonstration Boards**, UG037 (v1.4) June 03, 2004
- [20] **PLB IPIF (v2.01a)**, DS448 August 11, 2004
- [21] **Processor Local Bus (PLB) v3.4**, (v1.01a), DS400 September 23, 2004
- [22] **FIFO Generator v2.2**, DS317 August 31, 2005
- [23] **EDK Concepts, Tools, and Techniques**, P/N XTP013 (Version 9.1i)
- [24] **OS and Libraries Document Collection**, EDK 9.1i March 21, 2007
- [25] [http://en.wikipedia.org/wiki/Transmission\\_line](http://en.wikipedia.org/wiki/Transmission_line)